S.LI.DES
Smart strategies for sustainable tourism in LIvely cultural DEStinations

2014 - 2020 Interreg V-A
Italy - Croatia CBC Programme
Priority Axis: Environment and cultural heritage
Specific objective: 3.1 - Make natural and cultural heritage a leverage for sustainable and more balanced territorial development

Deliverable 3.2.2.
Now-casting models

Work Package:	**3 - The S.LI.DES Smart Destination Ecosystem**
Activity:	**2 - Developing the visitors' mobility models: now-casting, forecasting and simulations**
Responsible Partner:	**INSTITUTE FOR TOURISM**
Partners involved:	**LP – University of Cà Foscari (IT)** **PP1 - CISET (IT)** **PP2 - Ecipa (IT)** **PP3 - SIPRO Ferrara (IT)** **PP4 - City of Bari (IT)** **PP5 - City of Venice (IT)** **PP6 –CAST-University of Bologna (IT)** **PP7 – Institute for Tourism** **PP8- Craft College- Institution for adult education Subsidiary Rijeka** **PP9- Development Agency of the City of Dubrovnik-Dura** **PP10-Sibenik Tourist board**

Version:	**Final V.01**	**Date**	**03/30/2021**
Type:	**Report**		
Availability:	**Public**		
Editors:	**Institute for Tourism, University of Cà Foscari, and CAST-University of Bologna**		

Table of Contents

1. The Visitor Mobility Models

The CAST UNIBO partner has developed dynamical models to simulate the pedestrian flows on a road network integrating statistical data on tourist flows and the real time data recorded by the SLIDES experimental campaigns. The models are conceived to take into account:

1) The existence of attractive points of interest (POI) in the considered areas that determine the tourist mobility demand and the duration of the visit;
2) The interaction between the tourist flows and the mobility of local city users (resident and daily commuters) are present in the same area;
3) The availability of statistical data collected in the data hub on the expected tourist presences during the different periods of the year and new data collected by the experimental campaigns that have been realized by each city using distributed sensors and that also allow real time data collection.

The models have been focused on the simulation of pedestrian and public means mobility considering that the historical centres of the cities involved in the project are areas with strong traffic restrictions and the tourists perform manly a pedestrian mobility. The model outputs provide dynamical information on the evolution of the pedestrian flows on the road network (Dynamics mobility maps) and the distribution of presences in the considered area or on specific POI during a specific day or a ‘representative’ day for a chosen period (heat maps and plot of the presence evolution). In the Nowcasting mode the model simulation reproduces the evolution of the mobility flows during a specific day integrating the data collected by the sensors and the statistical mobility data performing a now-casting procedure that extends the information recorded at specific point to the whole considered area. The now casting models are useful to cope with the problems to understand how the tourists move in the city in different periods, to evaluate the effects of the new initiatives, to correlate the observed tourist flows with the information from other data sets (i.e., social media) or to verify the effects of policies. A technical description of the nowcasting model implementation is given in the following section and other detail are given in the simulation report D 3.5. The models represent the road network of the considered area using the Cartography data from openstreetmap.org that is associate to a mathematical graph. In Figure 1 we show the example for the Venice historical centre (other examples are in the simulation report D 3.5).

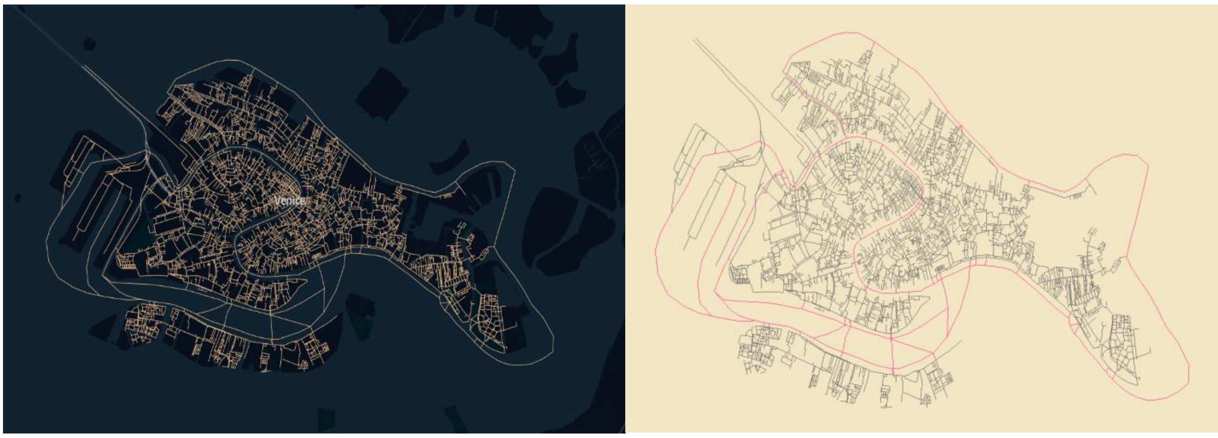


Figure 1: The road network of the Venice historical centre as provided by the Open Street Maps (https://www.openstreetmap.org) (left picture) and as implemented in the visitors' dynamical models (right picture).

A first study of the graph is performed to introduce a weight to each road that defines some preferred paths for the tourist flows. These paths are computed considering the tourist relevance of the roads as inferred from statistical data on the presence of tourist attractions or previous studies of the tourist mobility. The Figure 2 (top) shows an example of tourist preferred roads computed for the Venice historical centre. When a 'virtual tourist' realizes his mobility demand, the choice of a road at a crossing point considers the road weight in a probabilistic way. Therefore, the model creates tourist preferred paths from the entry points to the main attraction points. As an example, we show the preferred paths from the entry points in Venice towards San Marco Square (Figure 2-down).

Figure 2: (top) the road subnetwork that is considered preferred by the tourists in the mobility models for the Venice historical centre; (down) preferred paths simulated by the model from the entry points towards San Marco Square.

The mobility models perform the following steps to simulate the tourist flows:

1) The population distribution is divided in the three main categories Tourists, Commuters and Residents, according to statistical historical data and the model associates a mobility demands to each category according to the distribution of the attraction points, the activities present in the area. In particular, the

tourist mobility demand depends on the visit time budget and the distance of the attraction points: previous studies have shown that the mobility path lengths and visit duration of tourists have a characteristic distribution (see Figure 3 for an example) We use these universal statistical laws to define the mobility agenda associating to each visitors a sequence of attractions present in the area that is consistent with the expected visit duration and the mobility path lengths. The attractions are divided in two categories: the main attractions (denote by the capital letter A in the input data: see next section) and the secondary attractions (denotes by B,C letters): these categories have been discussed with the other partners. The mobility agenda are firstly built using the main attractions (that are associated to long visit times) and then completed by the secondary attractions nearby the main ones. The agenda takes also into account the opening times of attractions and the requires visit times. The sequences that define the daily mobility agenda of visitors start and end at the same point since we assume that the visitors exit from the considered area at the same entry point. The commuters are intrdiced in the simulation through the entry points and they move towards a destination randomly chosen in the road network. Finally, the resident mobility is simulated as a random mobility on the road network according to the circadian rhythms. In Fig. 4 we show the expect path mobility length associated by the model to the visitors in Venice and the number of attractions inserted in the mobility agenda; more details are given in the simulations report D 3.5.

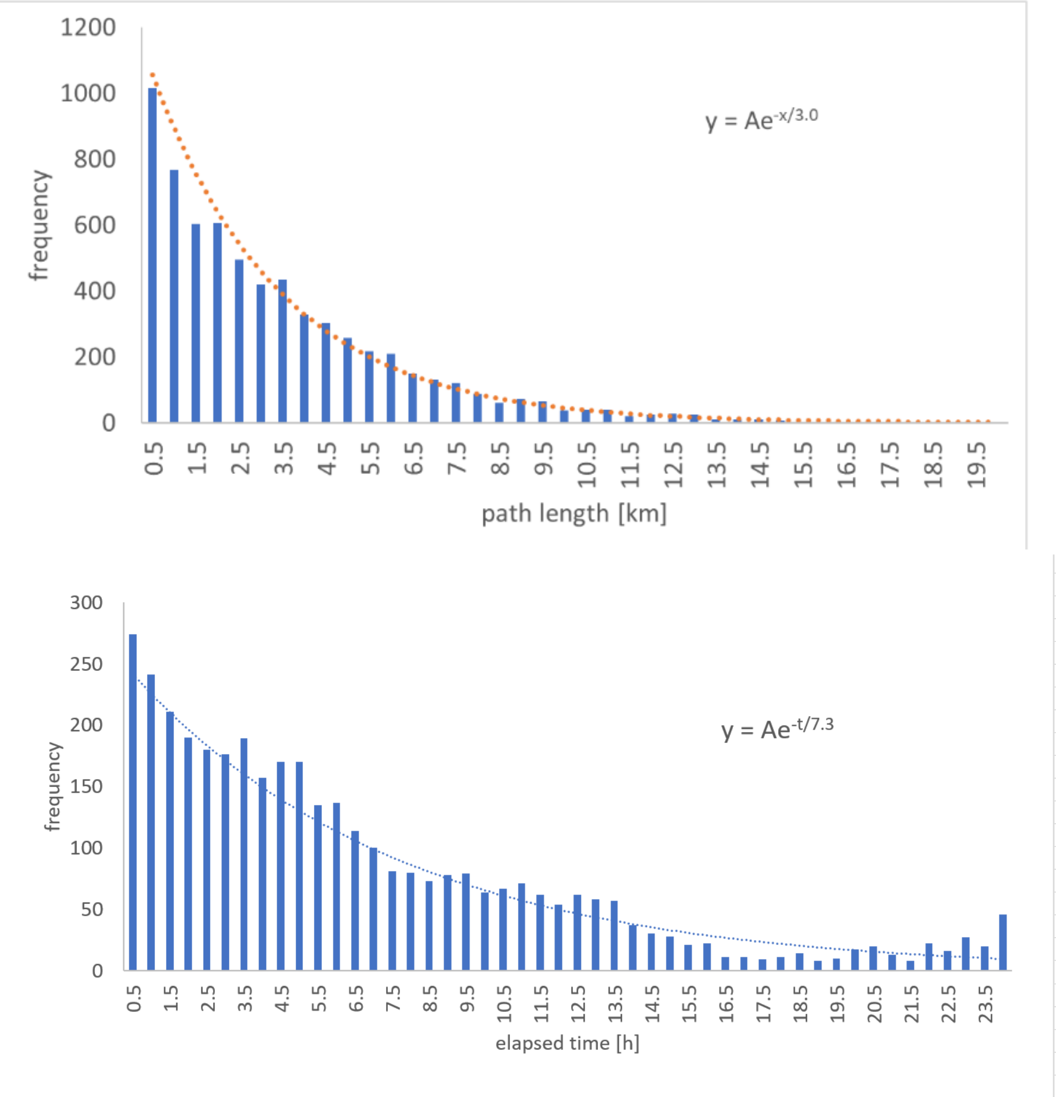


Figure 3: Path length distribution for a sample of visitors in Venice (top) and visit duration distribution (bottom) the dotted curve represents an exponential distribution with a characteristic length of 3 Km and a characteristic duration of 7.3 h (Chiara Mizzi et al. *Unraveling pedestrian mobility on a road network using ICTs data during great tourist events* EPJ Data Science volume 7, Article number: 44 (2018)).

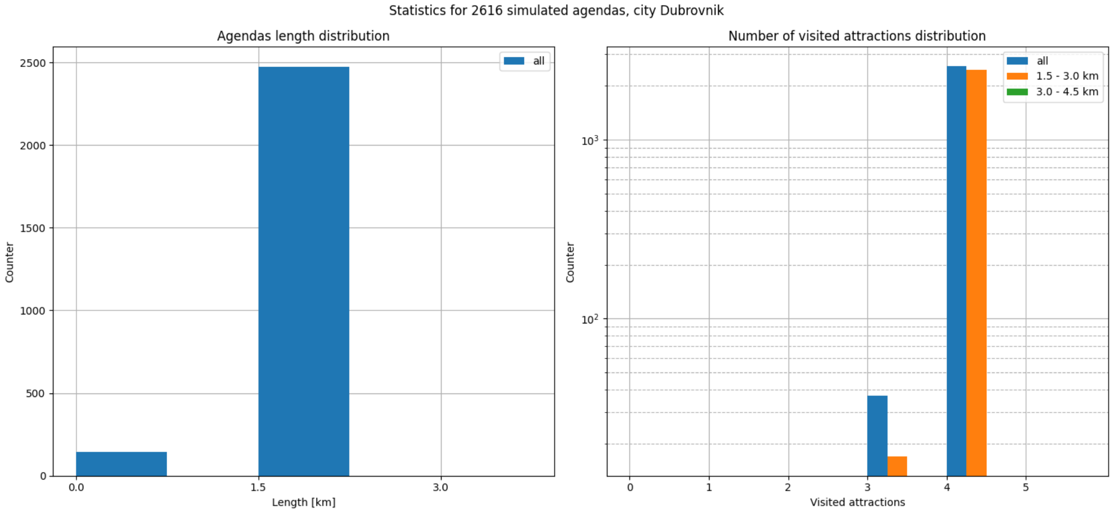


Figure 4: (Left) expected best path length associated to the mobility agenda of the tourist visiting the Venice historical centre and (right) the number of attraction points inserted by the model in the tourist mobility agenda.

2) The city users are introduced in the simulation using specific entry points (sources) for the tourist and the commuters according to the data recorded by the flow sensors of the experimental campaigns, whereas the residents move in the area according to the circadian rhythms (in presence of Hotels in the area a percentage of tourists are supposed to be stay in the area). In Fig. 5 we show an example of entry points and attractions in Venice.

Figure 5: Locations of some source entry points (red circle) and attractions (blue circle) in Venice.

3) Each virtual city user realizes its mobility demand moving on a georeferenced road network (the road networks are derived from Open Street Map (cfr. Figure 1)) according to dynamical rules that simulates the pedestrian dynamics on the road with a typical average velocity of 1 m/sec (the tourist velocity is also reduced due to the frequent expected stops) and the crowding effects that slow down the velocity (we assume the existence of a Fundamental Diagram for pedestrian dynamics).
4) At each crossing point the individuals may choose different directions according to the best paths towards the local destination, the accessibility of the road network (i.e., for the tourists we assume a partial knowledge of the road network so that their mobility is mainly concentrated on a subnetwork related to the most attractive roads due to the presence of shops or to the historical interest) and random effects that simulate the individual free will. The choice algorithm is based on the realization of random events with a probability proportional to a weight associated to the possible choices.
5) The model estimates the required time to visit the main tourist attractions in the area and consider delay effects due to the queues and consider the typical daily mobility path length for a pedestrian that is distribution according to an exponential with an average between 3-4 km, so that the individual mobility agenda could be modified is the elapsed time is much longer than expected visit time or the realized mobility paths exceed the expected length.

The visitor mobility models simulate the dynamics of single individual in the road network considering the crowding effect by means of an empirical fundamental diagram velocity-density: an example is shown in the Figure 6. The pedestrian velocity along a road is reduces linearly as the people density along the road increases. In this way the mobility time depends also on the number of visitors moving in the area and we can also measure the impact of the tourist mobility on the resident mobility. The model has also information of the capacity of attractions so that it introduces a queue effect (the visiting time the attraction increases) when the number of visitors overcomes the attraction capacity.

The interface of the model is shown in Figure 7 where the dots correspond to individual moving on the road network: the model is able to simulate many thousands of individuals in real time.

The output of the simulations is represented by dynamics flow maps or heatmaps that can be visualized in the dashboard of the SLIDES project. An example of these maps is shown in Figure 8 and more details on the dynamics flow maps are given in D 3.2.3.

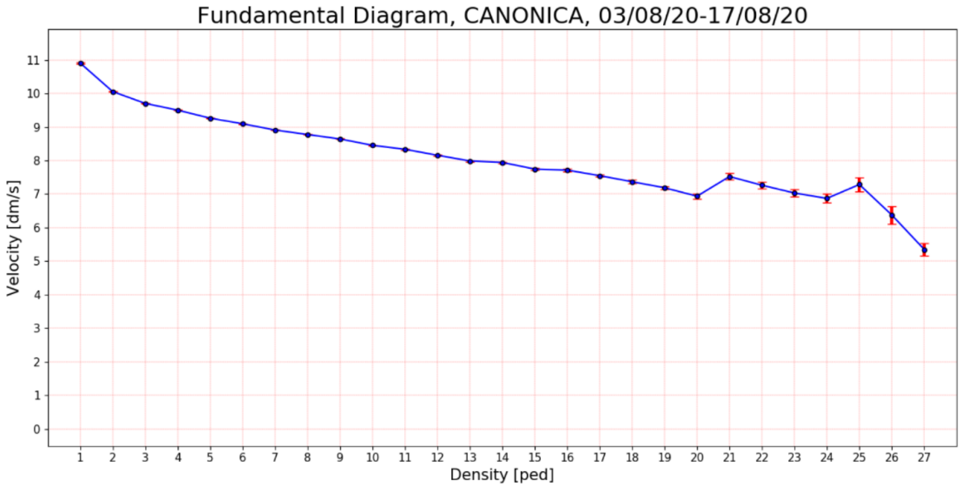


Figure 6: Empirical fundamental diagram velocity-density for pedestrian dynamics in Venice: the average pedestrian velocity (dm/sec) decreases as the density (n. of people in a given area) increases.

Fig. 7: Layout of the visitor mobility model where each dot represents an individual moving in the Venice road network.

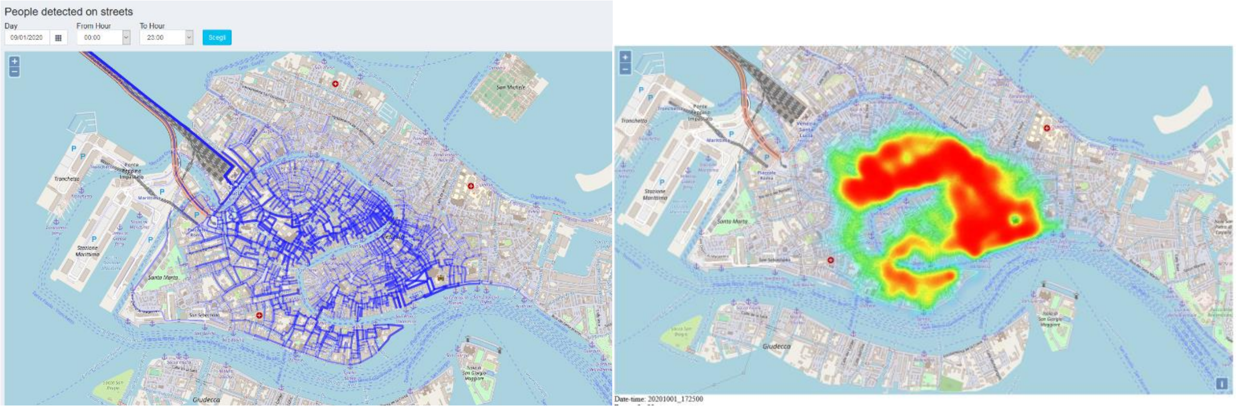


Figure 8: (Left) example of dynamics flow map: the thickness of each line is proportional to the daily tourist flows simulated by the models in the Venice historical centre. (Rigth) example of heatmap simulated by the model: the colour scale gives information on the visitor presences in the Venice historical centre.

2. Short description of the visitor’s mobility models implementation

The Nowcasting models provide simulations for a historical data period extending the information collected from the distributed flow sensors and the statistical data to the whole road network of the considered area. Namely the user can select a time window in the past and the model scrolls the historical databases available for the city of interest and performs a simulation recreating a virtual example of microscopic dynamics which is then reported as a dynamic flow map or density map and the heatmap.

To satisfy the requirements of user-friendliness, portability, scalability and modularity, the models are implemented as a unique RESTful set of APIs which silently differentiate their internal behavior according to the parameters of the user request. The same approach based on simplicity is carried over to the graphical dashboard, which is depicted in the Figure 9 below.

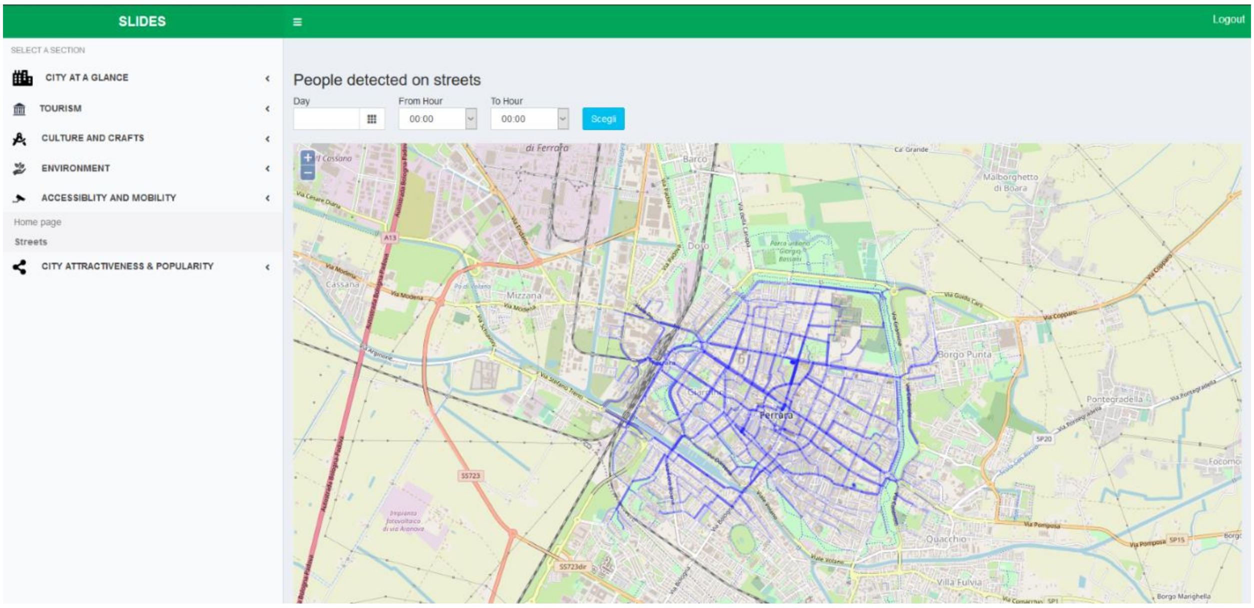


Figure 9: Graphical representation on a SLIDES Dashboard

As mentioned previously, the interface exposes minimal parameters to play with, namely the period and the city of interest. From the technical point of view once the user clicks the start button (“Scegli” in the picture above) a POST request with the given parameters is sent to an API endpoint that performs the simulation and returns the raw data. Then the raw data are rendered in real time as a layer for the OpenStreetMap interactive map.

The Visitor Mobility Models are implemented in a software package whose code is public and available at the internet site: https://github.com/physycom/slides

Some submodules are only available to SLIDES project members since they contain some sensible data recorded during the experimental campaigns.

Here we illustrate the input data required by the models:

1. Cartography data from OpenStreetMap internet site using the *.pbf* format.

The data from public OpenStreeMap source are used to construct a mathematical graph which encodes the road network and is moreover converted to an internal custom format to improve I/O performances. The details about the format are available at the github repository of the project.

2. A list of attractions/points of interest in *kml* format.

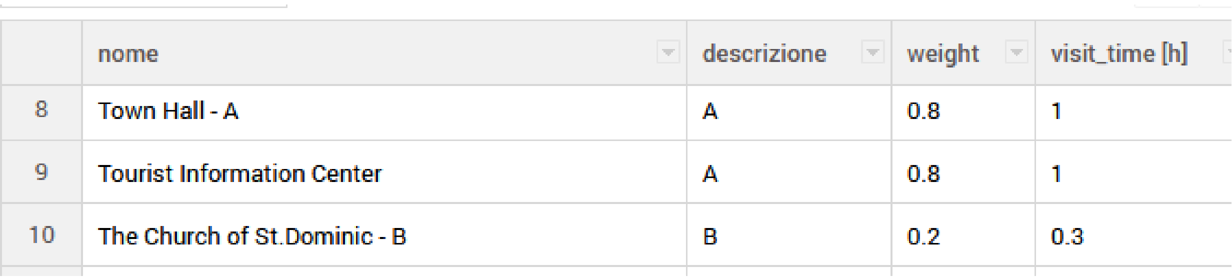
The list of attractions in *km*l format is comprehensive of a metadata table specifying:

   1. Name (field “nome”): the name of the attraction as string.
   2. Type (field “descrizione”): multi-class (A,B,C,D) label to indicate the relevance (rank) of the attraction; this information is used to build the tourist mobility agenda assuming that the probability to choose an attraction is higher if its rank is higher.
   3. Weight (field “weight”): A quantitative implementation of the ratios between Type classes: i.e. it gives a quantitative estimate of different attractivity of attractions when their rank changes.
   4. Visit time (file “visit_time [h]”): The average time spent by visitors in the given attraction; this information is used to generate a distribution of rest times for the visitors according to a Gamma-probability distribution.

LOW LEVEL parameters:

   1. Capacity: is the expected maximum number of visitors at the attractions after which a queue is formed at the entrance. The model uses this information to forecast the presence of queue at the most popular attractions and crowding effects. The model implements an agent-queue mechanism with waiting times for each attraction.
   2. Opening-Closing days: this parameter allows the model to consider seasonal attractions that are open for a specific period of the year.
   3. Opening-Closing daily hours: this information is used to build the visitor mobility agenda since an attraction can be considered by a virtual visitor only if the distance and the visit time is consistent with the opening hours.

All the information is editable by the users and in the Table 1 we give an example of the required data.



	nome	descrizione	weight	visit_time [h]
8	Town Hall - A	A	0.8	1
9	Tourist Information Center	A	0.8	1
10	The Church of St.Dominic - B	B	0.2	0.3

Table 1: An example of the required data

3. A list of sources/entry points of access in *kml* format. The model realizes the visitor mobility using this information to ‘create’ the visitors that enter the considered area and perform the mobility according to their agenda and the resident mobility that is a random mobility in the area with unform distribution of the origin and destinations point. The individuals entering from the entry points are divide between the tourist and the commuters. The last ones move randomly in the area (as the resident) according to the typical working timetable. Both the tourists and the commuters exit from the simulation at the same initial entry point at the end of the visit.
4. Real-time data collected from distributed sensors: wifi/bt sniffers, smart cameras, people counter devices. The visitor mobility models have been designed to integrate the visitor flow data or the presence data recorded by the distributed sensors installed by the experimental campaigns of the project. The real time data are available with a time interval between 1 and 5 minutes and they are used to measure the incoming flows and to perform validation procedures for tuning the model parameters. A careful analysis of the spatial distributions of sensors in relation to the position of the given sources make possible to construct area-specific mapping algorithm to aggregate (or even split) the real data and transform them into timetables of agents’ generation for the various sources of the models.
6. Historical dataset of the distributed sensors. The data collected by the sensors used in the experimental campaigns are stores in a datahub that allow to compute the daily evolution of the visitor flows and the presence in the monitored points and the models used this information to reconstruct the evolution of the daily mobility for the whole road network during the considered day (nowcasting procedure).

7. Statistical information on the characteristics of the considered area (total resident population, expected incoming tourist flows, …). The SLIDES project has collected statistical information of the expected average tourist flows in the cities during the year, the average commuters flows and the tourist offer in the considered area. The model integrates these data in order to simulate the mobility of the different populations.

The Nowcasting visitor mobility models provide mobility simulations using the flow and presence data collected by the experimental campaigns integrated with historical data collected in the SLIDES datahub on tourist flows.

The following block scheme (Figure 10) describes the structure of the code that implements the nowcasting model.

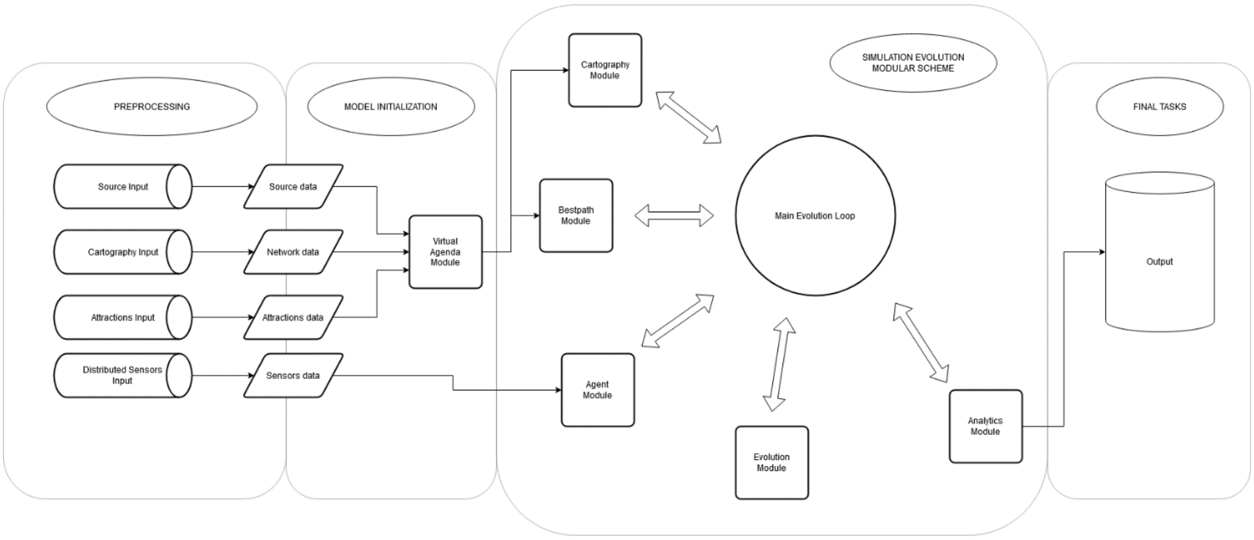


Figure 10: The block scheme of the code that implements the nowcasting model

3. Model Outputs

The visitor’s mobility model provides three type of outputs to illustrate the simulation results:

- Heatmap

  - Geospatial data: grid parameters are editable low-level parameters and are provided as geojson file.
  - Counters timeseries: a custom text format (influxdb ready) which contains the timeseries of model agents' counter.
- Dynamical flow maps
  - Geospatial data: network geometry with metadata but without graph information as a geojson file.
  - Counters timeseries: a collection of timeseries for each polyline of the network containing the counters.
- Timeseries of mobile population counters, on the whole area or portions of it, disaggregated by agents' types as csv text file.

The following figure (Figure 11) shows an example of mobile population counters simulated in the historical centre of Ferrara for the Tourist and the Residents (time step 5 minutes).

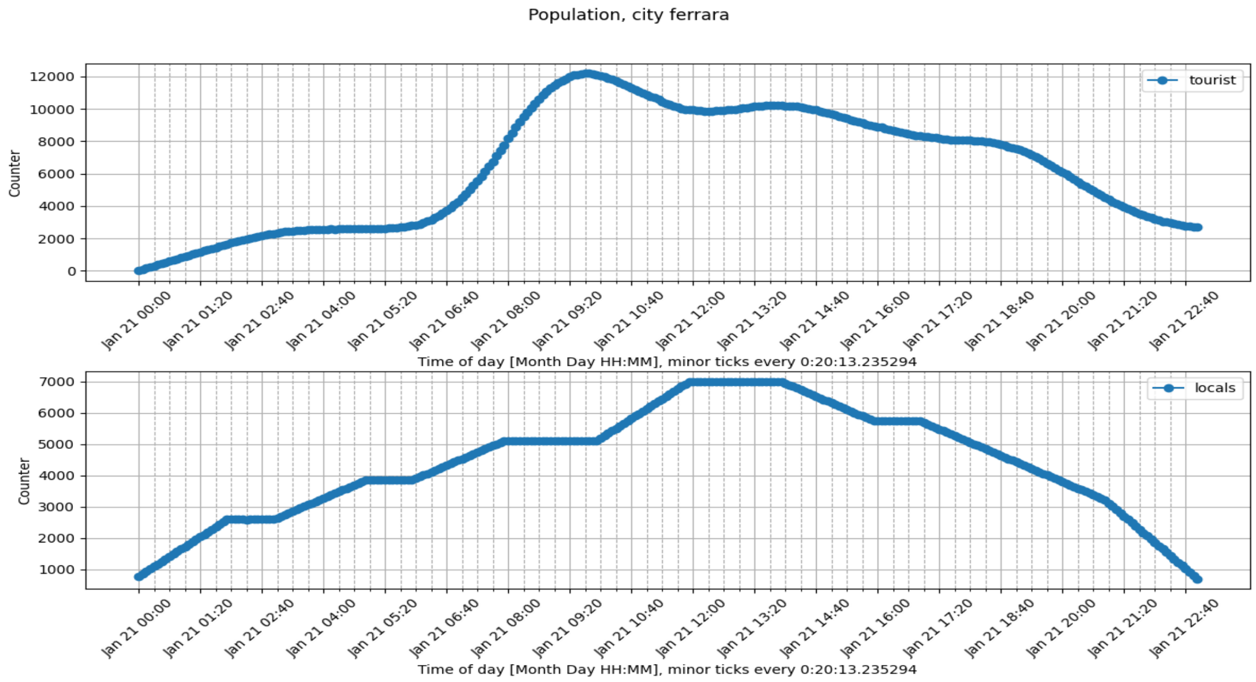


Figure 11: An example of mobile population counters simulated in the historical centre of Ferrara for the Tourist and the Residents

4. Model webservice

We have provided to the partners of the SLIDES project the possibility of using the mobility model through a webservice system. The webservice endpoint will be provided once the project dedicated server machine will be fully operative and is indicated in what follows as *apiurl*. To ensure portability the webservice is packed as a single Linux-based docker container. The webservice security is ensured through a fully featured OAuth2 authorization mechanism with permission-per-user based policies. The webservice provides a remote interface for using the model to run simulations. It constitutes of several APIs:

- apiurl/welcome : webservice version GET request.
- apiurl/login : access token POST request.
- apiurl/sim : simulation POST request.
- apiurl/poly : network geospatial data GET request.
- apiurl/grid : heatmap grid geospatial data GET request.

All the technical details related to requests will be available, in a fully openapi compliant way, at

- apiurl/redoc
- apiurl/docs

In the Figure 12 we show the OpenAPI fully-compliant interactive documentation page screenshot of the webservice.

SLIDES – Simulation webservice API doc

Sim Walker Post

QUERY PARAMETERS

string (Citytag)
Default: "null"

REQUEST BODY SCHEMA: application/json

Responses

> 200 Successful Response

> 422 Validation Error

Request samples

Content type
application/json

Response samples

Content type
application/json

null

Figure 12: The OpenAPI fully compliant interactive documentation page screenshot of the webservice