URBANITE

Supporting the decision-making in urban transformation with the use of disruptive technologies

Deliverable D3.4

Data Structure and Semantic Model Specification

Editor(s):	Fritz Meiners (FRAUNHOFER)
Responsible Partner:	Fraunhofer FOKUS
Status-Version:	Final – v1.0
Date:	26.12.2020
Distribution level (CO, PU):	PU

Project Number:	GA 870338
Project Title:	URBANITE

Title of Deliverable:	Data Structure and Semantic Model Specification
Due Date of Delivery to the EC:	31/12/2020

Workpackage responsible for the Deliverable:	WP3 – Data Management Platform
Editor(s):	TECNALIA, FRAUNHOFER
Contributor(s):	Sonia Bilbao (TECNALIA), Giuseppe Ciulla (ENGINEERING), Fritz Meiners (FRAUNHOFER)
Reviewer(s):	Juncal Alonso (TECNALIA)
Approved by:	All Partners
Recommended/mandatory readers:	WP4, WP5, WP6

Abstract:	This document presents the semantic model specification and common data structures used in URBANITE. This deliverable is the result of Task 3.2.
Keyword List:	Data model, FIWARE
Licensing information:	This document is licensed under Creative Commons Attribution-ShareAlike 4.0 International (CC-BY-SA 4.0) https://creativecommons.org/licenses/by-sa/4.0/
Disclaimer	This document reflects only the author's views and neither Agency nor the Commission are responsible for any use that may be made of the information contained therein

Document Description

Document Revision History

Version	Date	Modifications Introduced	
		Modification Reason	Modified by
v0.1	1/11/2020	Draft ToC	FRAUNHOFER
v0.2	4/11/2020	Updated ToC	FRAUNHOFER
v0.3	10/12/2020	All contributions ready	TECNALIA, ENGINEERING, FRAUNHOFER
v1.0	11/12/2020	Final version	FRAUNHOFER
v1.1	17/12/2020	Updated after review	TECNALIA, FRAUNHOFER
V2.0	26/12/2020	Ready for submission	TECNALIA

Table of Contents

List of Figures

List of Tables

Terms and abbreviations

API	Application Programming Interface
EC	European Commission
CC	Creative Commons
CSV	Comma Separated Values
DCAT	Data Catalogue Vocabulary
DCAT-AP	DCAT Application Profile
GIS	Geographic Information System
GML	Geography Markup Language
GPS	Global Positioning System
GTFS	General Transit Feed Specification
KPI	Key Performance Indicator
HTML	HyperText Markup Language
HTTP	HyperText Transport Protocol
JSON	JavaScript Object Notation
JSON-LD	JavaScript Object Notation Linked Data
MIF/MID	MapInfo Interchange Format
NGSI	Next Generation Service Interface
NGSI-LD	Next Generation Service Interface Linked Data
OD	Origin-Destination (matrix)
PDF	Portable Document Format
POI	Point of Interest
REST	Representational State Transfer
RDW	Specific Open Data Portal of Amsterdam
SOAP	Simple Object Access Protocol
SPDP	Standard for Publishing Dynamic Parking Data
URL	Uniform Resource Locator
UTC	Coordinated Universal Time
WFS	Web Feature Service
WMS	Web Map Service
XML	eXtensible Markup Language
XSD	XML Schema Definition

Executive Summary

This deliverable contains an analysis of the data sources collected in deliverable D3.1 [1] with regards to grouping them by domain. Based on those domains, data models for representing the relevant information have been evaluated. Out of these, the data models developed by FIWARE[1] were considered to be the most applicable in the URBANITE context. One key reason for this decision was the coherence and improved interoperability expected with nearly all data models being provided by one entity. For those domains/cases for which no data model is available in the FIWARE context, custom models have been developed. The domain-specific data models presented in this deliverable will be used for storing data in the data management platform. Additionally, a solution to modelling data required for visualization has been drafted, as well as one for representing origin-destination matrices. Overall, the topics addressed in this deliverable cover most aspects of standardising data storage and visualization models, which is expected to greatly improve interoperability between the individual tools and components utilised in the URBANITE context. Based on this, the implementation can soon commence.

[1] https://www.fiware.org/developers/data-models/

1 Introduction

Data that will be processed in the URBANITE context can vary wildly with regards to structure, purpose, and semantics. This deliverable contains an analysis of the types of data that are expected to be encountered and descriptions of applicable standards to model the data.

1.1 About this deliverable

This deliverable covers three distinct aspects of data processing in URBANITE. The largest section corresponds to the data models, which will be used to store data provided by the data sources from the pilots (WP6). As such, the basis for the corresponding analysis was deliverable D3.1, where the initial relationship of data sources in URBANITE was provided. Hence, any new types of data sources, for example, provided by the use cases at a later point in time, are not considered in the analysis. Instead, as its predecessor, this deliverable is considered a first snapshot. Changes and additions derived at a later point in time, for example during implementation, will be included in the respective deliverables, for example, D3.5 or D3.6 und lead from Fraunhofer. The same applies for the other two areas, visualization data models and Origin-Destination data models. While the first stems from FIWARE, a draft has been proposed for the latter. Both the origin-destination matrix and the visualization format may be subject to change.

1.2 Document structure

Section 2 contains a short introduction and overview over the methodology used for deriving the results presented in this deliverable. Next, the data models for representing the data provided by the data sources derived in deliverable D3.1 are covered in alphabetical order in section 3. Each data model is described with an introduction regarding its purpose, followed by a rundown of the key fields. The FIWARE KPI data model is discussed in section 4, which is followed by a proposal for a model for data visualization in section 5.

2 Methodology

In order to make informed decisions regarding the data models to be used in URBANITE, a formalised and systematic approach has to be taken. As such, this chapter outlines the decision-making involved in deriving the models described in the later parts of this document. The overall process is depicted in Figure 1.

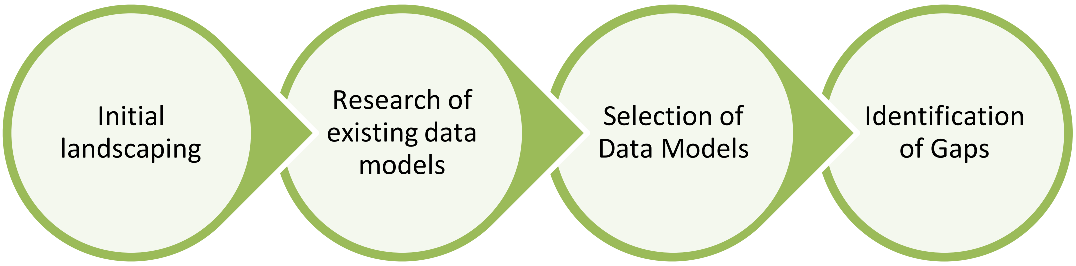

Figure 1: Process of Model Selection

2.1 Scope

Overall, the five following areas have been identified in the scope of the URBANITE data management platform, in which formal data models are applicable:

- Domain-specific data, for example, traffic data or maps,
- Data used for visualizations in the UI components,
- Representation of KPIs, to evaluate the progress or success of an activity,
- Metadata of datasets, and
- External APIs, for example between the storage and recommendation engine modules.

The Grant Agreement explicitly states that deliverable D3.4 shall be the result of task T3.2, the Data curation modules. Metadata model and API design are part of task T3.3, Data aggregation and storage, and will thus be provided as part of the documentation of deliverables D3.7 and D3.8. As such, the scope of this deliverable is limited to defining common models for the former three types of data.

2.2 Selection of Domains

The preliminary research and evaluation of the available data sources in deliverable D3.1 have revealed the nature of the data sources to be very heterogeneous. In conclusion, it was agreed that the data structure and model specification will be determined on a per-domain level. As a result, there will not be a single, global model to represent arbitrary data, but a number of data models that match the number of discrete domains relevant to the URBANITE context.

As such, the following domains were identified to be of relevance based on the analysis of the data sources in D3.1:

- Counts
- Environment
- Origin Destination Matrices
- Traffic
- Parking
- Maps and POI
- Schedules
- Weather

Additionally, the need for drafting a data model for exchanging data to be visualised in the URBANITE UI components has been identified, as well as modelling origin-destination matrices.

3 Data Source Models

Initially, research was conducted on existing standards for the domains identified in the previous step. Examples of existing ones include DATEX II for traffic data and CalDAV for schedules. However, in the case of some models (e.g. CalDAV), a number of equivalent standards exist. Also, all standards that were researched were mostly developed by independent parties under differing, partly contradicting, licenses. Therefore, in order to circumvent handling a plethora of individual standards by using distinct data models for each domain, a framework of data models covering all domains will be used in the URBANITE data management platform. This has the benefit of having a single set of standards developed by a central organisation, which reduces misunderstandings and problems among the involved parties during development. Also, the uniform approach and common basis with regards to serialisation format are expected to simplify implementation.

As such, the FIWARE "[..] curated framework of open source platform components to accelerate the development of Smart Solutions"[2] has been selected as a means to represent domain-specific data in the URBANITE data management platform. It is developed and maintained by an organisation of the same name. Its aim is to define a "[..] universal set of standards for **context data management** which facilitate the development of Smart Solutions [..]". These standards are accompanied by implementations of modules that can be used for orchestrating applications, i.e. FIWARE powered platforms.

All FIWARE data models[34] are licensed under MIT and are thus open-source. Four formats are available for serialisation; normalised NGSI-V2 and NGSI-LD[5] as well as key-value pairs for both NGSI-V2 and NGSI-LD. All formats are based on JSON and as such can be validated against provided JSON schemas. In the URBANITE context, NGSI-LD is considered to be the best fit since the metadata will also be stored as linked data (in accordance with DCAT-AP). Therefore, both the data and metadata can profit from a homogeneous infrastructure, which is thought to reduce the overall complexity of the data management platform. Whether the normalised or key-value version of NGSI-LD will be used depends on the compatibility and capabilities of the processing entities in both the data management platform and the recommendation engine.

Table 1**¡Error! No se encuentra el origen de la referencia.** shows the mapping between the domains derived in section 2.2 and the corresponding FIWARE data models, which mostly originate from the SmartCities[6] domain. FIWARE does not offer a direct data model for counts. Instead, the counts are included in the transportation group. Depending on the level of detail provided by the various data sources, the respective FIWARE models may not be able to store all information. In these cases, new FIWARE models will have to be created or existing ones extended. An example of the former is the electromagnetic noise readings provided by the Messina use case. In any case, it is planned to feed the results of such measures back to FIWARE, thereby directly contributing to the open-source project. If accepted, this would have the added benefit of the contributions being officially verified and the URBANITE project not relying on unofficial forks.

[2] https://www.fiware.org/
[3] https://fiware-datamodels.readthedocs.io/en/latest/howto/index.html
[4] https://github.com/smart-data-models
[5] https://www.etsi.org/deliver/etsi_gs/CIM/001_099/009/01.01.01_60/gs_CIM009v010101p.pdf
[6] https://github.com/smart-data-models/SmartCities

Table 1. Domain Data Model Mapping

Domain	Data Model
Counts	FIWARE Transportation/CrowdFlowObserved and Transportation/TrafficFlowObserved
Environment	FIWARE Environment[7]
Origin Destination Matrix	Own specification
Parking	FIWARE Parking[8]
Point of Interest	FIWARE POI[9]
Schedules	FIWARE Urban Mobility[10]
Traffic	FIWARE Transportation[11]
Weather	FIWARE Weather[12]

3.1 Counts

See section 3.6 (transportation).

3.2 Environment

3.2.1 Introduction

Environmental factors can play a big part in urban mobility planning. Awareness of citizens for pollution in cities, for example stemming from car exhausts fumes, is rising. In cases of exceeding air pollution temporary or permanent bans for some or all vehicles may be required. Therefore, environmental data is vital for recommending sound policies on urban mobility planning.

3.2.2 Format

The FIWARE Environment group[7] provides classes for the most common types of pollution, for example AirQuality, WaterQuality, and NoiseLevels. However, pollution originating in electromagnetic radiation, as provided by the Messina Use Case, is not covered. Therefore, a custom data model based on the existing FIWARE models has been developed, as shown below. **It is planned to contribute to the electromagnetic pollution data model to FIWARE for incubation**. Note that unlike the other data models covered in this document, the Environment data models are not part of the SmartCities group.

[7] https://github.com/smart-data-models/dataModel.Environment
[8] https://github.com/smart-data-models/dataModel.Parking
[9] https://github.com/smart-data-models/dataModel.PointOfInterest
[10] https://github.com/smart-data-models/dataModel.UrbanMobility
[11] https://github.com/smart-data-models/dataModel.Transportation
[12] https://github.com/smart-data-models/dataModel.Weather

3.2.2.1 Electro Magnetic Noise Observed

3.2.2.1.1 Description

It represents an observation of those electromagnetic parameters that estimate electromagnetic wave levels at a certain place and time. This entity is primarily associated with the Smart City and environment vertical segments and related IoT.

3.2.2.1.2 Data Model

The data model is defined as shown below:

- id : Unique identifier.
- type : Entity type. It must be equal to ElectromagneticNoiseObserved.
- source : A sequence of characters giving the source of the entity data.
  - Attribute type: Property. Text or URL
  - Optional
- dataProvider : Specifies the URL to information about the provider of this information
  - Attribute type: Property. URL
  - Optional
- dateCreated : Entity's creation timestamp.
  - Attribute type: Property. DateTime
  - Read-Only. Automatically generated.
- dateModified : Last update timestamp of this entity.
  - Attribute type: Property. DateTime
  - Read-Only. Automatically generated.
- location : Location of this observation represented by a GeoJSON geometry.
  - Attribute type: GeoProperty. geo:json.
  - Normative References: https://tools.ietf.org/html/rfc7946
  - Mandatory if address is not present.
- address : Civic address of this observation.
  - Attribute type: Property. Address
  - Normative References: https://schema.org/address
  - Mandatory if location is not present.
- name : Name given to this observation.
  - Attribute type: Property. Text
  - Normative References: https://uri.etsi.org/ngsi-ld/name equivalent to name
  - Optional
- description : Description given to this observation.
  - Attribute type: Property. Text
  - Normative References: https://uri.etsi.org/ngsi-ld/description equivalent to description
  - Optional

- dateObserved : The date and time of this observation represented by an ISO8601 interval. As a workaround for the lack of support of Orion Context Broker for datetime intervals, it can be used two separate attributes: dateObservedFrom, dateObservedTo.
  - Attribute type: Property. ISO8601 interval represented as Text.
  - Optional
- dateObservedFrom : Observation period start date and time. See dateObserved.
  - Attribute type: Property. DateTime.
  - Mandatory
- dateObservedTo : Observation period end date and time. See dateObserved.
  - Attribute type: Property. DateTime.
  - Mandatory
- refDevice : A reference to the device which captured this observation.
  - Attribute type: Relationship. Reference to an entity of type Device
  - Optional
- measurementType : Type of measurement performed
  - Attribute type: Property. Text
  - Allowed values: one of ("broadband", "frequency selective")
  - Optional
- sensorType : Type of sensor used
  - Attribute type: Property. Text
  - Allowed values: one of ("active", "passive")
  - Optional
- refPointOfInterest : A reference to a point of interest associated with this observation.
  - Attribute type: Relationship. Reference to an entity of type PointOfInterest
  - Optional

3.2.2.1.3 Representing electromagnetic parameters

The number of electromagnetic parameters measured can vary. *For each* electromagnetic measurand there *MUST* be an attribute which name *MUST* be exactly equal to the electromagnetic measurand name, as follows:

- Attribute name: Equal to the name of the measurand, for instance Veff.
- Attribute type: Property. Number
- Attribute value: corresponds to the value for the measurand as a number
- Attribute Metadata:
  - description: short description of the measurand. (optional)
  - Attribute type: Property. Text
  - Normative References: https://uri.etsi.org/ngsi-ld/name equivalent to name

3.3 Origin-Destination (OD) Matrix

3.3.1 Introduction

As described in deliverable D4.1, section "3.1. Transport demand model estimation", knowing the exact way public transportation is being used by citizens within a given city has crucial importance to take the correct strategic and/or operational decisions.

Origin-destination (OD) matrices are the most precise and complete structure that describes how people use a given transportation system or in other words, the transport demand estimation. These matrices represent the movement of people in a certain area, from an origin (O) to a destination (D). OD datasets contain details of trips between two geographic points or, more commonly, zones (which are often represented by a zone centroid). Each cell in the matrix represents the number of trips from the origin (row) to the destination (column).

FIWARE does not provide any data model for OD matrices, so a new data model has been proposed in URBANITE based on GeoJSON, an open standard format designed for representing simple geographical features.

3.3.2 Format

The format proposed to represent an OD matrix includes the type of transport used in the trips, the type of temporal aggregation and the definition of the zones in GeoJSON format. Latitude and longitude values are recommended to represent the coordinates. An example is shown in Figure 2.

```
{
  "zones": {
    "type": "FeatureCollection",
    "features": [{
      "type": "Feature",
      "geometry": {
        "type": "Polygon",
        "coordinates": [
          [[-2.479450507052775,42.96585925374888],
                  [-2.482574883775718,42.96478608962898],
                  [-2.485137285765441,42.964988424544806],
                  [-2.488221286691052,42.96431647509881] ]
        ]},
        "properties": {
          "zone_id": 111,
          "zone_name": "centre"
        }
      }, {
        "type": "Feature",
        "geometry": {
          "type": "Polygon",
          "coordinates": [
            [[-2.53412011768046,43.03983957051575],
             [-2.532387862764252,43.04095880319205],
             [-2.530063495663586,43.0449596547428],
             [-2.526706187554225,43.04567237183818] ]
            ]
        },
        "properties": {
          "zone_id": 199,
          "zone_name": "old_quarters"
        }
      }]
  },
  "type-transportation": "bus",
  "type-time-aggregation": "daily",
  "data": {
    "period": {
      "ini": "10:00:00",
      "end": "11:00:00"
    },
    "mat": [{
      "orig": 111,
      "dest": 111,
      "val": 2001
    }, {
      "orig": 111,
      "dest": 112,
      "val": 230
    }, {
      "orig": 199,
      "dest": 199,
      "val": 1001
    }]
  }
}
```


Figure 2: Example of an Origin-Destination Matrix

3.4 Parking

3.4.1 Introduction

When planning urban mobility policies, the pressure on a city's limited parking spots must be accounted for. Raising ticket prices, declaring new areas for parking or diverting traffic to other parts of town can prove to be viable measures to reduce mobility related issues. In the scope of the URBANITE project, the use cases of Bilbao and Helsinki provide parking related information.

3.4.2 Format

In order to cover this area, the FIWARE data models for storing static parking possibilities, namely Off- and OnStreetParking, ParkingGroups (e.g. parking lots), ParkingSpot, and ParkingAccess will be employed. All of these are similar with regards to their fields. In addition to the obvious data points like location/address and number of spots, the models are also capable of storing information regarding allowed vehicle types and maximum allowed parking duration, among others. At the time of writing, it is expected that this data model will sufficiently represent all information provided by the data sources.

3.5 POI

3.5.1 Introduction

Points of Interest are relevant to urban mobility planning in a variety of ways. For example, venues attracting a large number of guests in a short time frame have a big impact on the utilisation of local transport opportunities. Likewise, POIs relevant to specific means of transportation, for example filling stations for road vehicles, can significantly change mobility patterns.

3.5.2 Format

Unlike the other FIWARE data models covered in this document, the POI data model is a little too verbose for the purposes of the URBANITE context. It is designed to explicitly represent specific entities like Museums and Beaches. However, this level of granularity contradicts the approach of URBANITE in which domain-specific data models shall be applied, but not overly specific to branch out into a plethora of different data models. This would defeat the purpose of generalisation aimed for in an attempt to foster frictionless data exchange.

As such, the only data model in the POI group that will be used (as of now) is the data model of the same name, i.e. PointOfInterest. As is to be expected, the POI model is a rather general one, featuring fields like name, location, and category. The latter can be ones defined in the factual taxonomy[13].

[13] https://github.com/Factual/places/blob/master/categories/factual_taxonomy.json

3.6 Transportation

3.6.1 Introduction

The transportation domain covers all data related to transport on public infrastructure. In the URBANITE context, one of the main types of data in this domain is the traffic flow measurementnd metrics regarding road usage. However, more static information like the position of electrical car charging and bike-sharing stations is also part of this domain. Schedules of public transportation are explicitly not covered in this domain, see section 3.7 for more information on this kind of data.

3.6.2 Format

Based on the share of data sources providing traffic flow data, the TrafficFlowObserved[14] entity will be one of the key data models. It makes use of the Vehicle and RoadSegment[15] data models to specify which vehicle types were recorded at which location. The CrowdFlowObserved[16] data model serves the purposes of storing pedestrian counts. A number of properties are shared with its counterpart TrafficFlowObserved, for examples specification of locations via the RoadSegment model or stating how spread-out individuals are. The actual count is stored as a simple integer. At the time of writing two fleet related data models are still in incubation, i.e. development.

In addition to the obvious stats like address, opening times, and capacity the EVChargingStation[17] data model also covers more technical features like available socket types and voltage. Likewise, the BikeHireDockingStation model can also store information like location and available slots/bikes.

[14] https://fiware-datamodels.readthedocs.io/en/latest/Transportation/TrafficFlowObserved/doc/spec/index.html
[15] https://fiware-datamodels.readthedocs.io/en/latest/Transportation/RoadSegment/doc/spec/index.html
[16] https://github.com/smart-data-models/dataModel.Transportation/tree/master/CrowdFlowObserved
[17] https://github.com/smart-data-models/dataModel.Transportation/tree/master/EVChargingStation

3.7 Urban Mobility

3.7.1 Introduction

Urban Mobility data models[18] are used to map GTFS[19] to FIWARE NGSI entities (an NGSI entity is an instance of an NGSI data model). GTFS files are published by public transit agencies and provide details of schedules and paths of transportation means. Each GTFS file is a ZIP archive containing a collection of different text files[20] structured following CSV format[21]. Each Urban Mobility data model maps to a specific GTFS feed text file, as shown in Table 2.

Table 2: GTFS to Urban Mobility Mapping

GTFS feed text file	Urban Mobility Data Model
agency.txt	GtfsAgency[22]
stops.txt	GtfsStop[23], GtfsStation[24], GtfsAccessPoint[25]
routes.txt	GtfsRoute[26]
trips.txt	GtfsTrip[27], GtfsService[28]
stop_times.txt	GtfsStopTime[29]
calendar.txt	GtfsCalendarRule[30]
calendar_dates.txt	GtfsCalendarDateRule[31]
shapes.txt	GtfsShape[32]
frequencies.txt	GtfsFrequency[33]
transfers.txt	GtfsTransferRule[34]

[18] https://fiware-datamodels.readthedocs.io/en/latest/UrbanMobility/doc/introduction/index.html
[19] https://developers.google.com/transit/gtfs
[20] https://developers.google.com/transit/gtfs/reference
[21] https://tools.ietf.org/html/rfc4180
[22] https://fiware-datamodels.readthedocs.io/en/latest/UrbanMobility/GtfsAgency/doc/spec/index.html
[23] https://fiware-datamodels.readthedocs.io/en/latest/UrbanMobility/GtfsStop/doc/spec/index.html
[24] https://fiware-datamodels.readthedocs.io/en/latest/UrbanMobility/GtfsStation/doc/spec/index.html
[25] https://fiware-datamodels.readthedocs.io/en/latest/UrbanMobility/GtfsAccessPoint/doc/spec/index.html
[26] https://fiware-datamodels.readthedocs.io/en/latest/UrbanMobility/GtfsRoute/doc/spec/index.html
[27] https://fiware-datamodels.readthedocs.io/en/latest/UrbanMobility/GtfsTrip/doc/spec/index.html
[28] https://fiware-datamodels.readthedocs.io/en/latest/UrbanMobility/GtfsService/doc/spec/index.html
[29] https://fiware-datamodels.readthedocs.io/en/latest/UrbanMobility/GtfsStopTime/doc/spec/index.html
[30] https://fiware-datamodels.readthedocs.io/en/latest/UrbanMobility/GtfsCalendarRule/doc/spec/index.html
[31] https://fiware-datamodels.readthedocs.io/en/latest/UrbanMobility/GtfsCalendarDateRule/doc/spec/index.html
[32] https://fiware-datamodels.readthedocs.io/en/latest/UrbanMobility/GtfsShape/doc/spec/index.html
[33] https://fiware-datamodels.readthedocs.io/en/latest/UrbanMobility/GtfsFrequency/doc/spec/index.html
[34] https://fiware-datamodels.readthedocs.io/en/latest/UrbanMobility/GtfsTransferRule/doc/spec/index.html

It is important to underline that these data models do not cover the entire GTFS feed text files specification. Moreover, these additional data models are included to properly map GTFS entities to NGSI ones: PublicTransportRoute[35], PublicTransportStop[36] and ArrivalEstimation[37]

3.7.2 Format

To map each GTFS feed text files into the proper Urban Mobility data model, being compliant with NGSI specification, each field contained into its relative text file is mapped to the specific NGSI field taking advantage of the Linked GTFS[38] specification. Moreover, additional fields are also provided to properly support NGSI format. For instance, the *agency_name* field, contained in the *agency.txt* file, maps to the GtfsAgency's *name* field, taking advantage of the relative NGSI's *name* property definition that maps *foaf:name* LinkedGTFS definition.

Furthermore, to properly manage conditionals properties reported into GTFS specifications, different data models can be provided for the single GTFS feed text file. An example is the mapping of *stops.txt* to the GtfsStop, GtfsStation and GtfsAccessPoint data models, as shown in Table 2. This is due to the fact that *stops.txt* contains the *location_type* field that represents different types of stop. In the context of Urban Mobility data models:

- **GtfsStop**: maps the stop where *location_type* equals 0. In GTFS context it represents a location for boarding and disembarking of passengers; it is also called platform if linked to a station.
- **GtfsStation**: maps the stop where *location_type* equals 1. In GTFS context, it represents a station containing one or more platform.
- **GtfsAccessPoint**: maps the stop where *location_type* equals 2. In GTFS context, it represents a location used to enter or exit a station from the street.

[35] https://github.com/smart-data-models/dataModel.UrbanMobility/tree/master/PublicTransportRoute
[36] https://github.com/smart-data-models/dataModel.UrbanMobility/tree/master/PublicTransportStop
[37] https://github.com/smart-data-models/dataModel.UrbanMobility/tree/master/ArrivalEstimation
[38] https://github.com/OpenTransport/linked-gtfs/blob/master/spec.md

3.8 Weather

3.8.1 Introduction

The weather has a great influence on the type of transportation citizens will choose for their journeys. For example, despite heavy traffic, people may tend to pick a car over a bicycle when the rain is forecasted. As such, it is important to incorporate weather conditions when recommending policies for urban mobility.

3.8.2 Format

The FIWARE weather data model[39], which will be used in the URBANITE context for representing weather data, incorporates three distinct formats. WeatherAlert is a container for the information required for warning citizens of approaching dangerous weather conditions. The WeatherForecast model can be used to encode data regarding predictions of upcoming weather conditions. WeatherObserved contains data about weather conditions that were present at a certain place and time. It is expected that neither of these models will be extended for the project.

The latter two entities share properties directly related to weather, for example, temperature and humidity readings. While the former contains one set of readings linked to a specific date-time and coordinates, the latter contains expected minimum and maximum readings for a given date. Instead of coordinates, an address is provided for which the forecast is issued.

A WeatherAlert is a type of Alert with the category "Weather" being assigned. As such, it does not contain any weather-related readings. Instead, Alert properties like severity are present. The goal here is first and foremost to make people aware of some weather hazard.

[39] https://github.com/smart-data-models/dataModel.Weather

4 KPI Data Model

4.1 Introduction

KPIs (key performance indicators) allow measuring, evaluating and visualising the results of urban mobility projects by quantifying the progress in different categories or divisions. This is crucial for decision-making.

Some examples of urban KPIs that could be monitored in the project as a result of WP4's data analysis are as follows:

- Number of electric vehicle charging stations and hours of usage per day/month/year
- Number of public Wi-Fi zones and the number of connected users
- Number of public transportation journeys
- Kilometres of bike lanes per 100,000 inhabitants
- Kilometres travelled with public bikes
- Noise pollution levels

4.2 Format

The FIWARE data model[40] is flexible enough to accommodate for different usage scenarios: An entity per KPI calculation or a unique entity per KPI which value evolves along time.

The data model contains four mandatory fields: the indicator's name, the organisation which evaluated the KPI, the frequency in which the KPI is calculated (hourly, daily, weekly, monthly, yearly, quarterly, bimonthly, biweekly), the value of the KPI and the indicator's category (quantitative, qualitative, leading, lagging, input, process, output, practical, directional, actionable, financial).

[40] https://fiware-datamodels.readthedocs.io/en/latest/KeyPerformanceIndicator/doc/spec/index.html

5 Visualization Data Model

5.1 Introduction

The visualization of information plays a relevant role in decision-making processes; different data formats and data structures can be used to build data visualizations, depending on the specific needs, the purpose of the visualization itself and the technological tools used to build the visualizations. Since the most appropriate data formats to build visualizations may vary on the basis of different needs, the aim of this section is to provide some non-exhaustive examples of data structures that can be used to visualise information.

5.2 Format

Among data structures to represent information in a suitable manner to build visualization, two can be mentioned: tree data structures and table data structures.

A table data structure can be used to represent data through rows and columns and is more suitable for large ordered sets of data.

A tree data structure can be used to represents data in a hierarchical manner on the basis of parent-child relationships. This kind of structure can better support the realisation of visualization through which is could be possible to explore and navigate the data.

Both of these data structures can be realised from a technological perspective with different formats, such as JSON (JavaScript Object Notation), JSON-LD (JavaScript Object Notation for Linked Data), CSV (Comma-Separated Values) or XML (eXtensible Markup Language). In the context of URBANITE, NGSI-LD defines the rules to be followed for the definition of the specific data models to represent data to be visualised.

Furthermore, in the context data model for representing and visualising geographic information GeoJSON[41] is one of the data models most widely used. A GeoJSON represents a single spatial feature or a set of spatial features together with additional metadata and information. The basic features that can be represented through a GeoJSON are the following: Points, Lines or Polygons. In 2017, a new version of GeoJSON based on JSON-LD (GeoJSON-LD[42]) has been defined. NGSI-LD specifications comprise the possibility to include geospatial properties; these shall be represented using GeoJSON Geometries. Availability of GeoJSON-LD opens new possible scenarios to be investigated concerning the possibility to adopt GeoJSON-LD (instead of GeoJSON) for including geospatial properties within NGSI-LD data models.

[41] https://geojson.org/
[42] https://geojson.org/geojson-ld/

6 Conclusion

The aim of this deliverable was to specify models for storing and exchanging data in the URBANITE platform. As such, a first analysis of the type of data that will be encountered has been performed. Based on this, three main fields have been identified. First, models for the heterogeneous data provided by the various data sources collected in deliverable D3.1 have been presented. Next, models for representing both KPIs and visualization data have been described.

In conclusion, a main goal of not reinventing the wheel has mostly been achieved. Nearly all models stem from the widely established FIWARE framework. A high level of standardisation allows reusing existing implementations and ensures a solid quality of documentation, which benefits all parties involved. However, some of the data encountered required drafting custom models, namely origin-destination matrices and visualization. Since implementation has not commenced at the time of writing, all models may still be adapted or new ones introduced, should the circumstance require it. For reference, the main URBANITE data models are summarised in Table 3.

Table 3: URBANITE Data Models

Domain	Model(s)	Comment
Counts[43][44]	FIWARE CrowdFlowObserved FIWARE TrafficFlowObserved	Will need to be extended for counts not related to pedestrians or traffic.
Environment[45]	FIWARE AeroAllergenObserved FIWARE AirQualityObserved FIWARE NoiseLevelObserved FIWARE WaterObserved FIWARE WaterQualityObserved URBANITE ElectromagneticNoiseObserved	ElectroMagneticNoiseObserved shall be contributed to FIWARE for incubation.
Key Performance Indicator[46]	FIWARE KeyPerformanceIndicator	
Origin-Destination Matrix	URBANITE OriginDestinationMatrix	
Parking[47]	FIWARE OffStreetParking FIWARE OnStreetParking FIWARE ParkingAccess FIWARE ParkingGroup FIWARE ParkingSpot	
Point of Interest[48]	FIWARE PointOfInterest	The other data models may be included at a later point in time.

[43] https://github.com/smart-data-models/dataModel.Transportation/tree/master/CrowdFlowObserved
[44] https://github.com/smart-data-models/dataModel.Transportation/tree/master/TrafficFlowObserved
[45] https://github.com/smart-data-models/dataModel.Environment
[46] https://github.com/smart-data-models/dataModel.KeyPerformanceIndicator
[47] https://github.com/smart-data-models/dataModel.Parking
[48] https://github.com/smart-data-models/dataModel.PointOfInterest

Transportation[49]	FIWARE BikeHireDockingStation FIWARE EVChargingStation FIWARE Road FIWARE RoadSegment FIWARE Vehicle	The models Crowd- and TrafficFlowObserved are used for the Counts domain.
Urban Mobility[50]	FIWARE ArrivalEstimation FIWARE GtfsAccessPoint FIWARE GtfsAgency FIWARE GtfsCalendarDateRule FIWARE GtfsCalendarRule FIWARE GtfsFrequency FIWARE GtfsRoute FIWARE GtfsService FIWARE GtfsShape FIWARE GtfsStation FIWARE GtfsStop FIWARE GtfsStopTime FIWARE GtfsTrip FIWARE PublicTransportRoute FIWARE PublicTransportStop	
Weather[51]	FIWARE WeatherAlert FIWARE WeatherForecast FIWARE WeatherObserved	
Visualization	NGSI-LD as tree data structure NGSI-LD as table data structure	

[49] https://github.com/smart-data-models/dataModel.Transportation
[50] https://github.com/smart-data-models/dataModel.UrbanMobility
[51] https://github.com/smart-data-models/dataModel.Weather

7 References

[1] URBANITE, "D3.1. URBANITE Mobility Data Sources Analysis," European Commission, 2020.