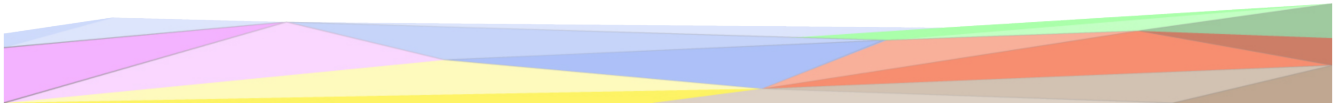
Dipartimento di Ingegneria e Scienza dell'Informazione

– KnowDive Group –

KGE 2022 - Project Report

Document Data:	Reference Persons:
January 17, 2023	Jose Enrique, Kasper Lange



Index:

Revision History:

Revision	Date	Author	Description of Changes
0.1	20.04.2020	Fausto Giunchiglia	Document created
0.2	10.11.2022	Kasper Lange	Template adjusted for specific project
0.3	11.11.2022	Kasper Lange	Added some data file descriptions inception phase
0.4	12.11.2022	Jose Enrique Barco	Added Purpose
0.5	12.11.2022	Kasper Lange	Finished Adding data file descriptions
0.6	13.11.2022	Kasper Lange	Added Knowledge sources
0.7	13.11.2022	Jose Enrique Barco	Added POI file description inception phase
0.8	16.11.2022	Jose Enrique Barco	Added ER schema
0.9	21.11.2022	Kasper Lange	Added Teleology
0.10	29.11.2022	Kasper Lange	Added choice explanations
0.11	01.12.2022	Jose Enrique Barco	Added Simplified ER schema
0.12	03.12.2022	Kasper Lange	Added Formal modelling
0.13	18.12.2022	Kasper Lange	Changed ER diagram
0.14	06.01.2023	Kasper Lange	Updated Formal Modeling Phase
0.15	08.01.2023	Kasper Lange	Started Data Integration Phase
0.16	12.01.2023	Jose Enrique Barco	Added Queries, Data Mapping, and Entity Matching
0.17	15.01.2023	Kasper Lange	Added Conclusion and Open Issues
0.18	15.01.2023	Jose Enrique Barco	Final touches in wordings
1.0	17.01.2023	Kasper Lange	Final Layout Touches

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 in order to enhance the reusabiltiy of the resources handled and produced during the process. A clear description of the resources and the process developed, provides a clear understanding of the KGE project, thus serving such an information to external readers in order to exploit that in new projects.

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

- Section 2: The project's purpose and the domain of interest and the resources involved (both schema and data resources) in the integration process.
- Section 2: The input resources considered by the KGE project.
- Section 4, 5, 6, 7: The integration process along the different iTelos phases, respectively.
- Section 8: How the result of the KGE process (the KG) can be exploited.
- Section 9: Conclusions and open issues summary.

2 Purpose and Domain of Interest (DoI)

This section has to report and describe:

- The project's purpose, by reporting the main Purpose as expressed by the final user.
- The definition of the project's Domain of Interest, by defining its boundaries in space and time.

2.1 Purpose

The purpose of this knowledge graph is providing the fastest and/or most convenient way to reach the most relevant landmarks in Trentino, using public transport. Among the users of this Knowledge graph can be found: tourists and local people that want to visit their region.

3 Data Sources

This section has to report and describe the input resources considered:

- **Knowledge sources**: The sources for reference schemas and ontologies initially collected to satisfy the purpose along the KGE process. The knowledge resources initial metadata have to be reported here.
- **Data sources**: The sources for datasets initially collected to satisfy the purpose along the KGE process. The data resources initial metadata have to be reported here.

3.1 Knowledge sources

3.1.1 GTFS

GTFS or General Transit Feed Specification is a schema that defines the common format for public transportation schedules and associated geographic information. This allows public transport providers to provide their data to developers who can create applications that use this data. For more information we refer you to `https://developers.google.com/transit/gtfs`.

3.2 Open issues:

3.3 Data sources

3.3.1 Trentino Transpot Data

This dataset can be found in the trentino-transporti webpage, in their open-data section, in the following url: `https://www.trentinotrasporti.it/open-data` We accessed for the last time the 8th of November of 2022. This dataset is divided into different files as the GTFS specification states. We have followed the following steps in order to verify the dataset:

- Using the validation tool provided by google, we validated the dataset obtaining the following results:
  1. There was an unknown file: stopslevel.txt.
  2. There are 2 recommended fields missing int the file feed info.txt.
  3. The column timepoint is missing from the file stop times.txt.
  4. There are 22 stops that are too far from the assigned shape.
  5. There are 14 pairs of stop which are considered to be in the wrong order.
- Checking this results we have taken the following decisions:
  1. The file stopslevel.txt is changed to levels.txt since it had the corrosponding fields and thus was only a different name compared to the standard gtfs names.
  2. For our purpose the file feed info.txt is not needed.
  3. We have set the column timepoint from the file stop times.txt to the value ”0”.
  4. For our purpose and abilities, we can’t assign the exact location to the stops that are considered to be too far away from the shape.
  5. For our purpose it’s not a problem the stops that are in the wrong order.

3.3.2 Points of interest

This dataset can be found in the OpenData Trentino web, in the following url: `https://dati.trentino.it/dataset/punti-di-interesse-del-trentino`. We accessed for the last time the 12th of November 2022. The dataset is not ready to use directly from the download

point so we have to perform the following transformations:

- Delete the substrings that have the following regular expresion: ”* [0-9]* *”.
- Merge all the different dictionaries into the same json.
- Erase all the columns that are not interesting for our purpose.

3.4 datafiles gtfs:

the following tables describe the file structre and files contained in gtfs and are originally created by google. The tables can be found at `https://developers.google.com/transit/gtfs/reference`.

- agency.txt

Field Name	Type	Required	Description
agency id	ID	Conditionally Required	Identifies a transit brand which is often synonymous with a transit agency. Note that in some cases, such as when a single agency operates multiple separate services, agencies and brands are distinct. This document uses the term ”agency” in place of ”brand”. A dataset may contain data from multiple agencies. This field is required when the dataset contains data for multiple transit agencies, otherwise it is optional.
agency name	Text	Required	Full name of the transit agency.
agency url	URL	Required	URL of the transit agency.
agency timezone	Timezone	Required	Timezone where the transit agency is located. If multiple agencies are specified in the dataset, each must have the same agency timezone.
agency lang	Language code	Optional	Primary language used by this transit agency. This field helps GTFS consumers choose capitalization rules and other language-specific settings for the dataset.
agency phone	Phone number	Optional	A voice telephone number for the specified agency. This field is a string value that presents the telephone number as typical for the agency’s service area. It can and should contain punctuation marks to group the digits of the number. Dialable text (for example, TriMet’s 503-238-RIDE) is permitted, but the field must not contain any other descriptive text.
agency fare url	URL	Optional	URL of a web page that allows a rider to purchase tickets or other fare instruments for that agency online.
agency email	Email	Optional	Email address actively monitored by the agency’s customer service department. This email address should be a direct contact point where transit riders can reach a customer service representative at the agency.

- stops.txt

Field Name	Type	Required	Description

stop ID	ID	Required	Identifies a stop, station, or station entrance. The term ”station entrance” refers to both station entrances and station exits. Stops, stations or station entrances are collectively referred to as locations. Multiple routes may use the same stop.
Stop code	Text	Optional	Short text or a number that identifies the location for riders. These codes are often used in phone-based transit information systems or printed on signage to make it easier for riders to get information for a particular location. The stop code can be the same as stop id if it is public facing. This field should be left empty for locations without a code presented to riders.
stop name	Text	Conditionally Required	Name of the location. Use a name that people will understand in the local and tourist vernacular. When the location is a boarding area (location type=4), the stop name should contains the name of the boarding area as displayed by the agency. It could be just one letter (like on some European intercity railway stations), or text like “Wheelchair boarding area” (NYC’s Subway) or “Head of short trains” (Paris’ RER). Conditionally Required: • Required for locations which are stops (location type=0), stations (location type=1) or entrances/exits (location type=2). • Optional for locations which are generic nodes (location type=3) or boarding areas (location type=4).
stop desc	Text	Optional	Description of the location that provides useful, quality information. Do not simply duplicate the name of the location.

stop lat	Latitude	Conditionally Required	Latitude of the location. Conditionally Required: • Required for locations which are stops (location type=0), stations (location type=1) or entrances/exits (location type=2). • Optional for locations which are generic nodes (location type=3) or boarding areas (location type=4).
stop lon	Longitude	Conditionally Required	Longitude of the location. Conditionally Required: • Required for locations which are stops (location type=0), stations (location type=1) or entrances/exits (location type=2). • Optional for locations which are generic nodes (location type=3) or boarding areas (location type=4).
zone id	ID	Conditionally Required	Identifies the fare zone for a stop. This field is required if providing fare information using fare rules.txt, otherwise it is optional. If this record represents a station or station entrance, the zone id is ignored.
stop url	URL	Optional	URL of a web page about the location. This should be different from the agency.agency url and the routes.route url field values.

location type	Enum	Optional	Type of the location: • 0 (or empty): Stop (or Platform). A location where passengers board or disembark from a transit vehicle. Is called a platform when defined within a parent station. • 1: Station. A physical structure or area that contains one or more platform. • 2: Entrance/Exit. A location where passengers can enter or exit a station from the street. If an entrance/exit belongs to multiple stations, it can be linked by pathways to both, but the data provider must pick one of them as parent. • 3: Generic Node. A location within a station, not matching any other location type, which can be used to link together pathways define in pathways.txt. • 4: Boarding Area. A specific location on a platform, where passengers can board and/or alight vehicles.
partent station	ID referencing stops.stop id	Confitionally Required	Defines hierarchy between the different locations defined in stops.txt. It contains the ID of the parent location, as followed: • Stop/platform (location type=0): the parent station field contains the ID of a station. • Station (location type=1): this field must be empty. • Entrance/exit (location type=2) or generic node (location type=3): the parent station field contains the ID of a station (location type=1) • Boarding Area (location type=4): the parent station field contains ID of a platform. Conditionally Required: • Required for locations which are entrances (location type=2), generic nodes (location type=3) or boarding areas (location type=4). • Optional for stops/platforms (location type=0). • Forbidden for stations (location type=1).

stop timezone	Timezone	Optional	Timezone of the location. If the location has a parent station, it inherits the parent station's timezone instead of applying its own. Stations and parentless stops with empty stop timezone inherit the timezone specified by agency.agency timezone. If stop timezone values are provided, the times in stop times.txt should be entered as the time since midnight in the timezone specified by agency.agency timezone. This ensures that the time values in a trip always increase over the course of a trip, regardless of which timezones the trip crosses.
wheelchair boarding	Enum	Optional	Indicates whether wheelchair boardings are possible from the location. Valid options are: For parentless stops: 0 or empty - No accessibility information for the stop. 1 - Some vehicles at this stop can be boarded by a rider in a wheelchair. 2 - Wheelchair boarding is not possible at this stop. For child stops: 0 or empty - Stop will inherit its wheelchair boarding behavior from the parent station, if specified in the parent. 1 - There exists some accessible path from outside the station to the specific stop/platform. 2 - There exists no accessible path from outside the station to the specific stop/platform. For station entrances/exits: 0 or empty - Station entrance will inherit its wheelchair boarding behavior from the parent station, if specified for the parent. 1 - Station entrance is wheelchair accessible. 2 - No accessible path from station entrance to stops/platforms.

level id	ID referencing levels. level ID	Optional	Level of the location. The same level can be used by multiple unlinked stations.
platform code	Text	Optional	Platform identifier for a platform stop (a stop belonging to a station). This should be just the platform identifier (eg. G or 3). Words like ”platform” or ”track” (or the feed’s language-specific equivalent) should not be included. This allows feed consumers to more easily internationalize and localize the platform identifier into other languages.

- routes.txt

Field Name	Type	Required	Description
route ID	ID	Required	Identifies a route.
Agency id	ID referencing agency.agency id	Conditionally required	Agency for the specified route. This field is required when the dataset provides data for routes from more than one agency in agency.txt, otherwise it is optional.
route short name	Text	Conditionally required	Short name of a route. This will often be a short, abstract identifier like ”32”, ”100X”, or ”Green” that riders use to identify a route, but which doesn’t give any indication of what places the route serves. Either route short name or route long name must be specified, or potentially both if appropriate.
route long name	Text	Conditionally required	Full name of a route. This name is generally more descriptive than the route short name and often includes the route’s destination or stop. Either route short name or route long name must be specified, or potentially both if appropriate.

route desc	Text	Optional	Description of a route that provides useful, quality information. Do not simply duplicate the name of the route. Example: ”A” trains operate between Inwood-207 St, Manhattan and Far Rockaway-Mott Avenue, Queens at all times. Also from about 6AM until about midnight, additional ”A” trains operate between Inwood-207 St and Lefferts Boulevard (trains typically alternate between Lefferts Blvd and Far Rockaway).
route type	Enum	Required	Indicates the type of transportation used on a route. Valid options are: 0 - Tram, Streetcar, Light rail. Any light rail or street level system within a metropolitan area. 1 - Subway, Metro. Any underground rail system within a metropolitan area. 2 - Rail. Used for intercity or long-distance travel. 3 - Bus. Used for short- and long-distance bus routes. 4 - Ferry. Used for short- and long-distance boat service. 5 - Cable tram. Used for street-level rail cars where the cable runs beneath the vehicle, e.g., cable car in San Francisco. 6 - Aerial lift, suspended cable car (e.g., gondola lift, aerial tramway). Cable transport where cabins, cars, gondolas or open chairs are suspended by means of one or more cables. 7 - Funicular. Any rail system designed for steep inclines. 11 - Trolleybus. Electric buses that draw power from overhead wires using poles. 12 - Monorail. Railway in which the track consists of a single rail or a beam.
route url	URL	Optional	URL of a web page about the particular route. Should be different from the agency.agency url value.

route color	Color	Optional	Route color designation that matches public facing material. Defaults to white (FFFFFF) when omitted or left empty. The color difference between route color and route text color should provide sufficient contrast when viewed on a black and white screen.
route text Color	color	Optional	Legible color to use for text drawn against a background of route color. Defaults to black (000000) when omitted or left empty. The color difference between route color and route text color should provide sufficient contrast when viewed on a black and white screen.
route sort order	non-negative integer	Optional	Orders the routes in a way which is ideal for presentation to customers. Routes with smaller route sort order values should be displayed first.
continuous pickup	Enum	Optional	Indicates whether a rider can board the transit vehicle anywhere along the vehicle's travel path. The path is described by shapes.txt on every trip of the route. Valid options are: 0 - Continuous stopping pickup. 1 or empty - No continuous stopping pickup. 2 - Must phone an agency to arrange continuous stopping pickup. 3 - Must coordinate with a driver to arrange continuous stopping pickup. The default continuous pickup behavior defined in routes.txt can be overridden in stop times.txt.

continous drop off	Enum	Optional	Indicates whether a rider can alight from the transit vehicle at any point along the vehicle's travel path. The path is described by shapes.txt on every trip of the route. Valid options are: 0- Continuous stopping drop-off. 1 or empty - No continuous stopping drop-off. 2 - Must phone an agency to arrange continuous stopping drop-off. 3 - Must coordinate with a driver to arrange continuous stopping drop-off. The default continuous drop-off behavior defined in routes.txt can be overridden in stop times.txt.

- trips.txt

Field Name	Type	Required	Description
route id	ID referencing routes.route id	Required	Identifies a route.
service id	ID referencing calendar.service id or calendar dates.service id	Required	Identifies a set of dates when service is available for one or more routes.
trip id	ID	Required	Identifies a trip.
trip headsign	text	Optional	Text that appears on signage identifying the trip's destination to riders. Use this field to distinguish between different patterns of service on the same route. If the headsign changes during a trip, trip headsign can be overridden by specifying values for the stop times.stop headsign.

trip short name	Text	Optional	Public facing text used to identify the trip to riders, for instance, to identify train numbers for commuter rail trips. If riders do not commonly rely on trip names, leave this field empty. A trip short name value, if provided, should uniquely identify a trip within a service day; it should not be used for destination names or limited/express designations.
direction id	Enum	Optional	ndicates the direction of travel for a trip. This field is not used in routing; it provides a way to separate trips by direction when publishing time tables. Valid options are: 0 - Travel in one direction (e.g. outbound travel). 1 - Travel in the opposite direction (e.g. inbound travel). Example: The trip headsign and direction id fields could be used together to assign a name to travel in each direction for a set of trips. A trips.txt file could contain these records for use in time tables: trip id,...,trip headsign,direction id 1234,...,Airport,0 1505,...,Downtown,1
block id	ID	Optional	Identifies the block to which the trip belongs. A block consists of a single trip or many sequential trips made using the same vehicle, defined by shared service days and block id. A block id can have trips with different service days, making distinct blocks. See the example below
shape id	ID referencing shapes.shape id	Conditionally required	Identifies a geospatial shape that describes the vehicle travel path for a trip. Conditionally required: This field is required if the trip has continuous behavior defined, either at the route level or at the stop time level. Otherwise, it's optional.

wheelchair accessible	Enum	Optional	Indicates wheelchair accessibility. Valid options are: 0 or empty - No accessibility information for the trip. 1 - Vehicle being used on this particular trip can accommodate at least one rider in a wheelchair. 2 - No riders in wheelchairs can be accommodated on this trip.
bikes allowed	Enum	Optional	Indicates whether bikes are allowed. Valid options are: 0 or empty - No bike information for the trip. 1 - Vehicle being used on this particular trip can accommodate at least one bicycle. 2 - No bicycles are allowed on this trip.

- stop times.txt

Field Name	Type	Required	Description
trip id	ID referencing trips.trip id	Required	Identifies a trip.

arrival time	Time	Conditionally required	Arrival time at a specific stop for a specific trip on a route. If there are not separate times for arrival and departure at a stop, enter the same value for arrival time and departure time. For times occurring after midnight on the service day, enter the time as a value greater than 24:00:00 in HH:MM:SS local time for the day on which the trip schedule begins. Scheduled stops where the vehicle strictly adheres to the specified arrival and departure times are timepoints. If this stop is not a timepoint, it is recommended to provide an estimated or interpolated time. If this is not available, arrival time can be left empty. Further, indicate that interpolated times are provided with timepoint=0. If interpolated times are indicated with timepoint=0, then time points must be indicated with timepoint=1. Provide arrival times for all stops that are time points. An arrival time must be specified for the first and the last stop in a trip.
departure time	Time	Conditionally required	Departure time from a specific stop for a specific trip on a route. For times occurring after midnight on the service day, enter the time as a value greater than 24:00:00 in HH:MM:SS local time for the day on which the trip schedule begins. If there are not separate times for arrival and departure at a stop, enter the same value for arrival time and departure time. See the arrival time description for more details about using timepoints correctly. The departure time field should specify time values whenever possible, including non-binding estimated or interpolated times between timepoints.

stop id	ID referencing stops.stop id	Required	Identifies the serviced stop. All stops serviced during a trip must have a record in stop times.txt. Referenced locations must be stops, not stations or station entrances. A stop may be serviced multiple times in the same trip, and multiple trips and routes may service the same stop.
stop sequence	NOnnegative integer	Required	Order of stops for a particular trip. The values must increase along the trip but do not need to be consecutive. Example: The first location on the trip could have a stop sequence=1, the second location on the trip could have a stop sequence=23, the third location could have a stop sequence=40, and so on.
Stop headsign	Text	Optional	Text that appears on signage identifying the trip's destination to riders. This field overrides the default trips.trip headsign when the headsign changes between stops. If the headsign is displayed for an entire trip, use trips.trip headsign instead. A stop headsign value specified for one stop time does not apply to subsequent stop times in the same trip. If you want to override the trip headsign for multiple stop times in the same trip, the stop headsign value must be repeated in each stop time row.
pickup Type	Enum	Optional	Indicates pickup method. Valid options are: 0 or empty - Regularly scheduled pickup. 1 - No pickup available. 2 - Must phone agency to arrange pickup. 3 - Must coordinate with driver to arrange pickup.

Drop off type	Enum	Optional	Indicates drop off method. Valid options are: 0 or empty - Regularly scheduled drop off. 1 - No drop off available. 2 - Must phone agency to arrange drop off. 3 - Must coordinate with driver to arrange drop off.
continuous pickup	Enum	Optional	Indicates whether a rider can board the transit vehicle at any point along the vehicle's travel path. The path is described by shapes.txt, from this stop time to the next stop time in the trip's stop sequence. Valid options are: 0 - Continuous stopping pickup. 1 or empty - No continuous stopping pickup. 2 - Must phone an agency to arrange continuous pickup. 3 - Must coordinate with a driver to arrange continuous stopping pickup. The continuous pickup behavior indicated in stop times.txt overrides any behavior defined in routes.txt.
continuous drop off	Enum	Optional	ndicates whether a rider can alight from the transit vehicle at any point along the vehicle's travel path as described by shapes.txt, from this stop time to the next stop time in the trip's stop sequence. 0 - Continuous stopping drop off. 1 or empty - No continuous stopping drop off. 2 - Must phone an agency to arrange continuous drop off. 3 - Must coordinate with a driver to arrange continuous stopping drop off. The continuous drop-off behavior indicated in stop times.txt overrides any behavior defined in routes.txt.

shape dist traveled	Non negative float	Optional	Actual distance traveled along the associated shape, from the first stop to the stop specified in this record. This field specifies how much of the shape to draw between any two stops during a trip. Must be in the same units used in shapes.txt. Values used for shape dist traveled must increase along with stop sequence; they cannot be used to show reverse travel along a route. Example: If a bus travels a distance of 5.25 kilometers from the start of the shape to the stop, shape dist traveled=5.25.
timepoint	Enum	Optional	Indicates if arrival and departure times for a stop are strictly adhered to by the vehicle or if they are instead approximate and/or interpolated times. This field allows a GTFS producer to provide interpolated stop-times, while indicating that the times are approximate. Valid options are: 0 - Times are considered approximate. 1 or empty - Times are considered exact.

- calendar.txt

Field Name	Type	Required	Description
service ID	ID	Required	Uniquely identifies a set of dates when service is available for one or more routes. Each service id value can appear at most once in a calendar.txt file.
monday	Enum	Required	Indicates whether the service operates on all Mondays in the date range specified by the start date and end date fields. Note that exceptions for particular dates may be listed in calendar dates.txt. Valid options are: 1 - Service is available for all Mondays in the date range. 0 - Service is not available for Mondays in the date range.

tuesday	Enum	Required	Functions in the same way as monday except applies to Tuesdays
wednesday	Enum	Required	Functions in the same way as monday except applies to Wednesday
thursday	Enum	Required	Functions in the same way as monday except applies to Rhursday
friday	Enum	Required	Functions in the same way as monday except applies to Friday
saturday	Enum	Required	Functions in the same way as monday except applies to Saturday
sunday	Enum	Required	Functions in the same way as monday except applies to Sunday
start date	Date	Required	Start service day for the service interval.
end date	Date	Required	End service day for the service interval. This service day is included in the interval.

- calendar dates.txt

Field Name	Type	Required	Description
service id	ID referencing calendar.service id or ID	Required	Identifies a set of dates when a service exception occurs for one or more routes. Each (service id, date) pair can only appear once in calendar dates.txt if using calendar.txt and calendar dates.txt in conjunction. If a service id value appears in both calendar.txt and calendar dates.txt, the information in calendar dates.txt modifies the service information specified in calendar.txt.
date	Daye	Required	Date when service exception occurs.

exception type	Enum	Required	Indicates whether service is available on the date specified in the date field. Valid options are: 1 - Service has been added for the specified date. 2 - Service has been removed for the specified date. Example: Suppose a route has one set of trips available on holidays and another set of trips available on all other days. One service id could correspond to the regular service schedule and another service id could correspond to the holiday schedule. For a particular holiday, the calendar dates.txt file could be used to add the holiday to the holiday service id and to remove the holiday from the regular service id schedule.

- fare attributes.txt

Field Name	Type	Required	Description
fare id	ID	Required	Identifies a fare class.
price	non negative float	Required	Fare price, in the unit specified by currency type.
currency type	Currency code	required	Currency used to pay the fare.
payment method	Enum	Required	Indicates when the fare must be paid. Valid options are: 0 - Fare is paid on board. 1 - Fare must be paid before boarding.
transfers	Enum	Required	Indicates the number of transfers permitted on this fare. The fact that this field can be left empty is an exception to the requirement that a Required field must not be empty. Valid options are: 0 - No transfers permitted on this fare. 1 - Riders may transfer once. 2 - Riders may transfer twice. empty - Unlimited transfers are permitted.

agency id	ID referencing agency.agency id	Conditionally Required	Identifies the relevant agency for a fare. This field is required for datasets with multiple agencies defined in agency.txt, otherwise it is optional.
transfer duration	non negative integer	Optional	Length of time in seconds before a transfer expires. When transfers=0 this field can be used to indicate how long a ticket is valid for or it can can be left empty.

- fare rules.txt

Field Name	Type	Required	Description
fare id	ID referencing fare attributes.fare id	Required	Identifies a fare class.
route id	ID referencing routes.route id	Optional	Identifies a route associated with the fare class. If several routes with the same fare attributes exist, create a record in fare rules.txt for each route. Example: If fare class ”b” is valid on route ”TSW” and ”TSE”, the fare rules.txt file would contain these records for the fare class: fare id,route id b,TSW b,TSE
origin id	ID referencing stops.zone id	Optional	Identifies an origin zone. If a fare class has multiple origin zones, create a record in fare rules.txt for each origin id. Example: If fare class ”b” is valid for all travel originating from either zone ”2” or zone ”8”, the fare rules.txt file would contain these records for the fare class: fare id,...,origin id b,...,2 b,...,8

destination id	ID referencing stops.zone id	Optional	Identifies a destination zone. If a fare class has multiple destination zones, create a record in fare rules.txt for each destination id. Example: The origin id and destination id fields could be used together to specify that fare class ”b” is valid for travel between zones 3 and 4, and for travel between zones 3 and 5, the fare rules.txt file would contain these records for the fare class: fare id,...,origin id,destination id b,...,3,4 b,...,3,5
contains id	ID referencing stops.zone id	Optional	Identifies the zones that a rider will enter while using a given fare class. Used in some systems to calculate correct fare class. Example: If fare class ”c” is associated with all travel on the GRT route that passes through zones 5, 6, and 7 the fare rules.txt would contain these records: fare id,route id,...,contains id c,GRT,...,5 c,GRT,...,6 c,GRT,...,7 Because all contains id zones must be matched for the fare to apply, an itinerary that passes through zones 5 and 6 but not zone 7 would not have fare class ”c”.

- shapes.txt

Field Name	Type	Required	Description
shape id	ID	Required	Identifies a shape.
shape pt lat	Latitude	Required	Latitude of a shape point. Each record in shapes.txt represents a shape point used to define the shape.
shape pt lot	Longitude	Required	Longitude of a shape point.

shape pt sequence	Non negative integer	Required	Sequence in which the shape points connect to form the shape. Values must increase along the trip but do not need to be consecutive. Example: If the shape ”A shp” has three points in its definition, the shapes.txt file might contain these records to define the shape: shape id,shape pt lat,shape pt lon,shape pt sequence A shp,37.61956,-122.48161,0 A shp,37.64430,-122.41070,6 A shp,37.65863,-122.30839,11
shape dist traveled	Non negative float	Optional	Actual distance traveled along the shape from the first shape point to the point specified in this record. Used by trip planners to show the correct portion of the shape on a map. Values must increase along with shape pt sequence; they cannot be used to show reverse travel along a route. Distance units must be consistent with those used in stop times.txt. Example: If a bus travels along the three points defined above for A shp, the additional shape dist traveled values (shown here in kilometers) would look like this: shape id,shape pt lat,shape pt lon,shape pt sequence,shape dist traveled A shp,37.61956,-122.48161,0,0 A shp,37.64430,-122.41070,6,6.8310 A shp,37.65863,-122.30839,11,15.8765

- frequencies.txt

Field Name	Type	Required	Description
trip id	ID referencing trips.trip id	Required	Identifies a trip to which the specified headway of service applies.

start time	Time	Required	Time at which the first vehicle departs from the first stop of the trip with the specified headway.
end time	Time	Required	Time at which service changes to a different headway (or ceases) at the first stop in the trip.
headway secs	Non negative integer	Required	Time, in seconds, between departures from the same stop (headway) for the trip, during the time interval specified by start time and end time. Multiple headways for the same trip are allowed, but may not overlap. New headways may start at the exact time the previous headway ends.
exact times	Enum	Optional	Indicates the type of service for a trip. See the file description for more information. Valid options are: 0 or empty - Frequency-based trips. 1 - Schedule-based trips with the exact same headway throughout the day. In this case the end time value must be greater than the last desired trip start time but less than the last desired trip start time + headway secs.

- transfers.txt

Field Name	Type	Required	Description
from stop id	ID referencing stops.stop id	Required	Identifies a stop or station where a connection between routes begins. If this field refers to a station, the transfer rule applies to all its child stops.
to stop id	ID referencing stops.stop id	Required	Identifies a stop or station where a connection between routes ends. If this field refers to a station, the transfer rule applies to all child stops.

transfer type	Enum	Required	Indicates the type of connection for the specified (from stop id, to stop id) pair. Valid options are: 0 or empty - Recommended transfer point between routes. 1 - Timed transfer point between two routes. The departing vehicle is expected to wait for the arriving one and leave sufficient time for a rider to transfer between routes. 2 - Transfer requires a minimum amount of time between arrival and departure to ensure a connection. The time required to transfer is specified by min transfer time. 3 - Transfers aren't possible between routes at the location. 4 - Passengers can stay onboard the same vehicle to transfer from one trip to another (an "in-seat transfer"). 5 - In-seat transfers aren't allowed between sequential trips. The passenger must alight from the vehicle and re-board.
min transfer time	Non negative integer	Optional	Amount of time, in seconds, that must be available to permit a transfer between routes at the specified stops. The min transfer time should be sufficient to permit a typical rider to move between the two stops, including buffer time to allow for schedule variance on each route.

- pathways.txt

Field Name	Type	Required	Description

pathway id	ID	Required	The pathway id field contains an ID that uniquely identifies the pathway. The pathway id is used by systems as an internal identifier of this record (e.g., primary key in database), and therefore the pathway id must be dataset unique. Different pathways can go from the same from stop id to the same to stop id. For example, this happens when two escalators are side by side in opposite direction, or when a stair is nearby and elevator and both go from the same place to the same place.
from stop id	ID referencing stops.stop id	Required	Location at which the pathway begins. It contains a stop id that identifies a platform, entrance/exit, generic node or boarding area from the stops.txt file.
to stop id	ID referencing stops.stop id	Required	Location at which the pathway ends. It contains a stop id that identifies a platform, entrance/exit, generic node or boarding area from the stops.txt file.

pathway mode	Enum	Required	Type of pathway between the specified (from stop id, to stop id) pair. Valid values for this field are: • 1: walkway • 2: stairs • 3: moving sidewalk/travelator • 4: escalator • 5: elevator • 6: fare gate (or payment gate): A pathway that crosses into an area of the station where a proof of payment is required (usually via a physical payment gate). Fare gates may either separate paid areas of the station from unpaid ones, or separate different payment areas within the same station from each other. This information can be used to avoid routing passengers through stations using shortcuts that would require passengers to make unnecessary payments, like directing a passenger to walk through a subway platform to reach a busway. • 7: exit gate: Indicates a pathway exiting an area where proof-of-payment is required into an area where proof-of-payment is no longer required.
is bidirectional	Enum	Required	Indicates in which direction the pathway can be used: • 0: Unidirectional pathway, it can only be used from from stop id to to stop id. • 1: Bidirectional pathway, it can be used in the two directions. Fare gates (pathway mode=6) and exit gates (pathway mode=7) cannot be bidirectional.
length	Non negative Float	Optional	Horizontal length in meters of the pathway from the origin location (defined in from stop id) to the destination location (defined in to stop id). This field is recommended for walkways (pathway mode=1), fare gates (pathway mode=6) and exit gates (pathway mode=7).

traversal time	Positive Integer	Optional	Average time in seconds needed to walk through the pathway from the origin location (defined in from stop id) to the destination location (defined in to stop id). This field is recommended for moving sidewalks (pathway mode=3), escalators (pathway mode=4) and elevator (pathway mode=5).
stair count	Non null Integer	Optional	Number of stairs of the pathway. Best Practices: one could use the approximation of 1 floor = 15 stairs to generate approximative values. A positive stair count implies that the rider walk up from from stop id to to stop id. And a negative stair count implies that the rider walk down from from stop id to to stop id. This field is recommended for stairs (pathway mode=2).
max slope	Float	Optional	Maximum slope ratio of the pathway. Valid values for this field are: • 0 or (empty): no slope. • A float: slope ratio of the pathway, positive for upwards, negative for downwards. This field should be used only with walkways (pathway type=1) and moving sidewalks (pathway type=3). Example: In the US, 0.083 (also written 8.3 percent) is the maximum slope ratio for hand-propelled wheelchair, which mean an increase of 0.083m (so 8.3cm) for each 1m.
min width	Positive Float	Optional	Minimum width of the pathway in meters. This field is highly recommended if the minimum width is less than 1 meter.

signposted as	Text	Optional	String of text from physical signage visible to transit riders. The string can be used to provide text directions to users, such as 'follow signs to '. The language text should appear in this field exactly how it is printed on the signs - it should not be translated.
reversed signposted as	Text	Optional	Same than the signposted as field, but when the pathways is used backward, i.e. from the to stop id to the from stop id.

- levels.txt

Field Name	Type	Required	Description
level id	ID	Required	Id of the level that can be referenced from stops.txt.
level index	Float	Required	Numeric index of the level that indicates relative position of this level in relation to other levels (levels with higher indices are assumed to be located above levels with lower indices). Ground level should have index 0, with levels above ground indicated by positive indices and levels below ground by negative indices.
level name	Text	Optional	Optional name of the level (that matches level lettering/numbering used inside the building or the station). Is useful for elevator routing (e.g. "take the elevator to level "Mezzanine" or "Platforms" or "-1").

- feed info.txt

Field Name	Type	Required	Description
feed publisher name	Text	Required	Full name of the organization that publishes the dataset. This might be the same as one of the agency.agency name values.

feed publisher url	URL	Required	URL of the dataset publishing organization's website. This may be the same as one of the agency.agency url values.
feed lang	Language code	Required	Default language for the text in this dataset. This setting helps GTFS consumers choose capitalization rules and other language-specific settings for the dataset. To define another language, use the language field in translations.txt. Multilingual datasets might be the default language with the original text in multiple languages. In such cases, use the ISO 639-2 language code mul in the feed lang field. Provide a translation for each of the languages used in the dataset in translations.txt. If all of the original text in the dataset is in the same language, don't use mul. For example, a dataset in Switzerland might set the original stops.stop name field populated with stop names in different languages. Each stop name is written in accordance with the dominant language in that stop's geographic location. Stop names include Genève for the French-speaking city of Geneva, Zürich for the German-speaking city of Zurich, and Biel/Bienne for the bilingual city of Biel/Bienne. Set feed lang=mul and provide the following translations in translations.txt: German: "Genf," "Zürich," and "Biel" French: "Genève," "Zurich," and "Bienne" Italian: "Ginevra," "Zurigo," and "Bienna" English: "Geneva," "Zurich," and "Biel/Bienne"

default lang	language code	Optional	Defines the language used when the data consumer doesn't know the language of the rider. It's often defined as en, English.
feed start date	Date	Optional	The dataset provides complete and reliable schedule information for service in the period from the beginning of the feed start date day to the end of the feed end date day. Both days can be left empty if unavailable. The feed end date date must not precede the feed start date date if both are given. Dataset providers are encouraged to give schedule data outside this period to advise of likely future service, but dataset consumers should treat it mindful of its non-authoritative status. If feed start date or feed end date extend beyond the active calendar dates defined in calendar.txt and calendar dates.txt, the dataset is making an explicit assertion that there is no service for dates within the feed start date to feed end date range but not included in the active calendar dates.
feed end date	Date	Optional	Refer to the feed start date row in this table.
feed version	Text	Optional	String that indicates the current version of their GTFS dataset. GTFS-consuming applications can display this value to help dataset publishers determine whether the latest dataset has been incorporated.
feed contact email	Email	Optional	Email address for communication regarding the GTFS dataset and data publishing practices. feed contact email is a technical contact for GTFS-consuming applications. Provide customer service contact information through agency.txt.

feed contact url	URL	Optional	URL for contact information, a web-form, support desk, or other tools for communication regarding the GTFS dataset and data publishing practices. feed contact url is a technical contact for GTFS-consuming applications. Provide customer service contact information through agency.txt.

- translations.txt

Field Name	Type	Required	Description
table name	Enum	Required	Defines the dataset table that contains the field to be translated. The following values are allowed: agency, stops, routes , trips, stop times, feed info, pathways, levels, attributions
field name	Text	Required	Provides the name of the field to be translated. Fields with the type ”Text” can be translated, while fields with the types ”URL,” ”Email,” and ”Phone number” can be included here to provide those resources in the correct language.
language	Language code	Required	Provides the language of translation. If this language is the same as the one from feed lang in feed info.txt, the original value of the field is the default value used in languages without specific translations.For example, in Switzerland, a city in a bilingual canton is officially called ”Biel/Bienne,” but it would simply be called ”Bienne” in French and ”Biel” in German.
translation	Text, URL, Email, or Phone number	Required	Provides the translated value for the specified field name.

record id	ID	Conditionally Required	Defines the record that corresponds to the field to be translated. The value in record id needs to be a main ID from a dataset table, as defined in the following table: The following conditions determine how this field can be used: Forbidden if table name equals feed info. Forbidden if field value is defined. Required if field value is empty.
record sub id	ID	Conditionally Required	Helps to translate the record that contains the field when the table referenced in record id doesn't have a unique ID. The value in record sub id is the secondary ID of a dataset table, as defined in the following table: The following conditions determine how this field can be used: Forbidden if table name equals feed info. Forbidden if field value is defined. Required if table name equals stop times and record id is defined.

field value	Text, URL, Email, or Phone number	Conditionally Required	Instead of using record id and record sub id to define which record needs to be translated, field value can be used to define the value for translation. When used, the translation is applied when the field identified by table name and field name contains the exact same value defined in field value. The field must exactly match the value defined in field value. If only a subset of the value matches field value, the translation isn't applied. If two translation rules match the same record, one with field value and the other one with record id, then the rule with record id is the one that needs to be used. The following conditions determine how this field can be used: Forbidden if table name equals feed info. Forbidden if record id is defined. Required if record id is empty.

- attributions.txt

Field Name	Type	Required	Description
attribtion ID	ID	Optional	Identifies an attribution for the dataset, or a subset of it. This field is useful for translations.
agency id	ID referencing	Optional	The agency to which the attribution applies. If one agency id, route id, or trip id attribution is defined, the other fields must be empty. If none are specified, the attribution applies to the whole dataset.
route id	ID referencing	Optional	This field functions in the same way as agency id, except the attribution applies to a route. Multiple attributions can apply to the same route.

trip id	ID referencing	Optional	This field functions in the same way as agency id, except the attribution applies to a trip. Multiple attributions can apply to the same trip.
organization name	Text	Required	The name of the organization that the dataset is attributed to.
is producer	Enum	Optional	The role of the organization is producer. Allowed values include the following: • 0 or empty: Organization doesn't have this role. • 1: Organization does have this role. At least one of the fields, either is producer, is operator, or is authority, must be set at 1.
is operator	Enum	Optional	Functions in the same way as is producer, except the role of the organization is operator.
is authority	Enum	Optional	Functions in the same way as is producer, except the role of the organization is authority.
attribution url	URL	Optional	The URL of the organization.
attribution email	Email	Optional	The email of the organization.
attribution phone	Phone number	Optional	The phone number of the organization.

3.5 Points of interest

- POI.json

Field Name	Type	Required	Description
id	Text	Required	Identifier of the point of interest.
domainId	Text	Required	Identification code of the entity in the domain 'it.trentour.domains.core.PointOfInterest'
type	Text	Required	Principal domain where the element can be found
allTypes	List	Required	All the domain where the element can be found
content	Dictionary	Required	Provides information about the POI(Check tables below).

- content key in POI.json

Field Name	Type	Required	Description
poiData	Dictionary	Required	Provides data about the point of interest represented.
objData	Dictionary	Required	Provides data about the object represented.

- poiData key in content key in POI.json

Field Name	Type	Required	Description
poiId	Text	Required	Identification code.
location	Dictionary	Required	Contains the elements to identify the location in a map. Among the information an be found: latitude, longitude, altitude and the name of the reference system used (srsName).
addresses	Dictionary	Required	Each key states the language of the address, the value of each key is the direction of the place defined by: country, state, region, city, postalCode and street.
contact	dictionary	Required	Defines different options to contact the POI. Inside the dictionary, the following information can be found: A dictionary with the information of the people responsible, a phone number, a fax, a mail and an url.
seating Capacity	Number	Optional	Provides information regarding the maximum amount of seats that can be found in the POI.
accessibility	Text	Optional	Provides information about the accessibility in the POI.
timetable	Dictionary	Optional	Provides information about the time schedule of the place. Each key represents the language in which the information is written.
price	Dictionary	Optional	Provides information about the prices to access the place. Each key represents the language in which the information is written.

- objData key in content key in POI.json

Field Name	Type	Required	Description
objectUri	Text	Required	Provides a way to identify the object.
entityUri	Text	Required	Provides a way to identify the enity.
name	Dictionary	Required	Provides a dictionary in which the key represents the language and the value the name that the POI receives in the language of the key.
description	Dictionary	Required	Provides a dictionary in which the key represents the language and the value a description of the POI in the language of the key.
long Description	Dictionary	Optional	Provides a dictionary in which the key represents the language and the value a long description of the POI in the language of the key.
service Description	Text	Optional	States the type of service that the POI offers (museum, stadium...).
topics	List	Optional	Provides a list with the activities that can be performed in the POI.
source	Dictionary	Optional	Provides information about the origin of the data.
images	List	Optional	Provides the url of images that belong to the POI.

4 Purpose Formalization

The Purpose formalization section has to report:

- **Scenarios**: a set of usage scenarios, describing examples of usage of the Purpose.
- **Personas**: a set of real users examples acting within the scenarios defined above. Each Persona is defined over a specific features included in the main Purpose.
- **Competency Questions (CQs)**: the list of CQs created considering the personas in the scenarios defined.
- **Entities identified**: the terms representing the entities to consider in the KGE project, classified using the popularity categories.

4.1 Scenarios

For this Knowledge Graph the following scenarios were created:

1. A local that will be in the area for a long period and wants to visit an important landmark in Trentino every week.

2. A tourist that will be in the area for a short period of time and wants to visit as many landmarks as possible.

3. A tourist wants to visit a specific place as early as possible in the morning.

4. A student with a limited amount of time that wants to visit something within the time limit.

5. A commuter in a wheelchair wants the path with the least amount of transfers to a specific landmark

6. A retiree that wants to go to a place but just wants to use a specific public transport.

7. A student wants to find the quickest route to a landmark on saturday.

4.2 Personas

For this Knowledge Graph the following fictive personas were created.

Name	Age	Interests	Usage	Description
John	23	Hiking	once a week	John likes hiking and is interested in easy hikes around the trentino province
Julia	35	Travelling and visiting the most important landmarks of a region	one time	Julia has arrived to the region of Trentino, she will be here for a weekend and she wants to visit as many places as she can
Barnet	65	Spending all the day visiting a place	2 times per month	Barnet loves staying all the day in an area of cultural interest and he wants to be the first one so he can enjoy all the day
Maria	25	Visiting places	1 time per week	Maria lives in the Trentino area and she likes visiting important places in the region. But she has a limited amount of time to visit this places.
Guiseppe	6	knights	once a week	is scared of busses because they are scary

4.3 Competency Questions

To see if the Knowledge Graph is efficient the following competency questions have been created. if they can be answered with the knowledge graph this means that the knowledge graph is competent and thus useful.

Scenario	Persona	Competency Questions
1	John	What can I visit today?
2	Julia	How can I visit the 5 most important landmarks of Trentino in a row?
3	Barnet	How can I arrive to a landmark when they open?
4	Maria	What can I visit, so I will be back in 5 hours?
6	Giuseppe	How can you get to the castle without the bus?
5	Barnet	How can I visit this landmark so I make as less transfers as possible?
7	Maria	Can the landmark be reached by just taking the bus?
1	Maria	Which landmark is the most convenient to visit now?
4	Giuseppe	What can I visit before I get bored?

5 Inception

This section aims to report the KGE sub process performed during the inception phase, by describing each activities both in schema and data layer.

Inception sub activities:

- Resources collection/scraping
- Resources filtering and classification over common, core and contextual
- Resources knowledge definition/extraction
- Resources formatting (semi-formal transformation)

The report of the work done during the first phase of the methodology, has to includes also the description of the different choices made, with their strong and weak points. In other words the report should provide to the reader, a clear description of the reasoning conducted by all the different team members. Moreover, the difficulties as well as open issues eventually involved in the inception phase sub process, should be reported.

6 Informal Modeling

This section is dedicated to the description of the informal modeling phase. Like in the previous section, it aims to describe the different sub activities performed by all the team members, as well as the phase outcomes produced.

More in details, this section provides a description of the following activities:

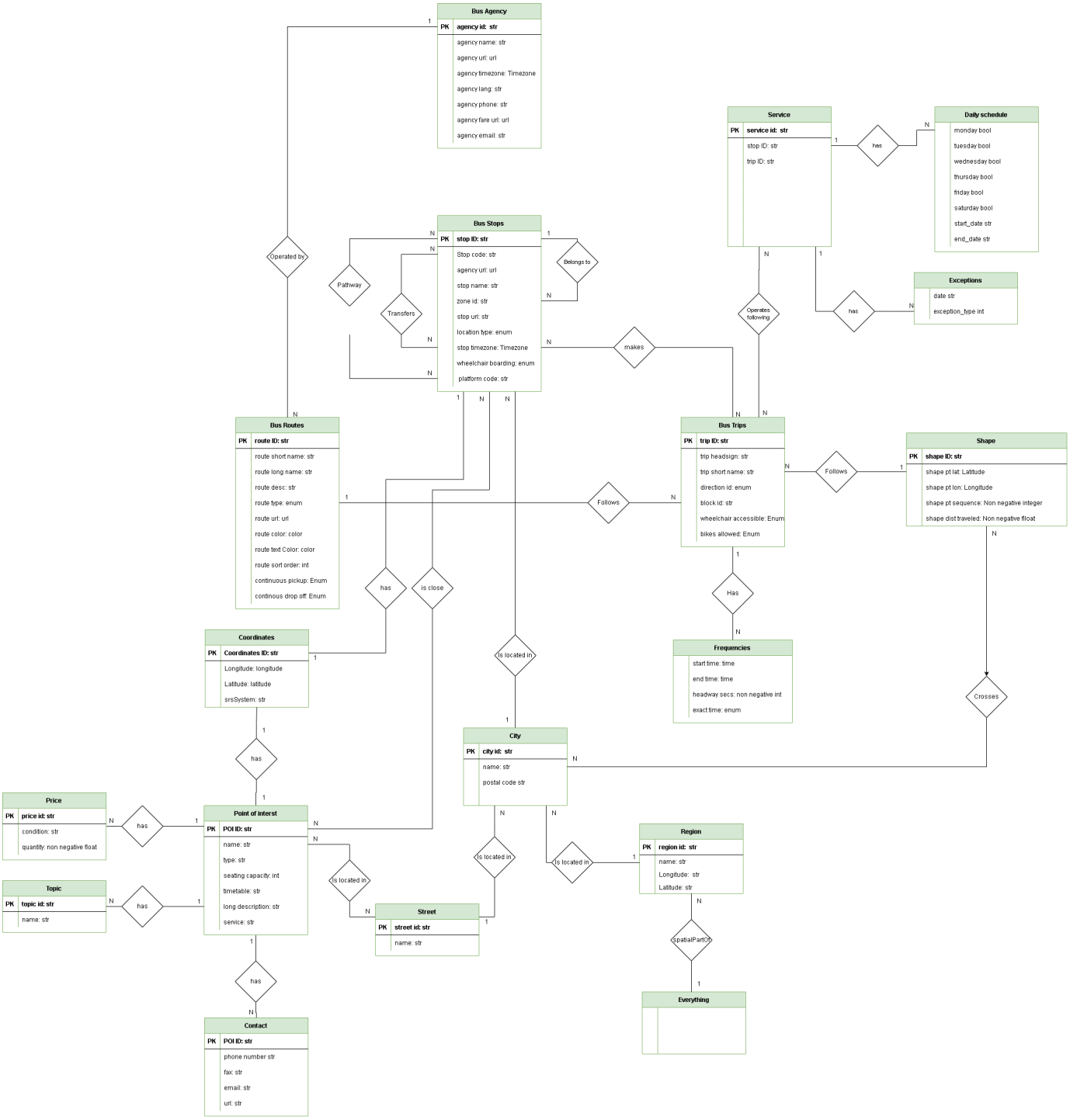
- ER model description. Both the creation process (decisions taken) and the outcome have to be described.
- Teleology building.

- Datasets filtering and alignment with teleology.
- Phase open issues.

Like the previous phase, also the current one has to report the decision made during the phase activities, with the weak and strong point associate to them. If difficulties and/or open issue have been encountered, they should be reported as well.

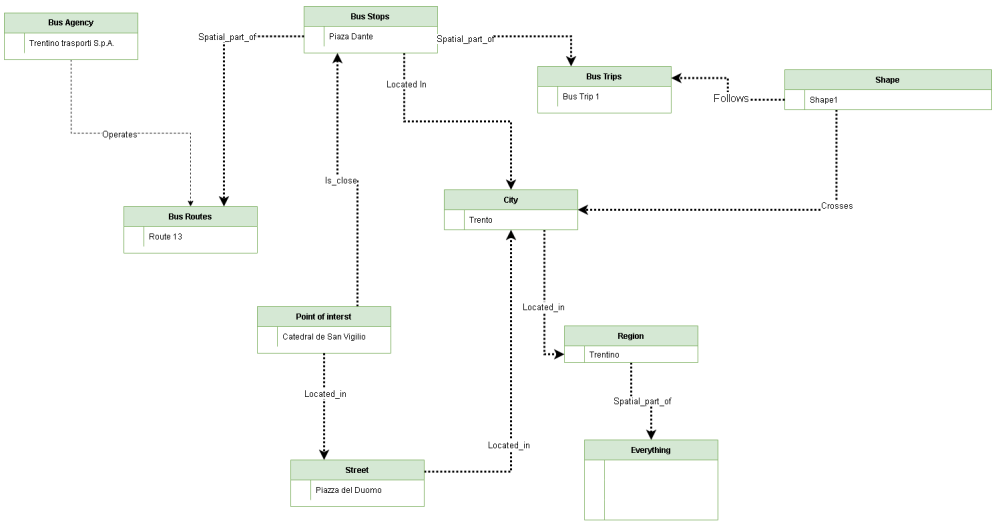
6.1 ER Diagram

Here we can see the full dataset modelled as an ER Diagram.



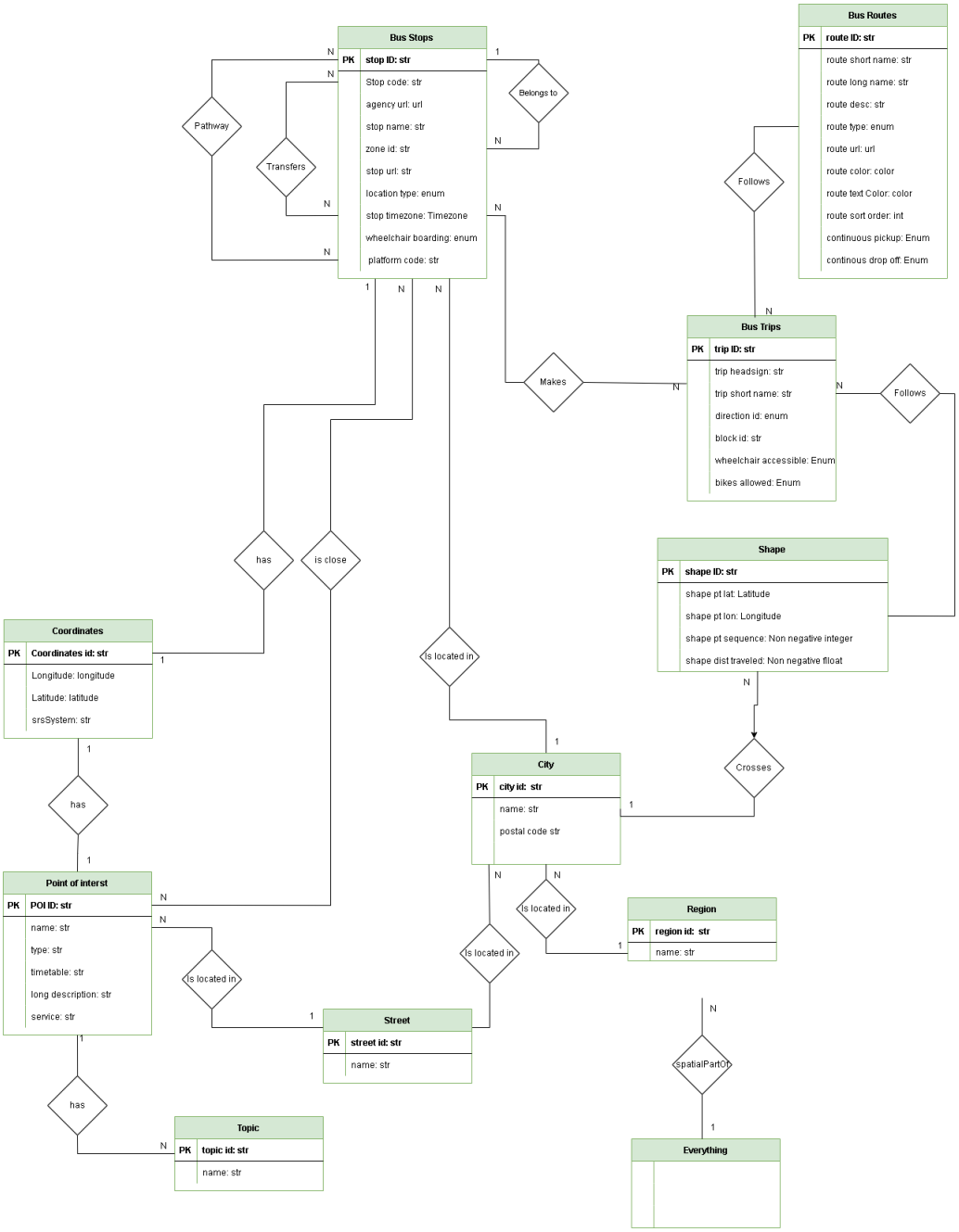
6.2 Teleology

Here we see a Teleology of the dataset in which the question is answered of how to get to the cathedral in Trento, one of the points of interest.



6.3 Data Filtering Process to Allign Teleology

In the above ER Diagram and Teleology it can be seen that quite a bit of the dataset is not needed to answer the competency questions. So A closer look at the ER was taken and the following was concluded: First the full schedule of the busses is not important in our case, just where the next bus arrives and which bus route to take. For this reason the service part of the ER was redundant and removed. Secondly for the points of interest it was not necessary to know the entrance fees or contact information for the point of interest nor to know images about it. Thus these were also removed. The topic of the point of interest was kept was kept since in some cases to answer the competency question the topic of the point of interest is needed. The last removed entity is the bus agency, because for this purpose it is not important who operates a bus route just where the bus is going. This process of eliminating entities let to the following simplified ER Diagram which is more suited for the purpose of this project.



6.4 Open Issues

7 Formal Modeling

This section is dedicate to the description of the formal modeling phase. Like in the previous section, the current one aims to describe the different sub activities performed by all the team members, as well as the phase outcomes produced.

More in details, this section provides a description of the following activities:

- ETG generation
- Language alignment
- Formal data creation, by aligning datasets with the ETG created.
- Phase open issues

Like the previous phase, also the current one has to report the decision made during the phase activities, with the weak and strong point associate to them. If difficulties and/or open issue have been encountered, they should be reported as well.

7.1 ETG generation

In order for the ETG to be generated successfully we need a single definition of language. For this we will use the iTelos tool provided to us. This tool will show us possible definitions of concepts in our ontology and if the right definition isn't there will allow us to create it. Once the ontology is given a language it will be our ETG.

7.2 Language alignment

First of all during the language alignment only 8 concepts were not yet in the KOS concepts and were thus added with the meaning in our case. These concepts are: BelongsTo, IsClose, Transfers, SpatialPartOf, Crosses, IsLocatedIn, and Has. For the other ones we were able to use an already existing concept to describe the issues.

7.3 Formal data creation

After these steps the aligned ETG was downloaded and combined with the datasets, since within the datasets there were no language ambiguities such as the same words with different meaning, this went very smoothly.

7.4 phase open issues

8 KGC

This section is dedicate to the description of the KGC phase. Like in the previous section, the current one aims to describe the different sub activities performed by all the team members, as well as the phase outcomes produced.

More in details, this section provides a description of the following activities:

- Data mapping. The mapping activities describe how the final KG is created. Provide a description of such activities for the datasets considered.
- Entity matching (semantic heterogeneity). Describe how different representation of the same real world objects have been handled.
- Phase open issues

8.1 Data Mapping

After creating the final ETG, we have used the tool Karma in order to map the real entities from our datasets to the proposed Etypes. We have used this tool to perform some variations on the identification codes of the different entities for a better understanding of the later KG. In the following image we can see how the different columns from the dataset stops clean where mapped to the Etype and it's properties.

8.2 Entity Matching

In our case no data matching technique was required. The data sources used where focusing on different parts of the KG. We created a new dataset that contains a pair of PointOfInterest coordinate and BusStop coordinates and a 3rd column stating if they both coordinates where less than 500m apart or not. This new dataset enabled us to merge the two main parts of our KG, the information about the public transport system and the information about the points of interest.

8.3 Open Issues

9 Outcome Exploitation

This section aims to provide a description of the KGE process outcome. Here you have to report the final Knowledge Graph information statistics (like, number of etypes and properties, number of entities for each etype, and so on). Moreover this section has to provide a description for the KG possible exploitation, like examples of queries executed, execution time, and so on.

9.1 Statistics

Below is the table with all the knowledge graph statisctics.

Etype	Nº of entities
Region	1
City	29
Street	197
Points of interest	452
Topics	21
BusStop	3,131
Coordinates	3,106
BusRoutes	125
BusTrips	3,199

Table 21: Knowledge Graph Statistics

9.2 Queries

In this section we show some queries which can retrieve information from our knowledge graph

- Retrieve the names of the cities that are part of the KG.
  Time:0.1s

```
PREFIX ontology:<http://www.semanticweb.org/kasper/
        ntologies/2022/10/StaticOntology#>

select ?city_name where {
    ?City a ontology:City .
    ?City ontology:name ?city_name .
            }
```

- Retrieve POI based on a topic. This query retrieve all the POIs that are not related with a topic.
  Time:0.1s

```
PREFIX ontology:<http://www.semanticweb.org/kasper/ontologies/2022/10/Stat

select distinct ?PointOfInterest where {
    ?PointOfInterest a ontology:PointOfInterest .
        ?PointOfInterest ontology:Has ?Topic .
    FILTER NOT EXISTS {?Topic ontology:name "arte".}
}
```

- Retrieve POI based on a city. This query helps to organize trips to a specific city.
  Time: 0.1s

```
PREFIX ontology:<http://www.semanticweb.org/kasper/
    ontologies/2022/10/StaticOntology#>

select ?POI where {
    ?City a ontology:City .
```

```
    ?Street a ontology:Street .
    ?POI a ontology:PointOfInterest .

    ?City ontology:name "CEMBRA" .
    ?Street ontology:IsLocatedIn ?City .
    ?POI ontology:IsLocatedIn ?Street .
}
```

- Retrieve Bus stops that are close to a POI. This query retrieves the busStops we could be looking for when wanting to search for a path to a specific Point of Interest.
  Time: 0.1s

```
PREFIX ontology:<http://www.semanticweb.org/kasper/
    ontologies/2022/10/StaticOntology#>

SELECT Distinct ?BusStop WHERE {
    ?PointOfInterest a ontology:PointOfInterest .
    ?BusStop a ontology:BusStop .
    ?Coordinates a ontology:Coordinates .
    ?Coordinates ontology:IsClose ?BusStop .
    ?PointOfInterest ontology:Has ?Coordinates .
    ?PointOfInterest ontology:name
        "ItinerarioRiservaNaturalisticadelLaghestel" .
}
```

- Retrieve Bus Routes close to a given POI. This query identifies the BusRoutes that could be use in order to arrive to a given Point of Interest.
  Time: 4.2s

```
PREFIX ontology:<http://www.semanticweb.org/kasper/
    ontologies/2022/10/StaticOntology#>

select distinct ?BusRoutes where {
        ?BusRoutes a ontology:BusRoutes .
    ?BusTrips a ontology:BusTrips .
    ?BusStops a ontology:BusStop .
    ?POICoordinates a ontology:Coordinates .
    ?BSCoordinates a ontology:Coordinates .
    ?PonitOfInteres a ontology:PointOfInterest .

    ?BusStop ontology:Has ?BSCoordinates .
    ?POICoordinates ontology:IsClose ?BSCoordinates .
    ?PointOfInterest ontology:Has ?POICoordinates .

    ?BusTrips ontology:Follows ?BusStops .
```

```
    ?BusTrips ontology:Follows ?BusRoutes .
    ?PointOfInterest ontology:name "AGRITURAISERCI" .
}
```

- Retrieve Bus Routes from a known bus stop. This query identifies the BusRoutes that could be taken from a given BusStop.
  Time:0.1s

```
PREFIX ontology:<http://www.semanticweb.org/kasper/
    ontologies/2022/10/StaticOntology#>

select distinct ?BusRoutes where {
    ?BusRoutes a ontology:BusRoutes .
    ?BusTrips a ontology:BusTrips .
    ?BusStop a ontology:BusStop .

    ?BusStop ontology:stop_code "1085" .
    ?BusTrips ontology:Follows ?BusStop .
    ?BusTrips ontology:Follows ?BusRoutes .
}
```

- Retrieve BusStops from a known bus route. This query identifies the BusStops from a known BusRoute.
  Time: 0.1s

```
PREFIX ontology:<http://www.semanticweb.org/kasper/
    ontologies/2022/10/StaticOntology#>

select distinct ?PointOfInterest where {
    ?BusRoutes a ontology:BusRoutes .
    ?BusTrips a ontology:BusTrips .
    ?BusStop a ontology:BusStop .
    ?POICoordinates a ontology:Coordinates .
    ?BSCoordinates a ontology:Coordinates .
    ?PointOfInterest a ontology:PointOfInterest .

    ?BusTrips ontology:Follows ?BusStop .
    ?BusTrips ontology:Follows ?BusRoutes .
    ?BusStop ontology:Has ?BSCoordinates .
    ?POICoordinates ontology:IsClose ?BSCoordinates .
    ?PointOfInterest ontology:Has ?POICoordinates .

    ?BusRoutes ontology:route_short_name "B114" .
}
```

- Retrieve if a known Point of Interest is accessible through a BusRoute. This query will return True or false to the question: Does the POI with name ”x” is accessible through the route with short name ”y”?
  Time:0.1s

```
PREFIX ontology:<http://www.semanticweb.org/kasper/
    ontologies/2022/10/StaticOntology#>

ASK {
    ?BusRoutes a ontology:BusRoutes .
    ?BusTrips a ontology:BusTrips .
    ?BusStop a ontology:BusStop .
    ?POICoordinates a ontology:Coordinates .
    ?BSCoordinates a ontology:Coordinates .
    ?PointOfInterest a ontology:PointOfInterest .

    ?BusTrips ontology:Follows ?BusStop .
    ?BusTrips ontology:Follows ?BusRoutes .
    ?BusStop ontology:Has ?BSCoordinates .
    ?POICoordinates ontology:IsClose ?BSCoordinates .
    ?PointOfInterest ontology:Has ?POICoordinates .

    ?PointOfInterest ontology:name ”ItinerarioRiservaNaturalisticadelLaghe
    ?BusRoutes ontology:route_short_name ”B114” .
}
```

A visual representation from the KGE can be seen in 1 where we see the link between the entity and the describtion of the entity.

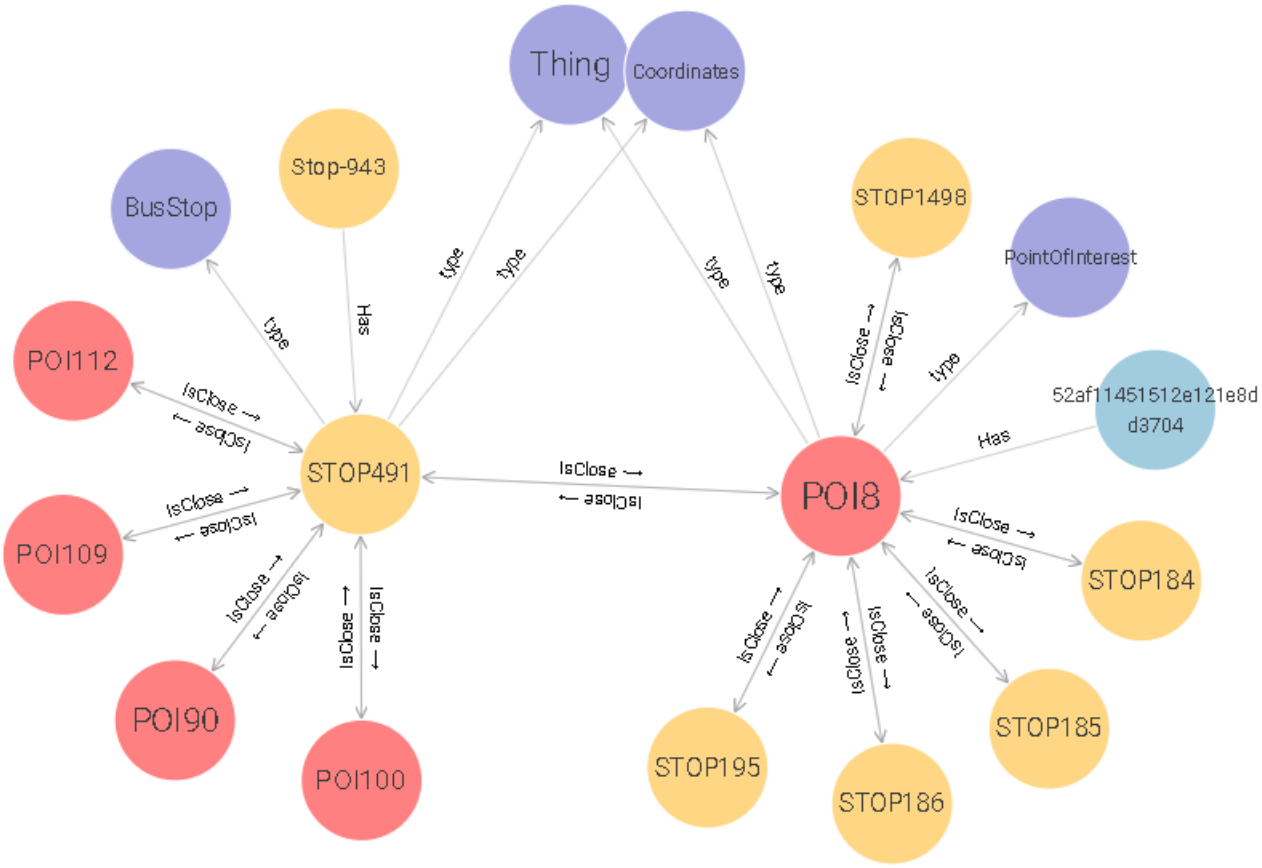


Figure 1: A Small part of the formal model visualized in GrapDB

10 Conclusions & Open Issues

This section concludes the current document with final conclusions regarding the quality of the process and final outcome:

- Did the project respect the scheduling expected in the beginning ?
- Are the final results able to satisfy the initial Purpose ?
  - If no, or not entirely, why ? which parts of the Purpose have not been covered ?

Moreover, this section aims to summarize the most relevant issues/problems remained open along the KGE process. The description of open issues has to provide a clear explanation about the problems, the approaches adopted while trying to solve them and, eventually, any proposed solution that has not been applied.

10.1 Final Results

10.1.1 Scheduling

During the project all intermediate deadlines were met. After which the feedback was implemented during the work on the next steps. For the final phase we were unable to hand in the project on an earlier date then the exam date. This is mainly caused by inexperience with the tools such as GraphDB.

10.1.2 Purpose fulfillment

Unfortunately the final results do not satisfy the initial Purpose fully because the KGE is not able to answer all competency questions successfully fully or partially in some cases. This is mainly due to two things missing from the KGE. Which are the times and transfers, with times being a broad concept covering both the travel times of the busses as well as the full timetables of the points of interest. The transfers in a specific bus station are also not directly included and for the KGE to be able to locate them would need to be added explicitly.
For more details on what is missing on each question see the Competency Questions part in Open Issues below.

we can say if there is a route just not the fastest.

10.2 Open Issues

10.2.1 Competency Questions

As mentioned in the Final Results above not all competency questions are answerable with out KGE. For this reason we will go over the questions now and discuss what is missing from the KGE for the question to be answered.

- What can I visit Today? For this question we are missing the current time as an input. Besides that the landmark timetables are included in the KGE but consists of mostly empty values so no real conclusions can be drawn from that. Finally for this question we would also need to consider travel time towards the POI which is missing from the KGE.
- How can I visit the 5 most important landmarks of Trentino in a row? For this there needs to be a scale of importance linked to the landmarks. the visiting in a row is possible if all landmarks are directly linked with a bus line and don't need transfers as the KGE is unable to look up transfers.
- How can I arrive to a landmark when they open? For this question we have the same issue as before that we do not reliably have the opening times of the landmarks. However if the landmark timetable is filled with proper datetime data and the trip requires no transfers the KGE will be able to answer it.
- What can I visit so I will be back in 5 hours? For this question we need travel time and time needed at the POI which could be the average time spent at this place which can be found on google maps.

- How can I get to the castle without the bus? Currently the KGE only includes busses so for this question to be answered outher modes of transport such as trains should be included.
- How can I visit this landmark so I make the least transfers possible? We can answer this if it is reachable without transfers, however if this is not possible the transfer information is needed.
- Can the landmark be reached by just taking the bus? If no transfers are needed this question can be answered.
- Which landmark is the most convenient to visit now? If the landmark is on a busroute connected to the current location (which we assume is a busstop or near one) this is answerable. However the definition of convenient can differ per persona so it will always be difficult to answer this question.
- What can I visit before I get bored? For this we would need the persona's intrests which are not included in the dataset.