Dipartimento di Ingegneria e Scienza dell'Informazione

– KnowDive Group –

KDI 2021 - Project Report

Document Data:	Reference Persons:
December 20, 2021	Giuliano Andronic, Diane Willaime

KnowDive (internal) reports are for internal only use within the KnowDive Group. They describe preliminary or instrumental work which should not be disclosed outside the group. KnowDive reports cannot be mentioned or cited by documents which are not KnowDive reports. KnowDive reports are the result of the collaborative work of members of the KnowDive group. The people whose names are in this page cannot be taken to be the authors of this report, but only the people who can better provide detailed information about its contents. Official, citable material produced by the KnowDive group may take any of the official Academic forms, for instance: Master and PhD theses, DISI technical reports, papers in conferences and journals, or books.

Index:

Chapter 1

Introduction

Reusability is one of the main principles in the Data Integration (DI) process defined by iTelos. The data integration project documentation plays an important role in order to enhance the reusabiltiy of the resources handled during the methodology, as well as for the resources produced by the data integration process. A clear description of the resources and the process that has to manage them, provides a clear understanding of the information handled in the DI project, allowing external readers to exploit the same resources in different projects.

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

- Section 1: The project's purpose and the resources involved (both schema and data resources) in the integration process.
- Section 2, 3, 4, 5: The integration process along the iTelos phases.
- Section 6: How the result of the integration process (KGs) can be exploited.

Paris is a city offering so many opportunities for any kind of activity for hundreds of thousands of tourists visiting it every year. For many of them it can be difficult to move through the many streets of the city and find the most interesting places and things to do. They often rely on travel guides they can find on the internet, but those guides don't always reflect the interests of the tourists. What if there was an app that could help tourists properly organize their days in Paris based exactly on their interests and time/commuting availability? This Report will describe the process we will follow to create a knowledge graph able to help tourists find the activities perfectly suited for them.

Chapter 2

Purpose and project's resources

2.1 Project's purpose

2.1.1 Project's domain

The DoI of this project is the city of Paris and its different kinds of tourists, this can also be expanded to Parisians who during vacation or free days are looking for what the city has to offer for them.

For example, you have 2 hours to spare in Paris, you don't know what to do: the model, given your position, the time slot and your preferred means of transportation, will tell you what are the points of interest close to you. So, we will need to focus on gathering data mostly of "Touristic" interest: monuments, historical buildings, museums, events, restaurants, etc. Besides this kind of data, as you can notice in the previous example, we will need access to the location of the users and of the places they're interested into. So that we can suggest them the closest ones.

When possible, additional data regarding the time tables of the places can be used to give the users pertinent results with respect to their time preferences.

2.1.2 Personas and Scenarios

In the following list you will find 8 personas of our Domain of Interest who have interest in exploiting the Data Integration project results. The common factor between these stereotyped users is the need to find and reach one or many interest points within the city of Paris. The use case scenarios are embedded within the persona description.

- **Clark Nuckerberg :** Clark is a 51-year-old American CEO going on a business trip in Germany. He must stop in Paris to switch planes and has 6 hours to spare. He has never visited Paris so he's very excited to spend a few hours there and see the most famous places. Since his travel expenses are completely covered by his company, he wants to take a taxi to travel the city in the most comfortable and secure way. Indeed, he carries his personal laptop and important business files with him, so the pickpockets in crowded streets must be avoided at all costs. Clark's passion is art and history, so he would like to find a museum which is open during his 6-hour stop and doesn't take more than 1 hour to visit. He also has to make an important Zoom call, so he would like to stop in a coffee shop where there is free wifi and a quiet environment.

- **Lilian Bidou :** Lilian is a 25 years old french tourist who is spending one week in Paris with his girlfriend. She doesn't know it, but he plans on proposing during this holiday. Lilian needs help to find romantic locations to create the best possible experience for his soon-to-be fiancée. He spend all his money on the ring, so he is on a tight budget and will go to parks or free-entry events rather than museums. He also would love to try tandem with his girlfriend, but he needs to find a safe bicycle track with a low speed limit because she doesn't know how to ride a bike !
- **Isabella Massini :** Isabella is a 47-year-old Italian nurse, who moved to a small city one hour away from Paris, two years ago. She loves religious history and spending her free time with her husband and their two children (6 and 9 years old). She occasionally visits Paris with them during the weekend, mostly to visit museums or participate in Christian community events. Isabella is very environmentally conscious, and tries as much as possible to use bikes instead of busses when she travels alone inside Paris. She wants to buy an electrical bike but is not sure if it is worth the money, so when she is in the city she rents electric bikes if they are available, to try them out. During her family cultural trips, she has to take the car because of her children. But she is not really familiar with the car parking areas in the city since she always use bikes. Isabella moved in a small village near Paris because she doesn't like the noise and crowed streets of the capital. This is why, when she visits museums, she wants to buy tickets as close as possible of opening hour or closing hour.
- **Ginno Zebest :** Gino is a 27 years old full-stack programmer in freelance. His business is working really well and he plans on buying a flat in Paris to have an permanent office and greet clients. Since this is a heavy investment, he needs to make sure that the location of the flat meets his criteria. Ginno doesn't have a car, but if there are parking area around it will be easier for clients to reach him. Also, he loves to party so he definitely needs to know if there are bars and nightclubs in his neighborhood. He also wants to make sure that the basic necessities (pharmacy, grocery store, bakery, ...) are in a 15 minute walking range.
- **Brigitte Norcam :** Brigitte is a loving 83 year-old grand-mother having her grand-children over one week-end per month. Even though Brigitte wants to do a lot of activities with her grand-children, she is quickly tired and needs to limit as much as possible the traveling time. So she will always select the closest alternative to her flat, in order to be able to go home to rest if needed.
- **Martina Rossi :** Martina is a 38-years-old elementary school teacher. For a school trip, Marina and her pupils are spending one day in Paris. She needs to plan the day to make sure everything goes smoothly. One of her pupil is in a wheelchair, so she needs to make sure that the proposed activities have a PMR access. This is also a constraint to take into account when taking public transports or going into a restaurant for lunch. She will usually aim for quiet places like parks or museums because the noise and the crowd can be overwhelming for some of her pupils.

- **Marie Dubois :** Marie is a 19-year-old Erasmus student in Beauvais. She is from a small town in Roumania and enjoys going to Paris with her friends during the week-end. She loves music festivals and tries to participate in as many cultural events as possible to make the most of her stay in France. Marie's other passion is shopping, but since she is a student, she needs to be very smart with the way she spends money. For this reason she usually prefers walking instead of taking public transports, and always looks for student discounts at events. When Marie and her friend decide to go shopping, they plan their itinerary in advance to save time and energy.
- **Tommaso Sopa :** He's a 31-year-old travel-blogger with a passion for food and original recipes. He travels the world looking for inspiration for his book on culinary arts. Tommaso is a real foodie and when he goes in a restaurant, he wants to taste authentic local cuisine. Usually, when he travels, he selects areas famous for their traditional dishes and spends around a month there. During his stay, he will try every single restaurant in the area and write comments about them on the Internet. He also goes to food festivals to meet people with the same passion as him and maybe get invited to share a homemade meal. Next month he will be in Paris, and will of course explore the restaurants of every district.

2.2 Knowledge resources

In order to satisfy the project's purpose, we selected the reference ontology from schema.org (click here for the link). Schema.org aims to create, maintain and promote schemes for structured data on the Internet. It enables webmasters to use a shared vocabulary so that their pages can be understood by search engines. We chose it because it is the most general (and thus the most flexible) reference ontology we found. It is able to fit to our needs and features one particular aspect that many ontologies are missing : the conceptualization is shared by a extremely large community. This is helpful to find forums and channels for questions and contributions. This tool has been created by developers for developers and maximizes transparency by providing clear pointers of provenance.

2.3 Data resources

The datasets source considered is OpenData-Paris, a web portal offering rich and complete datasets collected by the city of Paris (click here for the link). Those data are open in a spirit of transparency and innovation, and to encourage the citizens to be active and co-designers of the city's evolution. We will focus on the datasets that can be related to tourism and leisure activities.

- **Pedestrian Areas :** this is the list of all the pedestrian areas in Paris, associated with their geo_shape.
  Click here to find the dataset.
- **Public Wi-Fi hotspots :** when people travel, they may lack internet connection or have a very poor one. So, we consider it's important to have the list of the Wi-Fi hotspots offered

by the city.
Click here to find the dataset

- **Open air markets :** it's the list of the outdoors market that are present in Paris. It contains information about the kind of products sold in the market, its position and its timetable.
Click here to find the dataset

- **Unusual Walks and PoI :** Paris is not only about the Tour Eiffel, Notre Dame and the popular places. There are so many other interesting places to discover and this dataset is a great source for Paris hidden gems.
Click here to find the dataset

- **Green Area and Similar :** it's the list of open and green areas managed by the city of Paris. Many kinds of areas are available, such as: open walks, decorative gardens and road decorations.
Click here to find the dataset

- **Public Bike Stations :** Tourists may want to be move in a fast, green and affordable way so we will include the list of Paris bike stations. There are more than 1,400 bike stations available in Paris and its outskirts.
Click here to find the dataset

- **Cycling Tracks :** The list of cycling tracks in the city, associated with their geo_shape.
Click here to find the dataset

- **Taxi Stations :** The list of public taxi stations available in Paris, associated with their phone number and location.
Click here to find the dataset

- **Interesting Activities :** A list with thousands of events happening in Paris. Such as: festivals, conferences, workshops, guided tours and many more.
Click here to find the dataset

OpenData-Paris is offering us most of the data necessary for our DI project, yet there's some information useful for the tourists that we weren't able to find there. So we decided to retrieve this data from open-street maps. In order to extract data from open-street maps we used a tool called Overpass Turbo which allowed us to fetch the information we require. The data is stored as geojson format and not all the elements in the list have the same fields so particular attention is required when working with these datasets. Here's the list of elements we're going to retrieve:

- **Shopping center :** the list of malls and department stores useful for tourists who want to shop in Paris.
- **Restaurants :** list of restaurants in the city.
- **Bus station :** list of Paris bus stations.
- **Subway station :** list of Paris subway stations.
- **Other points of interest :** list of places that can be useful for some of the users: parking lot, bakeries, pharmacy.

2.4 Metadata

The metadata of the data sources and datasets collected are in two turtle files located in our github repository (Click here for the link).

In the appendix, you can also find extra information regarding the metadata of single attributes within datasets.

Chapter 3

Competency Queries

In the chapter you will find the first two stages of the CQs analysis.

3.1 Raw CQs

In the following table you will find questions that our users could ask to the knowledge graph. This set of competency queries describe the needs of the different personas.

Person	Number	Question	Action
Clark	1.1	Where is the closest taxi station ? I also want the phone number	Given the position, the system returns the address of the nearest taxi station and its phone number
Clark	1.2	Which are the open museums in a 6-hour range, with a taxi, of the Charles de Gaule airport ? What is their address ?	Given the position and the time, the system returns the names and address of the museums
Clark	1.3	What is the nearest coffee shop with free Wi-fi ?	Given the position, the system returns the position of the nearest coffee shop with free Wi-fi
Clark	1.4	What are the open Paris museums that take less than 2 hours to visit ?	Given the time, the system returns the name of the open museums that take less than 2 hours to visit
Clark	1.5	Are there events happening during my 6-hours stop with a free access ? What are they about ?	Given the time slot, the system returns the name and description of the events with a free access happening during Clark's stop

Clark	1.6	Where is the closest ATM ? Is it open 24/24h ?	Given the position of Clark, the system returns the position of the closest ATM and whether it is open 24/24h
Lilian	2.1	Where are the public Wi-fi hotspots 1 km around the hotel ?	Given the hotel position, the system returns the position of the public Wi-fi hotspots
Lilian	2.2	How can I discover new and unusual sides of Paris ? Give me general information and the referred website	Given keywords, the system returns the name of the walking path, the introduction text and the referred website
Lilian	2.3	Are there romantic parks around the Eiffel tower open at night ?	Given keywords about romance, the system returns the name and localisation of corresponding parks
Lilian	2.4	What are the points of interest on the way between the train station and the hotel ?	Given the position of the train station and hotel, the system returns the name and localization of the points of interest
Isabella	3.1	Were are the free parking lots in a 20 min range, by foot, of the Louvre museum ?	Given the time range and the mean of transportation, the system returns the position of the free parkings
Isabella	3.2	Were is the closest parking lot to the museum with a hourly fee less than 1.00€ ?	Given the position of the museum, the system checks for all the parking with a fee less than 1.00€ and returns the position of the closest one
Isabella	3.3	What is the speed limit of a specific bike track ?	Given the name of a bike track, the system returns the speed limit
Isabella	3.4	What are the bike tracks that cross woods ? How long are they ?	The system returns the name and length of the corresponding bike tracks
Ginno	4.1	How many parking spots are there in a 5 min walk for my office ?	Given the office position, the system returns the number of corresponding parking spots
Ginno	4.2	What is the closest subway station ? Where is it ?	Given the office position, the system returns the name and position of the nearest subway station
Ginno	4.3	How many bars are there in a 5 km range around my office ?	Given the office position, the system returns the number of corresponding bars

Ginno	4.4	Where is the closest pharmacy ?	Given the office position, the system returns the position of the closest pharmacy
Ginno	4.5	Where is the closest bakery ? What is the opening hour ?	Given the office position, the system returns the position of the closest bakery and its opening hour
Ginno	4.6	Where is the closest ATM ? Is it open 24/24h ?	Given the office position, the system returns the position of the closest ATM and true if its open 24/24h, false if not
Ginno	4.7	Where is the closest bike-renting station ? Give me also the capacity and if it is renting and returning	Given the office position, the system returns the position of the closest bike-renting station, as well as its capacity and if it is renting and returning
Brigitte	5.1	Where is the closest cinema playing a kid movie on Saturday ?	Given the position of the flat, the system returns the position of the closest cinema playing a kid movie on Saturday
Brigitte	5.2	What are the events in a 10 km range around my house during the week-end ? What is the access fee for me and the children ?	Given the position of the flat, the system returns the name, date and position of the events, as well as the type of access (free or booking) and the entry fee
Brigitte	5.3	Is there a pizza or burger restaurant that does takeaway in a 5 km radius ?	Given the position of the flat, the system returns the name and address of the corresponding restaurants
Martina	6.1	Where are the quiet green areas near the city center opened during lunch time ?	The system returns the position and characteristics of the quiet green areas in a 2 km radius around the city center
Martina	6.2	What are the interesting activities in Paris happening during the school trip and having a PMR access ? Do they also have a visually and hearing impaired access ?	Given keywords and time slot, the system returns the name and description of the events having a PMR access, and whether or not they also have a visually and hearing impaired access

Martina	6.3	Does a given bus stop has a shelter ? And is it available for wheelchairs ?	Given the name or position of the bus stop, the system returns whether it has a shelter and a wheelchair access
Marie	7.1	Where are the shopping centers in Paris, in a 10km range around the city center ? And do they have toilets ?	Given a range, the system returns the position of the shopping centers
Marie	7.2	What are the 3 top streets with the highest density of shops in the city center ?	Given a range and the number of streets, the system returns the names of the streets
Marie	7.3	Are there open air markets on a given date ? And what kind of products are sold ?	Given the date, the system returns the name, opening hours and description of the open air markets, with a list of what products are sold
Tommaso	8.1	What are the top-rated restaurants from a certain district ?	Given the district number and the minimum grade, the system returns a list of restaurants
Tommaso	8.2	Where are the top-rated restaurants from a certain district ?	Given the district number and the minimum grade, the system returns a list of positions
Tommaso	8.3	Where is the closest restaurant ? What are its opening hours ?	Given the position, the system returns the localisation of the closest restaurant
Tommaso	8.4	How many restaurants are there is a given district ?	Given the number of the district, the system returns the number of restaurants
Tommaso	8.5	Where are the open air markets that sell food ?	Given keywords, the system returns the position and opening dates of the markets that sell food
Tommaso	8.5	Are there food related activities during my stay ? If so, what is the date and location ?	Given the time slot, the system returns the name, description and position of food-related activities in the city
Tommaso	8.6	Where are the {cuisine type} restaurants ?	Given the type of cuisine, the system returns the name and position of the corresponding restaurants, sorted from the closest to the farthest

3.2 Kernel CQs

You will find, in the table below, the Kernel CQs. They are the Raw CQs minus the auxiliary words, resulting in each term being a concept.

Person	Number	Raw CQ	Kernel CQ
Clark	1.1	Where is the closest taxi station ? I also want the phone number	Position, Place, Station, Taxi, Contact information, Phone number
Clark	1.2	Which are the open museums in a 6-hour range, with a taxi, of the Charles de Gaule airport ? What is their address ?	Logistic information, Establishment, Museum, Means of Transportation, Taxi, Position, Address
Clark	1.3	What is the nearest coffee shop with free Wi-fi ?	Position, Establishment, Coffee shop, Service, Wi-fi
+ Clark	1.4	What are the open Paris museums that take less than 2 hours to visit ?	Establishment, Museum, Logistic information
Clark	1.5	Are there events happening during my 6-hours stop with a free access ? What are they about ?	Event, Logistic information, Access, Promotion information
Clark	1.6	Where is the closest ATM ? Is it open 24/24h ?	Position, Establishment, ATM, Logistic information
Lilian	2.1	Where are the public Wi-fi hotspots 1 km around the hotel ?	Position, Service, Wi-fi, Establishment, Hotel
Lilian	2.2	How can I discover new and unusual sides of Paris ? Give me general information and the referred website	Place, Promotion information, General information, Website
Lilian	2.3	Are there romantic parks around the Eiffel tower open at night ?	Place, Park, Position, Establishment, Eiffel tower, Logistic information
Lilian	2.4	What are the points of interest on the way between the train station and the hotel ?	Place, Establishment, Station, Train, Hotel
Isabella	3.1	Were are the free parking lots in a 20 min range, by foot, of the Louvre museum ?	Place, Parking lot, Logistic information, Position, Establishment, Museum
Isabella	3.2	Were is the closest parking lot to the museum with a hourly fee less than 1.00€ ?	Position, Place, Parking lot, Logistic information

Isabella	3.3	What is the speed limit of a specific bike track ?	Logistic information, Place, Bike track
Isabella	3.4	What are the bike tracks that cross woods ? How long are they ?	Place, Bike track, Woods, Logistic information
Ginno	4.1	How many parking spots are there in a 5 min walk for my office ?	Place, Parking spot, Position
Ginno	4.2	What is the closest subway station ? Where is it ?	Position, Place, Station, Subway, Position
Ginno	4.3	How many bars are there in a 5 km range around my office ?	Establishment, Bar, Position
Ginno	4.4	Where is the closest pharmacy ?	Position, Establishment, Pharmacy
Ginno	4.5	Where is the closest bakery ? What is the opening hour ?	Position, Establishment, Bakery, Logistic information
Ginno	4.6	Where is the closest ATM ? Is it open 24/24h ?	Position, Establishment, ATM, Logistic information
Ginno	4.7	Where is the closest bike-renting station ? Give me also the capacity and if it is renting and returning	Position, Place, Station, Bike-renting, Logistic information
Brigitte	5.1	Where is the closest cinema playing a kid movie on Saturday ?	Position, Establishment, Cinema, Event, Movie, Logistic information
Brigitte	5.2	What are the events in a 10 km range around my house during the week-end ? What is the access fee for me and the children ?	Event, Position, Logistic information
Brigitte	5.3	Is there a pizza or burger restaurant that does takeaway in a 5 km radius ?	Establishment, Restaurant, Logistic information, Position
Martina	6.1	Where are the quiet green areas near the city center opened during lunch time ?	Place, Green area, Position, Logistic information
Martina	6.2	What are the interesting activities in Paris happening during the school trip and having a PMR access ? Do they also have a visually and hearing impaired access ?	Event, Logistic information, Access

Martina	6.3	Does a given bus stop has a shelter ? And is it available for wheelchairs ?	Place, Bus stop, Logistic information
Marie	7.1	Where are the shopping centers in Paris, in a 10km range around the city center ? And do they have toilets ?	Establishment, Shopping center, Position, Logistic information
Marie	7.2	What are the 3 top streets with the highest density of shops in the city center ?	Place, Street, Logistic information, Position
Marie	7.3	Are there open air markets on a given date ? And what kind of products are sold ?	Place, Market, Logistic information
Tommaso	8.1	What are the top-rated restaurants from a certain district ?	Logistic infomation, Establishment, Restaurant, Position
Tommaso	8.2	Where are the top-rated restaurants from a certain district ?	Position, Logistic information, Establishment, Restaurant
Tommaso	8.3	Where is the closest restaurant ? What are its opening hours ?	Position, Establishment, Restaurant, Logistic information
Tommaso	8.4	How many restaurants are there is a given district ?	Establishment, Restaurant, Position
Tommaso	8.5	Where are the open air markets that sell food ?	Position, Place, Market, Logistic information
Tommaso	8.5	Are there food related activities during my stay ? If so, what is the date and location ?	Event, Activities, Logistic information, Position
Tommaso	8.6	Where are the {cuisine type} restaurants ?	Position, Logistic information, Establishment, Restaurant

3.3 Analysed CQs

In the list below, you will find Analysed CQs, where each concept in the Kernel CQs is classified as common, core or contextual concepts

1. 1.1. Common : Position, Place
   Core : Station, Contact information
   Contextual : Taxi, Phone number

   1.2. Common : Establishment, Means of transportation, Position
   Core : Logistic information
   Contextual : Museum, Taxi, Address

1.3. Common : Position, Establishment
Core : Service
Contextual : Coffee shop, Wi-fi

1.4. Common : Establishment
Core : Logistic information
Contextual : Museum

1.5. Common : Event
Core : Logistic information, Promotion information
Contextual : Access

1.6. Common : Position, Establishment
Core : Logistic information
Contextual : ATM

2. 2.1. Common : Position, Establishment
Core : Service
Contextual : Wi-fi, Hotel

2.2. Common : Place
Core : Promotion information
Contextual : General information, Website

2.3. Common : Place, Establishment
Core : Park, Logistic information
Contextual : Eiffel tower

2.4. Common : Establishment
Core : Place
Contextual : Station, Train, Hotel

3. 3.1. Common : Establishment, Position
Core : Place, Logistic information
Contextual : Parking lot, Museum

3.2. Common : Position
Core : Place, Logistic information
Contextual : Parking lot

3.3. Common : Place
Core : Logistic information
Contextual : Bike track

3.4. Common : Place
Core : Logistic information
Contextual : Bike track, Woods

4. 4.1. Common : Position
Core : Place
Contextual : Parking spot

4.2. Common : Place
Core : Position, Station
Contextual : Subway

4.3. Common : Position
Core : Establishment
Contextual : Bar

4.4. Common : Position
Core : Establishment
Contextual : Pharmacy

4.5. Common : Position
Core : Establishment, Logistic information
Contextual : Bakery

4.6. Common : Position
Core : Establishment, Logistic information
Contextual : ATM

4.7. Common : Position, Establishment
Core : Station, Logistic information
Contextual : Bike-renting

5. 5.1. Common : Position, Establishment
Core : Event, Logistic information
Contextual : Cinema, Movie

5.2. Common : Position
Core : Logistic information
Contextual : Event

5.3. Common : Position
Core : Establishment, Logistic information
Contextual : Restaurant

6. 6.1. Common : Position
Core : Place, Logistic information
Contextual : Green area

6.2. Common : Event
Core : Logistic information
Contextual : Access

6.3. Common : Place
Core : Logistic information
Contextual : Bus stop

7. 7.1. Common : Establishment, Position
Core : Logistic information
Contextual : Shopping center

7.2. Common : Position
Core : Place, Logistic information
Contextual : Street

7.3. Common : Place
Core : Logistic information
Contextual : Market

8. 8.1. Common : Position
Core : Establishment, Logistic information
Contextual : Restaurant

8.2. Common : Position
Core : Logistic information, Establishment
Contextual : Restaurant

8.3. Common : Position
Core : Establishment
Contextual : Restaurant

8.4. Common : Position
Core : Place, Logistic information
Contextual : Market

8.5. Common : Position
Core : Logistic information, Event
Contextual : Activities

8.6. Common : Position
Core : Establishment, Logistic information
Contextual : Restaurant

Chapter 4

Inception

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

4.1 Knowledge resource collection

To create our ontology, we used the schemas from schema.org. Since our project purpose is to enable tourists in Paris to find activities, we selected the Event and Place Etypes. Thoses Etypes have sub-categories, which we are going to use to be more specific in our CQ. You can find in the table below the selection we made.

4.2 Resource classification

We're now going to have a look out the data we can collect and see what kind of category (Common, Core, Contextual) they can satisfy.

- **Pedestrian Areas dataset** [*common*] : it contains very simple data related to locations.
- **Public Wi-Fi hotspots dataset** [*common, contextual*] : besides location information there's detailed data about the Wi-Fi hotspot.
- **Open air markets dataset** [*common, core, contextual*] : this dataset is complete with most of the data relevant with markets such as: position, logistic information and type of market.
- **Unusual Walks and PoI dataset** [*common, core, contextual*]: this dataset is also covering all categories because it has information about: position, place and descriptive text.
- **Green Areas and Similar dataset** [*common, core, contextual*]: here we have very detailed information like: location, logistic information, opening hours, details about perimeter and horticultural area.
- **Public Bike Stations dataset** [*common, contextual*] : besides location information we can find some contextual information such as the capacity of each station.

- **Cycling Tracks dataset** [*common, core, contextual*] : the dataset is touching over all 3 categories with information like: location, station and layout.
- **Taxi Stations dataset** [*common, core, contextual*] : the fields of this dataset is
- **Interesting Activities dataset** [*common, core, contextual*] : this dataset is extremely detailed offering us any kind of information we could desire of.
- **OpenStreetMap dataset** [*common*] : we consider the OpenStreetMap to contain only common data, because only basic information is available for every place. While core and contextual data is not always present.

We can see that most of our datasets are offering a complete view over the elements we need for our DI project. It happens because OpenData Paris tends to aggregate data in very big tables with a high number of fields (attributes).

4.3 Evaluation

The evaluation phase is essential for a wide adoption of our model, since it assesses the quality and the ”fitness for use” of the model. You will find below the set parameters and the evaluation metrics. We used the ontology from schema.org to compute the metrics.

1. - Classes in CQs : CQ_c = {ATM, activity, airport, bakery, bar, bike-rentingStation, bikeTrack, busStop, cinema, cityCenter, coffeeShop, district, event, greenArea, hotel, museum, open-airMarket, park, parkingLot, parkingSpot, pharmacy, POI, publicWifi, restaurant, shoppingCenter, street, subwayStation, taxiStration, trainStation, wood }
     $\text{Num}(\text{CQ}_c) = 30$
   - Properties in CQs : CQ_p = { about, access, adress, capacity, cuisineType, date, fee, freeWifi, length, location, openingHours, phoneNumber,PMR, productsSold, program, rating, renting, returning, shelter, speedLimit, takeaway, toilets, visitingTime, website }
     $\text{Num}(\text{CQ}_p) = 30$
2. - Classes in ontology : Ont_c = {accomodation, action, amusementPark, apartment, automatedTeller, bakery, barOrPub, businessEvent, busStop cafeOrCoffeeShop, entertainementBusiness, event, exerciseAction, festival, financialService, foodEstablishment, foodEvent, hotel, intanglible, internetCafe, localBusiness, lodgingBusiness, movieTheater, organization, place, playAction, product, publicToilet, restaurant, service, shoppingCenter, store, taxi, thing, touristInformationCenter, touristTrip, trip }
     $\text{Num}(\text{Ont}_c) = 37$
   - Properties in ontology : Ont_p = {about, accessibilityFeature, adress, areaServed, arrivalTime, brand, currenciesAccepted, departureTime, distance, doorTime, duration, email, endDate, hasDriveThroughService, hoursAvailable, inLaguage, isAccessibleForFree, knowsLanguage, legalName, location, makesOffer, maximumAttendeeCapacity, paymentAccepted,performer, priceRange, review, review, servesCuisine, serviceType, smokingAllowed, startDate, telephone, typicalAgeRange,URL}
     $\text{Num}(\text{Ont}_p) = 34$

- - Classes in data : $\mathrm{Num}(\mathrm{D}_c) = 17$ (see the appendix for more details)
  - Properties in data : $\mathrm{Num}(\mathrm{D}_p) = 194$ (see the appendix for more details)

The results are summarized in the following table :

	Class	Property
Coverage	11/30 = 0,37	17/30 = 0,57
Extensiveness	7/67 = 0,10	4/64 = 0,06
Sparsity	1-(20/47) = 0,57	1-(27/224) = 0,88

Chapter 5

Informal Modeling

This section is dedicated to the description of the informal 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:

- Purpose formalization (informal modeling part) and Modeling sheet description
- ER model description
- Informal Modeling evaluation

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.

In order to execute the Informal Modeling phase, we became familiar with the notion of Ontology and Teleology in iTelos. It is not possible to produce a global schema with the objective to integrate all the data available. For this reason the Datasets Selection activity plays a crucial role in the identification of those datasets containing all and only the information required to satisfy the Purpose.

According to Gruber (1993) and Studer (1998), an ontology is a formal, explicit specification of a shared conceptualization. It is used to capture knowledge about some domain of interest and describes the concepts and the relationships that hold between those concepts. Teleologies are ontologies but with the specificality that teologies focus on function and how a chosen representation fits a certain purpose.

5.1 ETypes definition

We divided this section in three categories :

- Common for the ETypes that are associated to aspects which are commin to all domains, also outside our project's DoI;

- Core for the ETypes that carry information about the most important aspects regarding our project's purpose;
- Contextual for ETypes related to specific, possibly unique, information from the domain of interest. Their main goal is to create added value.

5.1.1 Common

Thing

A common etype describing the most generic type of item

Relations :

- it may be a Place
- it may be an Establishment

Attributes : it hasn't specific attributes

Place

A common etype describing entities that have somewhat fixed, physical extension.

Relations :

- it is the super class of GreenArea, ParkingLot, ATM, BikeStation, BikeTrack, TaxiStation, PointOfInterest and BusStop

Attributes :

- ID : it has an ID which allows to identify it in the database (string)
- name : it has max 1 string that refers to the name of the place

Location

A common etype describing

Relations :

- it may be the position of a Place
- it may be the position of an Establishment

Attributes :

- latitude : latitude coordinate in the world (real)
- longitude : longitude coordinate in the world (real)

Establishment

A common etype describing establishments where services and/or products are provided

Relations :

- it is the super class of Cinema, OpenAirMarket, Museum, CoffeeShop, Bar, Bakery, Pharmacy, ShoppingCenter, Restaurant

Attributes :

- ID : it has an ID which allows to identify it in the database (string)
- name : it has max 1 string that refers to the name of the establishment

5.1.2 Core

GreenArea

A core etype describing green areas (parks, public gardens, ...).

Relations :

- It is a possible function of a Place, therefore it inherits all the Place relations and attributes

Attributes :

- type : a string describing what kind of green area it is (pen walks, decorative gardens, private gardens, peripheral, etc)
- area : the surface size (integer)
- category : category of the area (string)
- is24hrs : a boolean true if the area is open 24h/24

ParkingLot

A core etype describing parking lots and other parking facilities.

Relations :

- It is a possible function of a Place, therefore it inherits all the Place relations and attributes

Attributes :

- type : a string describing the type of the parking lot (underground for example)
- fee : an integer indicating the price

ATM

A core etype describing the ATMs and other ways of getting cash money.

Relations :

- It is a possible function of a Place, therefore it inherits all the Place relations and attributes

Attributes :

- provider : the name of the bank operator (string)
- wheelchairAccess : boolean true if the ATM is accessible for wheelchair users
- is24hrs : boolean true if the ATM is open 24h/24

BikeStation

A core etype describing the bike-renting stations.

Relations :

- It is a possible function of a Place, therefore it inherits all the Place relations and attributes

Attributes :

- capacity : number of bikes the station can hold (integer)
- isRenting : boolean true if the station allows the user to rent bikes
- isReturning : boolean true if the station allows the user to return bikes

BikeTrack

A core etype describing the tracks made for biking in the city.

Relations :

- It is a possible function of a Place, therefore it inherits all the Place relations and attributes

Attributes :

- type : a string describing the type of track
- speedLimit : speed limit of the track (integer)
- isBidirectional : boolean true if bikes can go in both directions on the track
- hasForest : boolean true if the track is going through the small forests in Paris

TaxiStation

A core etype describing the places where a taxi service is provided.

Relations :

- It is a possible function of a Place, therefore it inherits all the Place relations and attributes

Attributes :

- phoneNumber : phone number of the taxi station (string)

PointOfInterest

A core etype describing interesting public spots in the city.

Relations :

- It is a possible function of a Place, therefore it inherits all the Place relations and attributes

Attributes :

- category : a string describing the type of PoI: place-street, architectural-element, museum-cultural-place, religious heritage, etc.
- description : natural language text describing the PoI (string)
- keywords : comma separated keywords related to the PoI (string)
- website : official website of the PoI (string)

BusStop

A core etype describing bus stops.

Relations :

- It is a possible function of a Place, therefore it inherits all the Place relations and attributes

Attributes :

- hasShelter: true if the bus station offers a shelter place
- wheelchairAccess : boolean true if the ATM is accessible for wheelchair users

SubwayStation

A core etype describing subway stations.

Relations :

- It is a possible function of a Place, therefore it inherits all the Place relations and attributes

Attributes :

- wheelchairAccess : boolean true if the ATM is accessible for wheelchair users

Wi-fiHotspot

A core etype describing the public Wi-fi hotspots.

Relations :

- It is a possible function of a Place, therefore it inherits all the Place relations and attributes

Attributes :

- Status : true if the Wi-Fi hotspot is working

Event

A core etype describing an event happening at a certain time and location, such as a concert, lecture, or festival.

Relations :

- Every event, has a starting time and a finishing time, thus it's related to the OpenHours table.

Attributes :

- Title : name of the event
- Website : website with information about the event and booking
- Description : description of the event
- wheelchairAccess : boolean true if the ATM is accessible for wheelchair users
- PMR access : true if the event allows PMR access
- Phone number : phone number of the administrator of the event
- Type of access : indicates if the event requires booking or it's free access
- Fee : amount of money need for booking at the event (can be 0)

Cinema

A core etype describing cinemas and movie theaters.

Relations :

- it has a timetable, so it is a possible function of Establishment. Therefore it inherits all the Establishment relations and attributes

Attributes :

- Website: website of the cinema
- brand: name of the cinema brand
- Phone number: number of the cinema place

OpenAirMarket

A core etype describing outdoors markets happening at a certain time and location.

Relations :

- it has a timetable, so it is a possible function of Establishment. Therefore it inherits all the Establishment relations and attributes

Attributes :

- Product: type of products sold in the market
- District: Parisian district where the market is located

Museum

A core etype describing museums.

Relations :

- it has a timetable, so it is a possible function of Establishment. Therefore it inherits all the Establishment relations and attributes

Attributes :

- Website : official website of the museum
- Phone number: number of the Museum info point
- wheelchairAccess : boolean true if the museum is accessible for wheelchair users

CoffeeShop

A core etype describing the cafe and coffee shops.

Relations :

- it has a timetable, so it is a possible function of Establishment. Therefore it inherits all the Establishment relations and attributes

Attributes :

- Cuisine : type of coffees prepared by the shop
- hasTakeaway : true if the shop offers takeaway
- hasWiFi : true if the coffee shop has free wi-fi

Bar

A core etype describing bars.

Relations :

- it has a timetable, so it is a possible function of Establishment. Therefore it inherits all the Establishment relations and attributes

Attributes :

- Website : official website of the restaurant
- Phone number: number of the bar
- wheelchairAccess : boolean true if the bar is accessible for wheelchair users

Bakery

A core etype describing bakeries.

Relations :

- it has a timetable, so it is a possible function of Establishment. Therefore it inherits all the Establishment relations and attributes

Attributes :

- wheelchairAccess : boolean true is the bakery provides an access for wheelchairs

Pharmacy

A core etype describing pharmacies.

Relations :

- it has a timetable, so it is a possible function of Establishment. Therefore it inherits all the Establishment relations and attributes

Attributes :

- wheelchairAccess : boolean true is the pharmacy provides an access for wheelchairs

ShoppingCenter

A core etype describing shopping centers or malls.

Relations :

- it has a timetable, so it is a possible function of Establishment. Therefore it inherits all the Establishment relations and attributes

Attributes :

- type : type of the center (mall, department store) (string)
- hasToilets : boolean true is the shopping center has public toilets
- wheelchairAccess : boolean true is the shopping center provides an access for wheelchairs

Restaurant

A core etype describing restaurants and places you can eat.

Relations :

- it has a timetable, so it is a possible function of Establishment. Therefore it inherits all the Establishment relations and attributes

Attributes :

- cuisine : type of food cooked in the restaurant (string)
- phoneNumber : phone number of the restaurant (string)
- hasTakeaway : boolean true if the restaurants provide a takeaway service

5.1.3 Contextual

OpenHours

A contextual etype describing the general opening hours for a business.

Relations :

-

Attributes :

- day : string containing the days where the establishment is open
- start : a datetime corresponding to the opening hour
- end : a datetime corresponding to the closing hour

Movie

A contextual etype describing movies on the program of a specific cinema.

Relations :

- A movie is broadcasted in a specific Cinema, thus it's directly related to it.

Attributes :

- title : a string containing the title of the movie
- genre : a string describing the genre of the movie (horror, comedy, ...)
- duration : the number of minutes of the movie (integer)

5.2 Modeling sheet description

In order to continue towards the creation of the basic ER model we had to further analyse the competency questions and extract the objects, functions and actions related to our domain. With that classification we were then able to better formulate and understand the different ETypes and Data Properties required for our project. Whenever we worked with the CQs we made sure to take into consideration the data we have available, so that we can do our best to describe it. The dataset descriptions helped us a lot in doing that (see appendix 10.1). The Modeling Sheet file is collecting the results of the classified and attributed CQs. The first one is a quite sparse table because some kernel CQs have a small number of elements or take into consideration some contextual attributes such as the phone number of specific services.

5.3 ER Model

With the Modeling Sheet at hand we were able to create our informal ER, representing the objects, functions and actions of our domain. We have two main objects (place and establishment) having many different functions. The first object is related to geographical areas without a specific timetable (people can always go there) while the second object is describing geographical areas where people can go only during certain days and hours. This division into two objects is useful both for better organizing our data and for potential tourists who want to plan their trips using the final result of our project. Another important element of our informal ER is the event EType, which represents the different events happening in the city of Paris.

5.4 Evaluation

5.4.1 Schema level

For the evaluation, we aim to measure :

- if the proposed informal ER model covers CQs, using coverage
- if the proposed informal ER model properly extend CQs, using extensiveness

1. - Classes in CQs : CQ_c = {ATM, activity, airport, bakery, bar, bike-rentingStation, bikeTrack, busStop, cinema, cityCenter, coffeeShop, district, event, greenArea, hotel, museum, open-airMarket, park, parkingLot, parkingSpot, pharmacy, POI, publicWifi, restaurant, shoppingCenter, street, subwayStation, taxiStration, trainStation, wood }
     $\text{Num}(CQ_c) = 30$
   - Properties in CQs : CQ_p = { about, access, adress, capacity, cuisineType, date, fee, freeWifi, length, location, openingHours, phoneNumber,PMR, productsSold, program, rating, renting, returning, shelter, speedLimit, takeaway, toilets, visitingTime, website }
     $\text{Num}(CQ_p) = 30$
2. - Classes in ER model : ER_c = { Thing, Place, Establishement, GreenArea, ParkingLot, ATM, BikeStation, BikeTrack, TaxiStation, Location, OpenHours, Event Cinema, Movie, OpenAirMarket, Museum, CoffeeShop, Bar, Restaurant, Bakery, Pharmacy, Shopping-Center }
     $\text{Num}(ER_c) = 26$
   - Properties in ER model : ER_p = {name, type, surfaceArea, category, is24hrs, fee, provider, wheelchairAccess, capacity, isRenting, isReturning, speedlimit, isBidirectional, hasForest, lenght, phoneNumber, description, keywords, website, hasShelter, status, blindAccess, deafAccess, typeOfAccess, day, from, to, latitude, longitude, brand, products, title, genre, duration, cuisine, hasTakeaway, hasWifi, hasToilets}
     $\text{Num}(ER_p) = 38$

The results are summarized in the following table :

	Class	Property
Coverage	15/30 = 0,5	26/30 = 0,87
Extensiveness	4/56 = 0,07	8/68 = 0,12

5.4.2 Data level

For the evaluation, we aim to measure :

- if the informal ER model aligns with collected datasets, using coverage
- if the informal ER model is much different from collected datasets, using sparsity
- Classes in data : Num(D_c) = 17 (see the appendix for more details)
- Properties in data : Num(D_p) = 194 (see the appendix for more details)

The results are summarized in the following table :

	Class	Property
Coverage	13/17 = 0,76	38/194 = 0,20
Sparsity	1-(13/43) = 0,70	1-(38/232) = 0,84

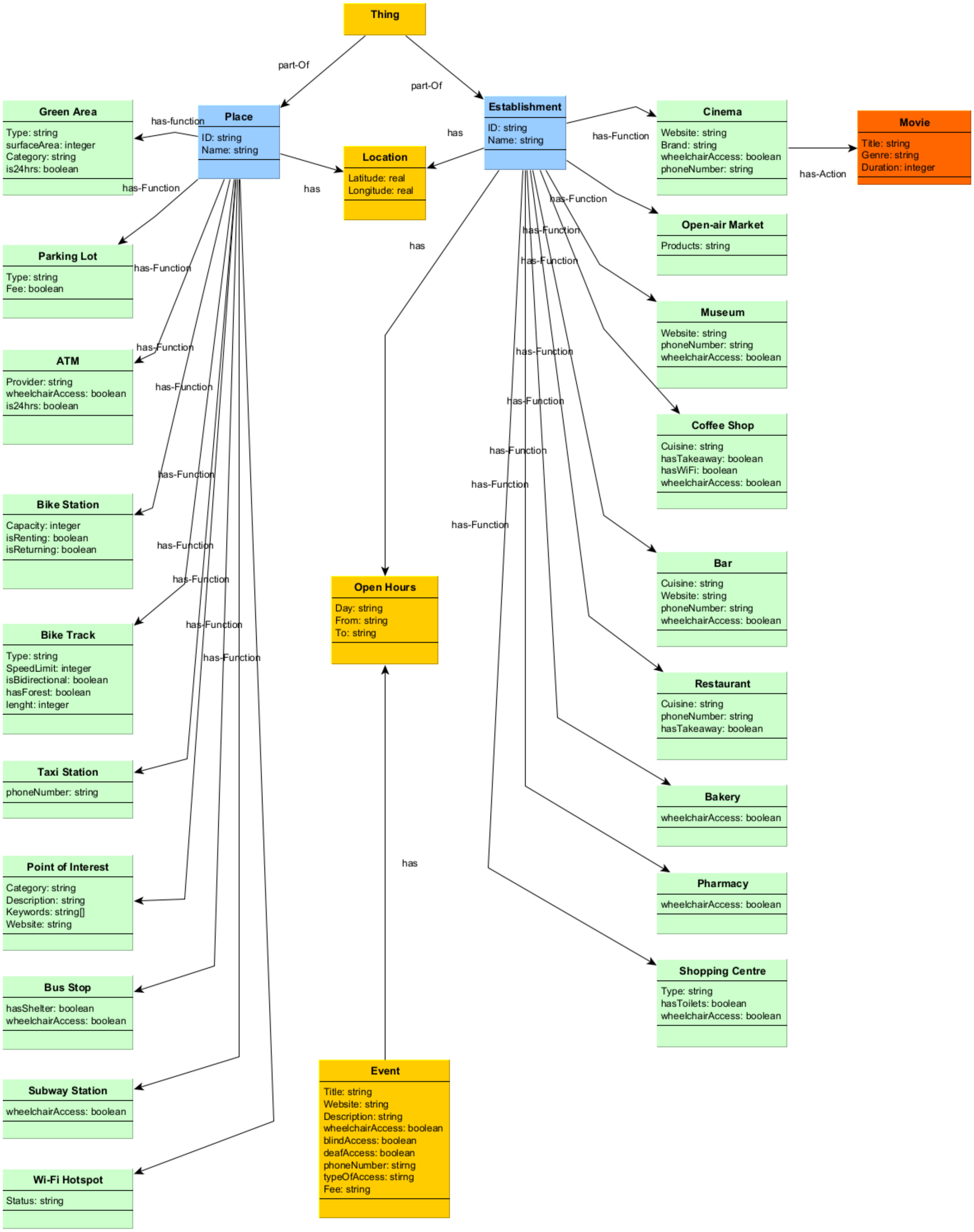


Figure 5.1: ER Model

Chapter 6

Formal Modeling

This section is dedicated 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
- Data management (syntactic heterogeneity)
- Formal Modeling evaluation

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 ETG generation

In order to produce the ETG graph, we have to take into account our teleology and the following foundational realtions :

- *hasFunction* relates objects to functions and illustrates the fact that objects can have one or more admissible functions
- *hasFunctionAction* relates functions to actions and illustrates the fact that functions can be realized via one or more admissible functions
- *hasObjectAction* relates objects to actions and illustrates the fact that objects can have one or more admissible functions
- *ObjectToObjectRelation* models the diverse array of semantic relations existing between different objects

6.1.1 Ontology selection

This step is focused on reusing ontology elements which are semantically synonymous with concepts which model our ER. Doing this improves the reusability of our system. We studied other ontologies such as the Geospatial ontology (click here for the link) and DBpedia (click here for the link). We also found dedicated papers and in particular the ”Unified ontology for data integration for tourism sector” by C. Virmani, S. Sinha and S. K. Khatri [1].
After the analysis of these sources, we came up with a possible modification of our ER model : add a class **Person**, with the attributes *uses* (a service), *eatsAt* (a food establishement), *staysIn* (an accomodation) and *enjoys* (an event). But we didn't found any data on the activity of tourists in Paris, so we did not pursue with this idea and kept our ER as it was before.

6.1.2 Language alignment

The terms we used until now to represent our data are still informal and there is no unique interpretation of their meaning. Thus we had to focus on the language alignment process in order to assign a unique meaning (GID + definition) to every concept involved in our DI project.
We picked every term and searched for it in the UKC KB via KOS. In case of synonymous match, the term was saved with its GID and definition, otherwise a new negative GID and definition was created.
We created an excel file where you can see the outcome result of our language alignment process (click here for the link) . On column C you will see the name of the concept as found in the UKC KB, column D contains the name of the concepts as found in the ER and column E is the name of the concepts that will be found in the ontology file. Column B and F have the GID and description of the concepts.

6.1.3 Schema alignment

With the informal ER, the language alignment spreadsheet and the foundational teleology at hand we were ready to generate the formal ETG. We used Protégé to populate the foundational teleology with our domain-specific concepts, by taking care of adding each one of them in the correct subcategory. The process took quite some time because we have many classes and data properties but when we finally uploaded it to KOS our ontology was immediately accepted, no error was generated.
We later found out that it was actually not the right way to do it so we started from scratch, upoloaded the file on KOS without GID or types and resolved the issues.

6.2 Data management

In this phase, we re-organised the data in order to align it better to the ER schema.

The data we collected comes from various sources and similar information is represented in many different ways that we need to elaborate and align. We used our informal ER model as a reference for “cleaning” the data through a node JS application; you can find its code in this

Github repository (Click here for the link). We took each dataset and we performed the following steps:

1. Pick the dataset fields required by the ER
2. Assign a standard name to the required fields
    - Many of our datasets are in French language so we had to translate their fields names in English
    - Many of our datasets are in French language so we had to translate their fields names in English
3. Fix the field format, if required
    - Fix the field format, if required Some fields would allow only binary values such as “Oui”/”Non”, those were transformed into boolean values
    - Some fields would contain an unstructured string representing complex information such as the schedule of a restaurant. In this case, every string is carefully analysed through a function returning a structured representation of the data

6.3 Evaluation

For the evaluation, we aim to measure :

- If the formal ETG and its Etypes are properly defined by their properties, using Cue
- If the proposed ETG is different from the reference ontologies, using Sparsity
- If the ETG is well-designed, and information in the ETG is correct. By sampling from ETG and then checking manually

To calculate Cue validity, we first need to generate a FCA lattice for our ETG.
We were not able to generate the Cur indicator, so we used coverage, extensiveness and sparsity.

1.
    - Classes in the formal ETG : ETG_c = { Thing, Place, Establishement, GreenArea, ParkingLot, ATM, BikeStation, BikeTrack, TaxiStation, Location, OpenHours, Event, Cinema, Movie, OpenAirMarket, Museum, CoffeeShop, Bar, Restaurant, Bakery, Pharmacy, ShoppingCenter }
      $\text{Num}(\text{ETG}_c) = 26$
    - Properties in the formal ETG : ETG_p = {name, type, surfaceArea, category, is24hrs, fee, provider, wheelchairAccess, capacity, isRenting, isReturning, speedlimit, isBidirectional, hasForest, lenght, phoneNumber, description, keywords, website, hasShelter, status, blindAccess, deafAccess, typeOfAccess, day, from, to, latitude, longitude, brand, products, title, genre, duration, cuisine, hasTakeaway, hasWifi, hasToilets}
      $\text{Num}(\text{ETG}_p) = 38$
2. More details on the schema.org website. We chose to consider the entire schema.org ontology.

- Classes in reference ontology : $\mathrm{Num}(\mathrm{Ont}_c) = 792$
- Properties in reference ontology : $\mathrm{Num}(\mathrm{Ont}_p) = 1447$

The results are summarized in the following table :

	Class	Property
Coverage	17/26 = 0,65	30/38 = 0,79
Extensiveness	766/818 = 0,94	1409/1485 = 0,95
Sparsity	1-(17/818) = 0,98	1-(30/1485) = 0,98

Chapter 7

Data Integration

This section is dedicate to the description of the data integration 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 management (semantic heterogeneity)
- Entity matching
- Data integration phase evaluation

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 Data management and entity matching

The Data Integration phase aims to build the final EG, populating the ETG previously produced with the datasets entities. In this phase the knowledge and data layer are merged together using Kermalinker and the data semantic heterogeneity is handled.

7.1.1 Datasets manipulations

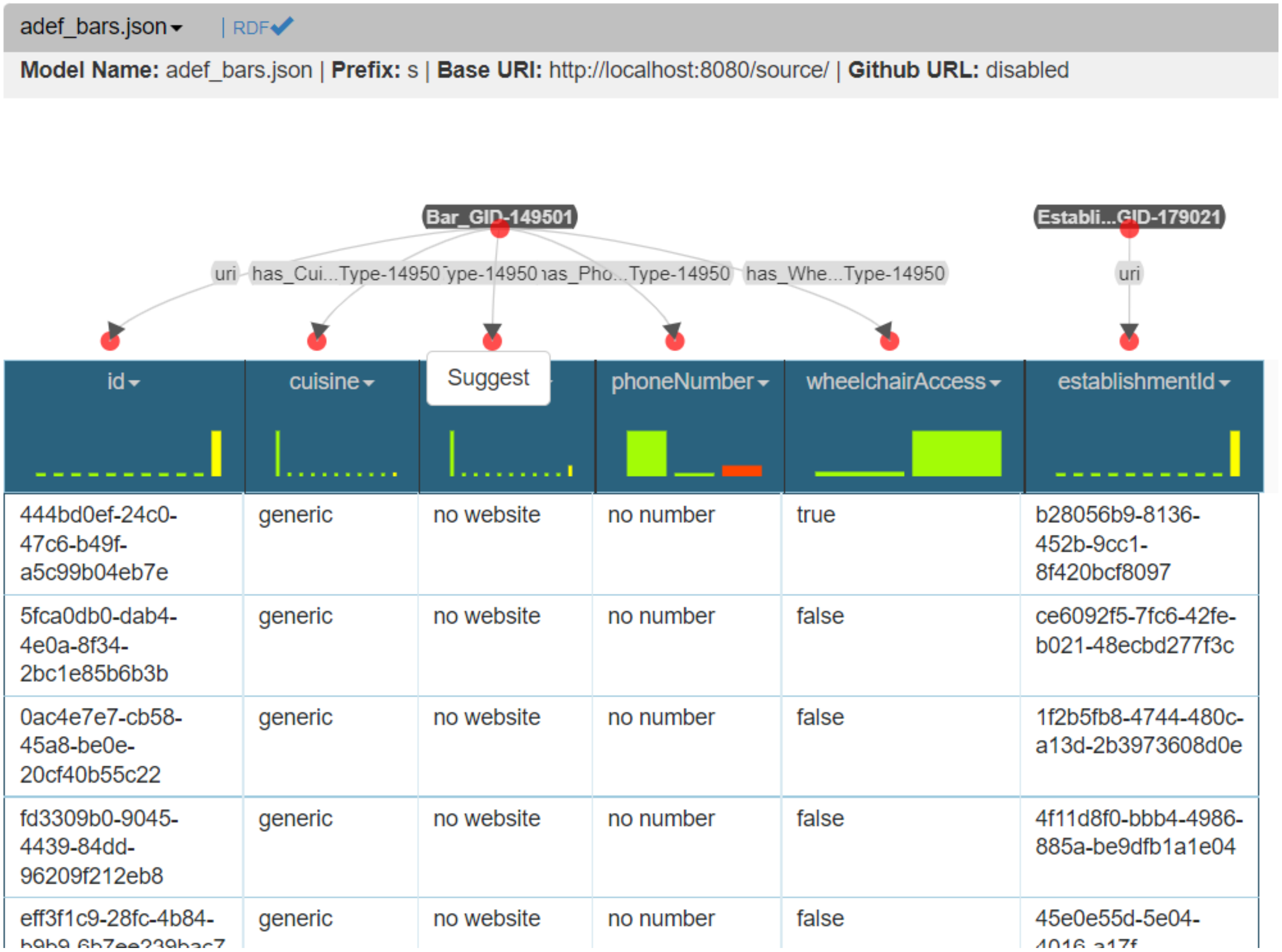
The semantic heterogeneity is a: ”consequence of the more general phenomenon of the diversity of the world and of the world descriptions.” (Giunchiglia, Fumagalli 2020).
Our datasets didn’t present the scenario where the same real word entity is represented using different properties due to the different function associated to the entity within the two datasets. Actually, since the beginning of the project, we selected our datasets keeping this issue in mind. We didn’t detect any major problems of entity alignment or matching. This is why this phase didn’t required the development of any extra code to handle the heterogeneity or merge conflicts. But it’s true that we found the same Etype in multiple datasets having different representations, a great example of this aspect is the opening hours of different places coming from different

datasets. Sometimes if an establishment was open every day, the opening hours could be {”lun”, ”mar”, ”mer”, ”jeu”, ”ven”, ”sam”, ”dim”} or {”touslesjours” (everyday) . The original code we wrote is taking care of that part by parsing and saving all the opening hours in one single array of data while processing all the datasets.

7.1.2 Karmalinker

The main tool we used during this phase is Karmalinker (click here for the link): a data linking tool allowing us to align reference data to the ontology created in the formal phase.

You will find below screenshots of the datasets aligned with the model.



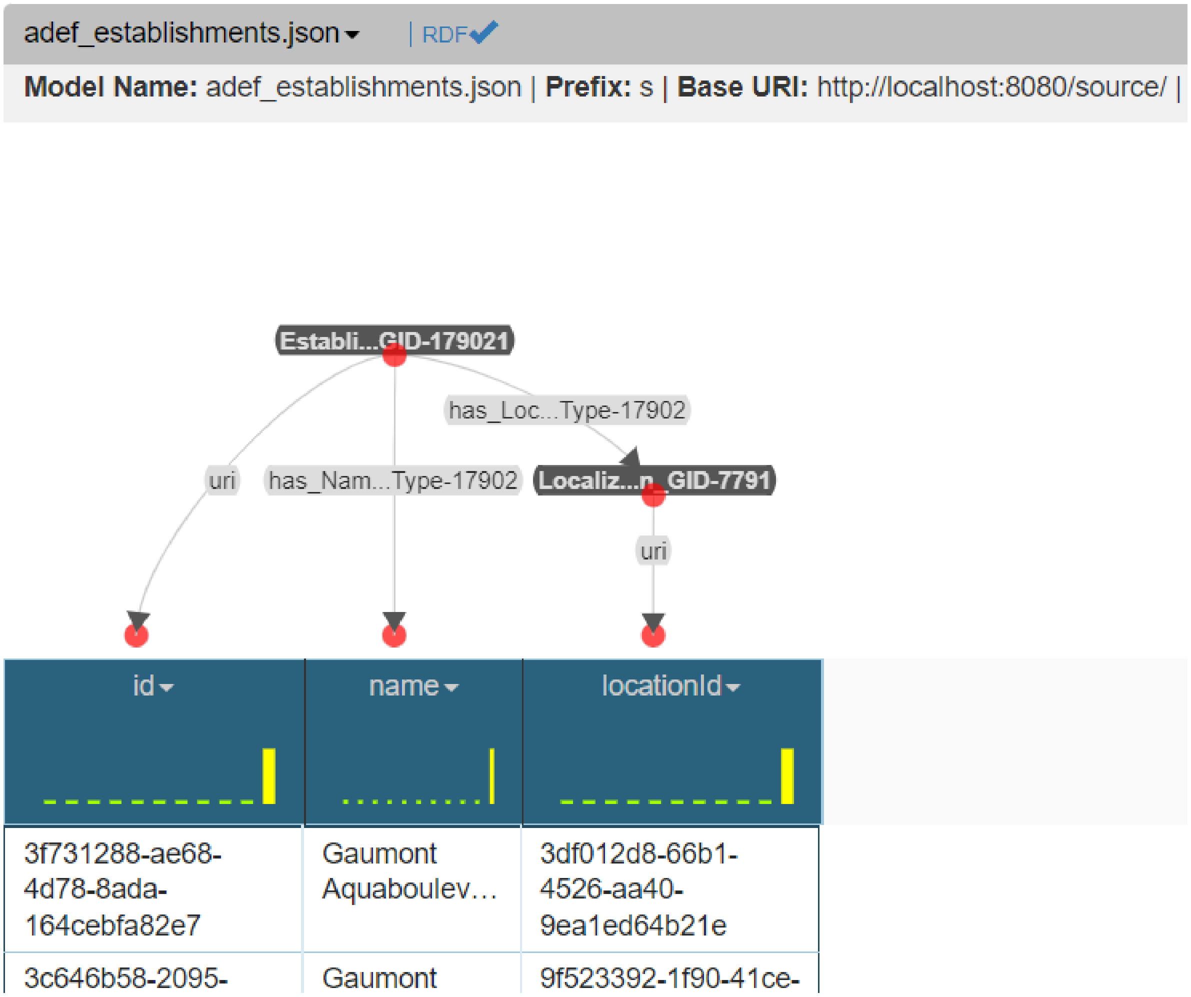
Model Name: adef_bars.json | **Prefix:** s | **Base URI:** http://localhost:8080/source/ | **Github URL:** disabled

id▾	cuisine▾	Suggest	phoneNumber▾	wheelchairAccess▾	establishmentId▾
444bd0ef-24c0-47c6-b49f-a5c99b04eb7e	generic	no website	no number	true	b28056b9-8136-452b-9cc1-8f420bcf8097
5fca0db0-dab4-4e0a-8f34-2bc1e85b6b3b	generic	no website	no number	false	ce6092f5-7fc6-42fe-b021-48ecbd277f3c
0ac4e7e7-cb58-45a8-be0e-20cf40b55c22	generic	no website	no number	false	1f2b5fb8-4744-480c-a13d-2b3973608d0e
fd3309b0-9045-4439-84dd-96209f212eb8	generic	no website	no number	false	4f11d8f0-bbb4-4986-885a-be9dfb1a1e04
eff3f1c9-28fc-4b84-	generic	no website	no number	false	45e0e55d-5e04-

Figure 7.1: Dataset about the bars in Paris, aligned with our model

7.2 Evaluation

During Data Integration phase, we need to evaluate on data level, with the proposed final EG. We aim to measure:



adef_establishments.json | RDF

Model Name: adef_establishments.json | **Prefix:** s | **Base URI:** http://localhost:8080/source/ |

id	name	locationId
3f731288-ae68-4d78-8ada-164cebfa82e7	Gaumont Aquaboulev…	3df012d8-66b1-4526-aa40-9ea1ed64b21e
3c646b58-2095-	Gaumont	9f523392-1f90-41ce-

Figure 7.2: Dataset about establishments, aligned with our model

- If the CQs in inception phase can be answered by our constructed EG. We can do evaluation based on practical applications, like SQL.
- If our collected dataset is sufficiently used. By using Sparsity to check if the dataset schema is aligned to ETG properties. Otherwise, there will be a loss of dataset information.

The calculation of the Sparsity in data integration phase is similar to the Formal ETG modelling phase, so we won't detail our steps this time. Here is the result : Sparsity(Class) = 0,98 and Sparsity(Property) = 0,98

We couldn't record the running time to test if our constructed EG can effectively solve the CQs because even if we have sparQL queries, we still need to navigate the graph manually at the end of the process. But we counted that the graph is able to answer 23/37 = 0,62 CQs completely and 8/37 = 0,22 partially.

Chapter 8

Issues

This section aims to describe any issues/problems that occurred along the DI process.

8.1 Data gathering

We managed to get information about the cinemas of Paris, but not the movies that are played. Are the beginning we wanted to write a code that would analyse the cinema's website and retrieve the daily program to update our datasets. The goal was to show that we can get data from different sources (and not only openData datasets). We also thought about retrieving the reviews for the restaurants (on TripAdvisor for example) to be able to suggest a selection of the best restaurants in an area.
But we quickly abandoned those ideas because it would have been too time consuming.

8.2 Datasets exploitation

Most of our datasets were in French, and some of them were using french abbreviations (like ”lun” for monday). It was sometimes difficult to translate concepts, for example the ”arrondissements” that are specific districts in Paris. or the little woods within the city that are called ”bois”. We did our best to correctly translate everything, but it was time consuming and we lost a bit of semantic meaning in the process.

8.3 The localisation

At the beginning of our project, we said in our scenarios that we will use the location of the user to suggest places that are close to them. But this required an extra step of developing some code that, given the coordinates of the user (given in the query), is able to process all possibilities and calculate the shortest distance, within GrapbDB. Our knowledge of SparQL isn't enough to allow us to do that.

8.4 The waterfall model

Since the DI process is an iterative process where each phase is based on the outputs of the previous one, if you find out a bit late that you made a mistake, then you will have to repeat all the steps again or carry on with a damaged product. We faced this issue a couple of times during the formal modeling and data integration phases. Since the tools we used (KOS and Karmalinker) don't allow to change just a part of the work done, we lost a lot of time just doing the same steps all over again to correct little mistakes.

Chapter 9

Outcome exploitation

The final section of the current document aims to provide a description of the data integration process outcome.

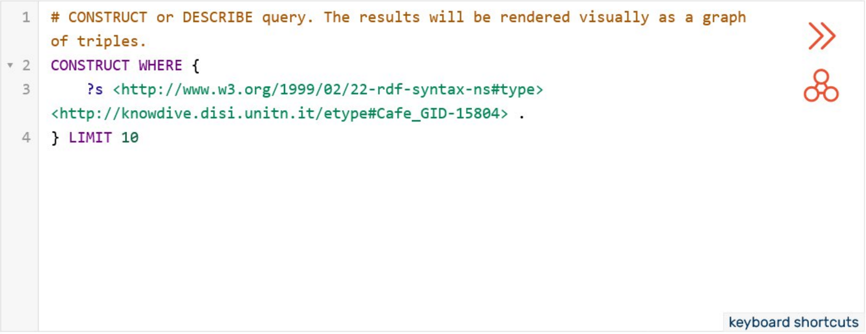
9.1 Final KG information

In this part you will find the general information about our Knowledge Graph :

- Number of Etypes : 24
- Number of object properties : 24
- Number of data properties : 91

9.2 Exploitation of the KG

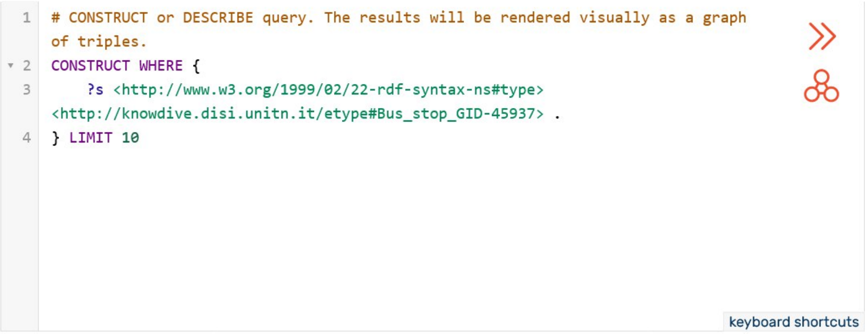
Here are examples of the SparQL queries that we wrote (they all follow this stucture) :



```
# CONSTRUCT or DESCRIBE query. The results will be rendered visually as a graph of triples.
CONSTRUCT WHERE {
    ?s <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://knowdive.disi.unitn.it/etype#Cafe_GID-15804> .
} LIMIT 10
```

Figure 9.1: Query to fetch the coffe shops' information

The rest of the KG must be navigated by hand (as seen in the presentation).



```
# CONSTRUCT or DESCRIBE query. The results will be rendered visually as a graph of triples.
CONSTRUCT WHERE {
    ?s <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://knowdive.disi.unitn.it/etype#Bus_stop_GID-45937> .
} LIMIT 10
```

Figure 9.2: Query to fetch the bus stops' information

9.3 General conclusion

In many ways, this project was challenging. We are happy that we managed to finish every phase and come up with a working Knowledge Graph. We started with a big idea in mind but we soon understood it was impossible in our time-frame to collect and integrate all the data we wanted. But our results are still reusable and we made sure that they could be easily expanded. During this project we encountered some very common issues, such as the impossibility to translate certain concepts, and by facing them ourselves we understood a lot better why we need to create methodologies such as iTelos to deal with Data Integration.

Chapter 10

Appendix

10.1 Resources

Here are the some useful resources that were used for this project:

- Data cleaning code: (click here for link)
- Inception sheet: (click here for link)
- Modeling sheet: (click here for link)
- Overpass turbo: (click here for link)
- OpenData Paris: (click here for link)

10.2 Metadata of attributes

In the following pages you can find specific information regarding the metadata of each attribute we're taking into consideration, within the datasets.

Pedestrian Areas

- **Name** [`string`] : name of the area. It's the name Parisians use to refer to the area.
- **District** [`string`] : city district of the area.
- **Geo Shape** [`geo_shape`] : geometrical shape composed by multiple geographic coordinate points.
- **Geo Point** [`geo_point`] : a single geographic point representing the location of the place.

Public Wi-Fi hotspots

- **Name** [`string`] : name of the area. It's the name Parisians use to refer to the area.
- **Address** [`string`] : the address of the hotspot.

- **Postal Code** [string] : postal code of the area in which the hotspot is located.
- **Site Code** [string] : another kind of code indicating where the hotspot is located.
- **Number of Wi-Fi terminals** [integer] : how many antennas are part of the hotspot.
- **Status** [string] : this field indicates if the Wi-Fi hotspot is operative.
- **Geo Shape** [geo_shape] : geometrical shape composed by multiple geographic coordinate points.
- **Geo Point** [geo_point] : a single geographic point representing the location of the place.

Open Air Markets

- **ID** [integer] : unique identifier of the market.
- **Short name** [string] : short name of the market.
- **Long name** [string] : long name of the market.
- **Product** [string] : type of products being sold at the market: Food, Organic food, Flowers, Birds, Stamps, Fleas, Flea market, Artistic creation.
- **District** [integer] : city district of the market.
- **Location** [string] : location of the market described in natural language.
- **Holding days** [string] : weekdays of open market.
- **Sector** [string] : sector of the market.
- **Supervisor** [string] : name of the market administrator.
- **Linear length** [integer] : maximum length of the market in meters.
- **Opening hour weekdays** [string] : opening hour during weekdays.
- **Closing hour weekdays** [string] : closing hour during weekdays.
- **Opening hour Saturday** [string] : opening hour on Saturday.
- **Closing hour Saturday** [string] : closing hour on Saturday.
- **Opening hour Sunday** [string] : opening hour Sunday.
- **Closing hour Sunday** [string] : closing hour Sunday.
- **Geo Shape** [geo_shape] : geometrical shape composed by multiple geographic coordinate points.
- **Geo Point** [geo_point] : a single geographic point representing the location of the place.

Unusual walks and Points of Interest

- **ID** [integer] : unique identifier of the PoI.
- **Address** [string] : the address of the area.
- **Postal Code** [string] : postal code of the area in which the PoI is located.
- **Associated track** [string] : Identifier of the related touristic track.
- **Image URL** [string] : URL to an image of the PoI.
- **Copyright** [string] : copyright note of the image.
- **Image caption** [string] : short caption of the image of the PoI.
- **Category** [string] : type of PoI: place-street, architectural-element, museum-cultural-place, religious heritage, etc.
- **PoI name** [string] : it's the name Parisians use to refer to the PoI.
- **Date entered** [string] : when the PoI was inserted into the dataset.
- **Keywords** [string] : comma separated keywords related to the PoI.
- **POI introductory text** [string] : short text serving as short introduction to the PoI.
- **Descriptive text** [string] : natural language text describing the PoI.
- **Official website** [string] : website of the PoI (if available).
- **Image file** [string] : image file representing the PoI.
- **Geo Shape** [geo_shape] : geometrical shape composed by multiple geographic coordinate points.
- **Geo Point** [geo_point] : a single geographic point representing the location of the place

Green areas and similar

- **ID** [integer] : unique identifier of the area.
- **Name** [string] : name of the area.
- **Type** [string] : type of area: open walks, decorative gardens, private gardens, peripheral, etc.
- **Address – Number** [integer] : the number of the area address.
- **Address – Complement** [string] : the complement of the area address (if existing).
- **Address – Street type** [string] : type of street: street, square, boulevard, avenue, etc.
- **Address – Street label** [string] : the actual name of the street related to the area.
- **Postal Code** [string] : postal code of the area in which the green area is located.

- **Calculated Area** [integer] : surface area in square meters.
- **Actual** total Area [integer] : real surface value of the area.
- **Horticultural Area** [integer] : actual green surface (with grass and other plants).
- **Closing presence** [Boolean] : true if there's someone in charge of closing the area when necessary.
- **Perimeter** [integer] : perimeter of the area.
- **Year of opening** [integer] : opening year of the area.
- **Renovation year** [integer] : renovation year of the area.
- **Former name of the green space** [string] : former name of the area (if present).
- **Year of name change** [integer] : year when the name was changed to the actual one.
- **Number of entities** [integer] : number of entities in the area.
- **Open 24h** [Boolean] : true if the area is open 24h.
- **Division ID** [integer] : ID of the division.
- **Horticultural ID** [integer] : ID of Paris horticultural area.
- **Equipment ID** [string] : the ID of the set of equipment used in the area.
- **Category** [string] : category of the area.
- **Geo Shape** [geo_shape] : geometrical shape composed by multiple geographic coordinate points.
- **Geo Point** [geo_point] : a single geographic point representing the location of the place.

Public bike stations

- **ID** [integer] : unique identifier of the bike station.
- **Name** [string] : name of the station.
- **Is Operative** [Boolean] : true if the station is working.
- **Capacity** [integer] : number of bikes the station can hold.
- **Is renting** [Boolean] : true if the station allows to rent bikes.
- **Is returning** [Boolean] : true if the station allows to return bikes.
- **Geo Point** [geo_point] : a single geographic point representing the location of the place.
- **Municipality** [string] : municipality of the station location (Paris or outskirts areas).

Cycling tracks

- **Typology** [string] : type of track.
- **Is bidirectional** [Boolean] : true if bikes can go in both ways of the track.
- **Speed limit** [integer] : speed limit of the track
- **Cycling direction** [string] : indicates it the cyclists go in the same direction as the main traffic.
- **Street** [string] : name of the street of the track.
- **District** [integer] : District in which the track is located.
- **Forest** [Boolean] : true if the track is going through the small forests in Paris.
- **Length in meters** [integer] : length of the track in meters.
- **Length in kilometers** [integer] : length of the track in kilometers.
- **Layout** [string] : indicate whether the track layout is lateral (side of the road) or axial (middle of the road).
- **Traffic prohibition** [Boolean] : true if cars cannot access the track area.
- **Track name** [string] : name of the track.
- **Bus lane type** [string] : it specifies whether busses can pass next to the track.
- **Continuity** [string] : indicates what kinds of obstacles bikers may find on the track (such as crossroads).
- **Cycling network** [string] : network of the track.
- **Opening date** [string] : first opening date of the track.
- **Geo Shape** [geo_shape] : geometrical shape composed by multiple geographic coordinate points. It represents the actual track on the map.
- **Geo Point** [geo_point] : a single geographic point representing the location of the place.

Taxi stations

- **Station number** [string] : identifier of the station.
- **Station name** [string] : It's the name Parisians use to refer to the station.
- **Address** [string] : address of the station.
- **Postal code** [string] : postal code of the area in which the station is located.
- **Taxi station phone number** [string] : phone number of the station.
- **Coordinates X_LB93** [decimal] : vertical coordinates of the station, in LB93 format.

- **Coordinates Y_LB93** [`decimal`] : horizontal coordinates of the station, in LB93 format.
- **Geo Shape** [`geo_shape`] : geometrical shape composed by multiple geographic coordinate points.
- **Geo Point** [`geo_point`] : a single geographic point representing the location of the place.

Interesting activities

- **ID** [`string`] : unique identifier of the event.
- **URL** [`string`] : official URL to the event website.
- **Title** [`string`] : title(name) of the event.
- **Lead Text** [`string`] : lead text of the event.
- **Description** [`string`] : description of the event wrapped in HTML code.
- **Category** [`string`] : category of event. It's formatted as category -> sub-category.
- **Keywords** [`string`] : comma-separated tags related to the event: Libraries, With-Families, Children, etc.
- **Start date** [`string`] : timestamp of the starting date plus time.
- **End date** [`string`] : timestamp of the ending date plus time.
- **Occurrences** [`string`] : if the event has multiple occurrences they will be found in this field.
- **Date description** [`string`] : HTML code describing the dates of the event.
- **Place name** [`string`] : name of the location/building where the event will be held.
- **Address** [`string`] : address of the place of the event.
- **Postal code** [`string`] : postal code of the area in which the event is located
- **Municipality** [`string`] : municipality of the event (Paris or outskirt areas).
- **Geographical coordinates** [`geo_point`] : a single geographic point representing the location of the place.
- **Wheelchair access** [`Boolean`] : if true, there's wheelchair access.
- **Visually impaired access** [`Boolean`] : if true, the event is available for the visually impaired.
- **Hearing impaired access** [`Boolean`] : if true, the event is available for deaf people.
- **Transport** [`string`] : public transport stations nearby the event location.
- **Contact name** [`string`] : name of the event manager.
- **Contact phone** [`string`] : phone number of the event manager.
- **Contact email** [`string`] : email of the event manager.

- **Contact URL** [string] : URL of the event manager website.
- **Associated Facebook URL** [string] : Facebook page URL for the event.
- **Associated Twitter URL** [string] : Twitter page URL for the event.
- **Type of access** [string] : indicates if the event requires booking or it's free access.
- **Reservation URL** [string] : website for the booking.
- **Reservation phone** [string] : phone number for booking.
- **Reservation email** [string] : email for booking.
- **Price type** [string] : indicates if the event requires a payment.
- **Price detail** [string] : description of the price.
- **Cover image** [file] : cover image for the event.
- **Image URL** [string] : URL of the image.
- **Image credit** [string] : credits of the image.
- **Image alt text** [string] : short description of the image.
- **Programs** [string] : programs associated with the event.
- **Date of update** [string] : last time the event was updated.

Shopping center
Data available for every field:

- **@ID** [string] : OpenStreet maps unique identifier of the place.
- **Name** [string] : name of the center.
- **Coordinates** [lat/lon] : location of the shopping center in latitude and longitude.
- **Shop** [string] : type of center: mall or dept store.

Here is some relevant data which is not available for every field:

- **Postal code** [string] : postcode of the area in which the shopping center is located.
- **Street** [string] : name of the street of the shopping center.
- **Phone number** [string] : phone number of the center.
- **Opening hours** [string] : opening hours of the center.
- **ATM** [Boolean] : true if there are ATMs in the shopping center.
- **Toilets** [Boolean] : true if there are toilets in the shopping center.
- **Wheelchair** [Boolean] : true if the center is available for wheelchair users.

Restaurants
Data available for every field:

- **@ID** [`string`] : OpenStreet maps unique identifier of the place.
- **Name** [`string`] : name of the restaurant.
- **Coordinates** [`lat/lon`] : location of the restaurant in latitude and longitude.

Here is some relevant data which is not available for every field:

- **Postal code** [string] : postcode of the area in which the restaurant is located.
- **Street** [string] : name of the street of the restaurant.
- **House number** [integer] : house number of the street of the restaurant.
- **Phone number** [`string`] : phone number of the restaurant.
- **Opening hours** [`string`] : opening hours of the restaurant.
- **Cuisine** [`string`] : type of food cooked in the restaurant.

Bus station
Data available for every field:

- **@ID** [`string`] : OpenStreet maps unique identifier of the place.
- **Name** [`string`] : name of the bus stop.
- **Coordinates** [`lat/lon`] : location of the bus stop in latitude and longitude.

Here is some relevant data which is not available for every field:

- **Shelter** [`Boolean`] : true if the station offers shelter from the weather to the pedestrians.
- **Tactile paving** [`Boolean`] : true is the station is available for the blind people.
- **Wheelchair** [`Boolean`] : true if the bus station is available for wheelchair users.
- **Route ref** [`String`] : route numbers of the busses stopping at the station.

ATM
Data available for every field:

- **@ID** [`string`] : OpenStreet maps unique identifier of the place.
- **Operator** [`string`] : The name of the bank operator.
- **Coordinates** [`lat/lon`] : location of the ATM in latitude and longitude.

Here is some relevant data which is not available for every field:

- **Opening hours** [`string`] : opening hours of the ATM.
- **Wheelchair** [`Boolean`] : true if the ATM for wheelchair users.

Subway station
Data available for every field:

- **@ID** [`string`] : OpenStreet maps unique identifier of the place.
- **Name** [`string`] : name of the subway stop.
- **Coordinates** [`lat/lon`] : location of the subway stop in latitude and longitude.

Here is some relevant data which is not available for every field:

- **Wheelchair** [`Boolean`] : true if the subway station is available for wheelchair users.

Other points of interest
Data available for every field:

- **@ID** [`string`] : OpenStreet maps unique identifier of the place.
- **Name** [`string`] : name of the place.
- **Coordinates** [`lat/lon`] : location of the place in latitude and longitude.

Here is some relevant data which is not available for every field, this data strictly related to the type of place so we're going to group them by type of place:

- **Parking** : the parking lots can have extra attributes such as: address, type (like underground), fee (if present), opening hours.
- **Bakery** : the bakeries can have extra attributes such as: address, opening hours, phone number, wheelchair access.
- **Pharmacy** : the pharmacies can have extra attributes such as: address, opening hours, phone number, wheelchair access.

Bibliography

[1] C. Virmani, S. Sinha and S. K. Khatri, ”Unified ontology for data integration for tourism sector,” 2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions) (ICTUS), 2017, pp. 152-156, doi: 10.1109/ICTUS.2017.8285995.