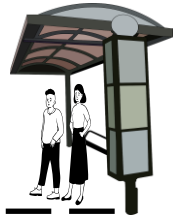
UNIVERSITÀ DEGLI STUDI FIRENZE
DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE
DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB DISTRIBUTED DATA INTELLIGENCE AND TECHNOLOGIES LAB

BIG DATA Architecture
prof. P. Nesi
Public Transport Demand and Offer Match Analysis using SUMO

Ammarah Irum
PhD Information Engineering



Outline

- Introduction
- Public Transport Demand and Offer
- Goal
- Key Tools and Scripts
- Data Representation in SUMO
- General Workflow
- Detailed Workflow
- Public Transport Demand and Offer Data Generation Methods
- Running Simulation
- Results
- Probelm & Solutions
- Reinforcement Learning
- Summary

Introduction

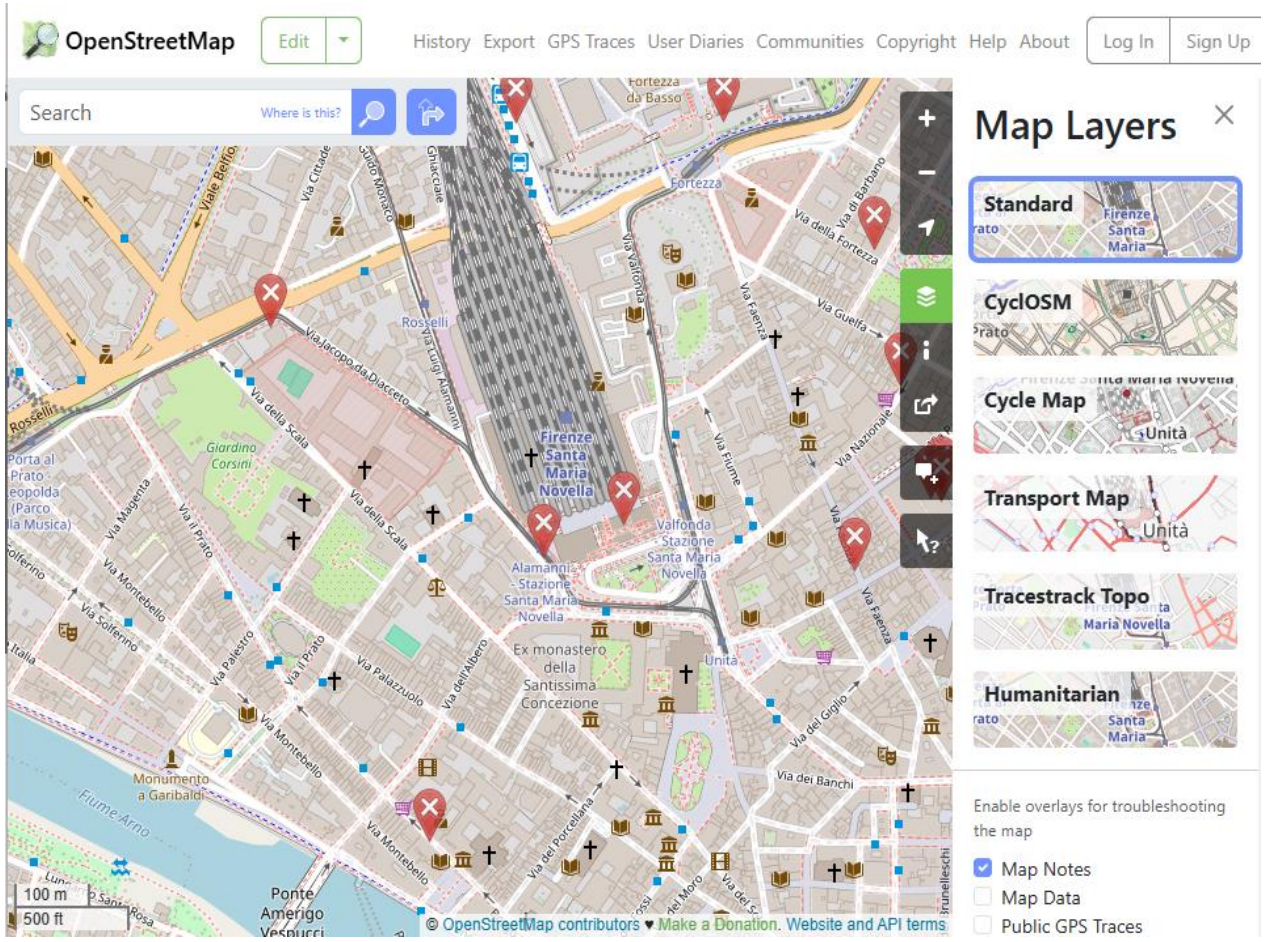
- SUMO:
  - "**S**imulation of **U**rban **MO**bility"
  - Open source
  - Highly portable
  - Microscopic
  - High interoperability through usage of XML-data only
  - Complete traffic simulation package designed to handle large networks.
  - Intermodal simulation including pedestrians and various modes of vehicles
  - It comes with a large set of tools for scenario creation.
  - It is mainly developed by employees of the institute of transportation systems at the German aerospace center.

Introduction

- **OSM:**
  - Road Infrastructure (road links, nodes, lanes and

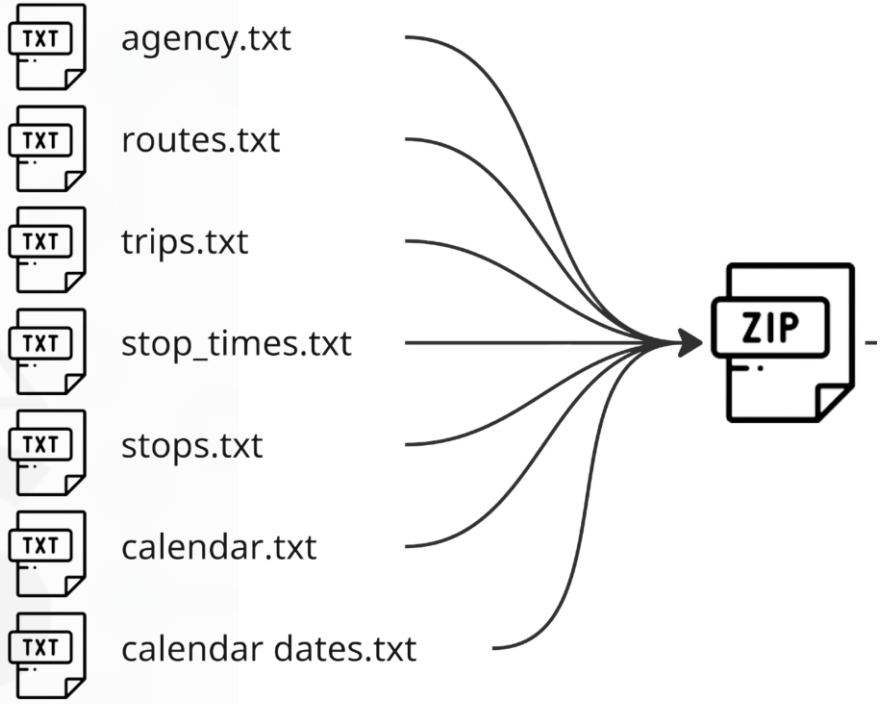
    associated attributes)
  - POIs (buildings, various facilities etc.)
  - Land use (type, location, shape, size)
  - Public transport timetables and stops information
  - **Pros**: Heterogeneity, responsiveness, flexibility,

    volunteered geographic information
  - **Cons**: Crowdsourced, not up-to-date, lack of reliability



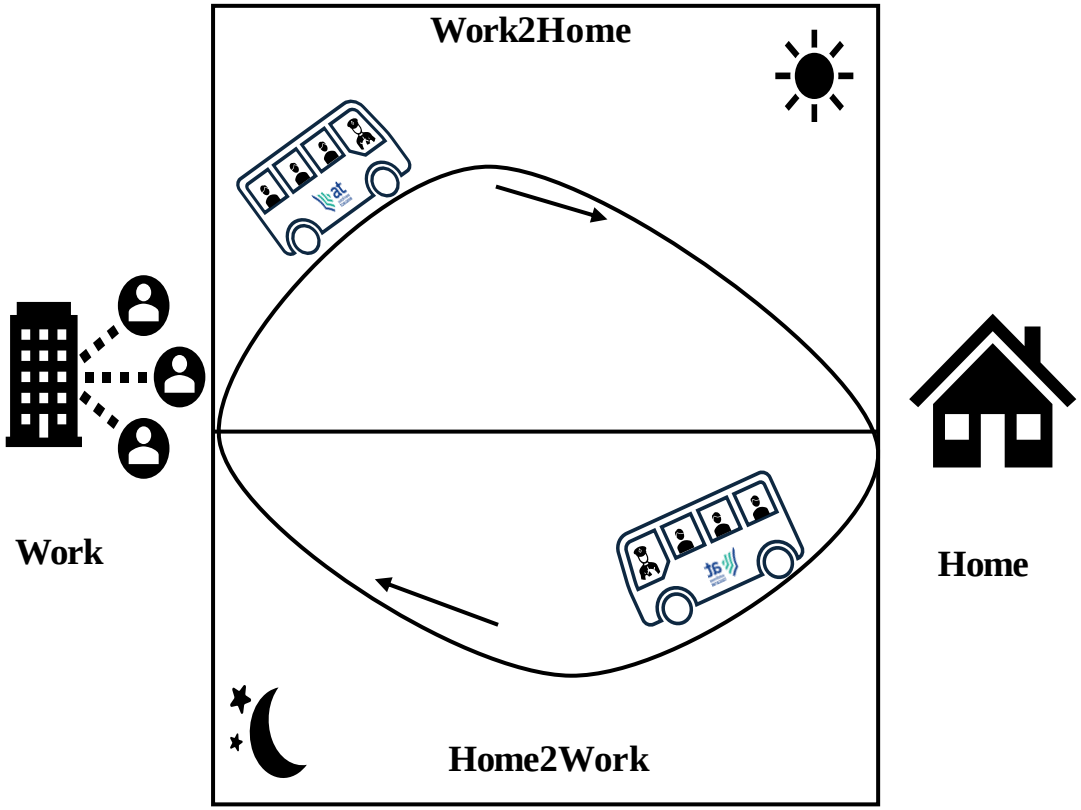
Introduction

- **GTFS:**
  - Network and service information of a transport service provider
  - Routes, stops and timetables along with associated geographic information
  - Openly available
  - Industry standard
  - Precise data
  - Versions (Standard and Real-time)



Public Transport Demand and Offer

- Demand:
  - Mobility demand by persons in a dedicated area
  - Persons trips from work to home and vice versa
- Offer:
  - Public transport service providers
  - Maintaining low cost yet seamless offer
- Public transport demand should be satisfied by the offer made by service providers.



Goal

"To deploy various scenarios in SUMO for creating a balanced tradeoff between public transport demand and offer in such a way that real world transport schedules (GTFS) are integrated with real-world mobility demand (OD matrix based) to cater the needs of people using public transport by considering minimal trip cost to public transport service providers"

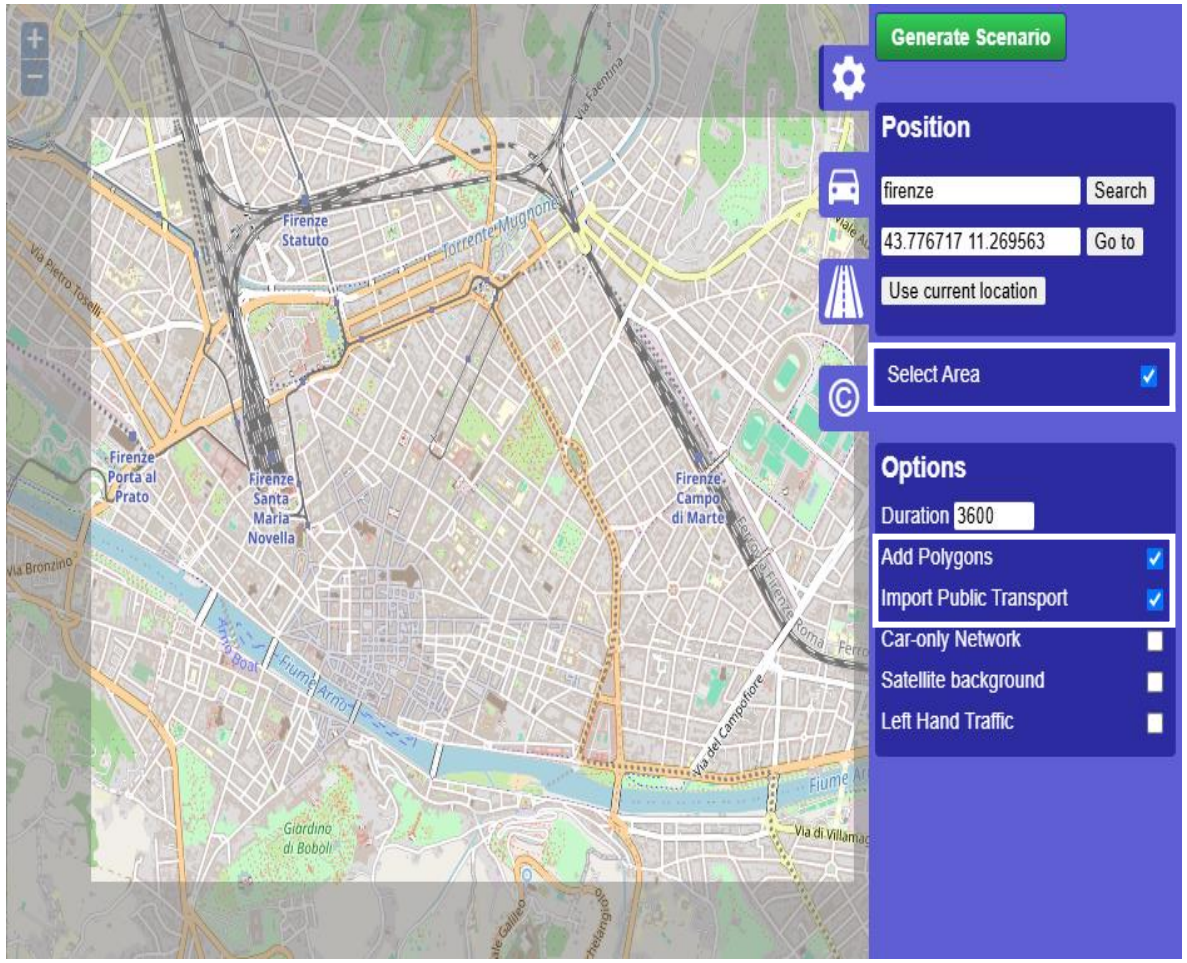
Key Tools & Scripts

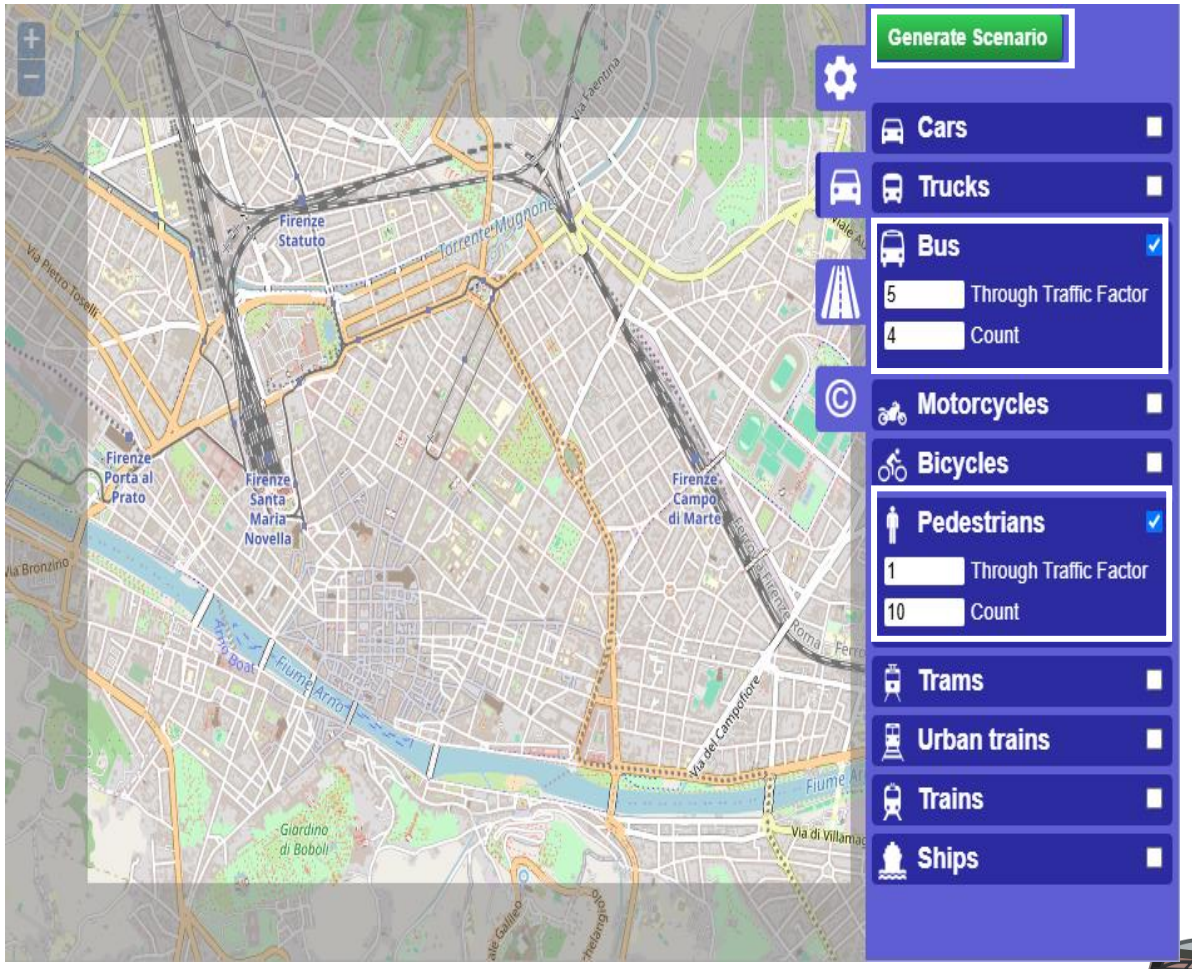
- **OSM Web Wizard:**
  - Scenario creation toolkit
- **gtfs2pt.py**
  - main script to import public transport from GTFS
  - python gtfs\gtfs2pt.py –n myNet.net.xml --gtfs Florence-GTFS.zip --date 20241023 --repair --modes bus --vtype-output pt_vtypes.xml
- **randomtrips.py**
  - generates a set of random trips for a given network
  - python tools/**randomTrips.py** -n net.net.xml -o pass.trips.xml -r persontrips.xml -b 0 -e 3600

Key Tools & Scripts

- **od2trips**
  - imports O/D-matrices and splits them into single vehicle trips.
  - **od2trips** -n districts.taz.xml -d routes2od_file.od.xml -z routes2od_file.add.xml -o passenger_trips.xml --persontrips --persontrips.modes --begin 0 --end 3600 --spread.uniform --vtype bus
- **ActivityGen**
  - imports O/D-matrices and splits them into single vehicle trips.
  - **activitygen** -n myNet.net.xml --stat-file activitygen.stat.xml --output-file trips.xml --seed 1

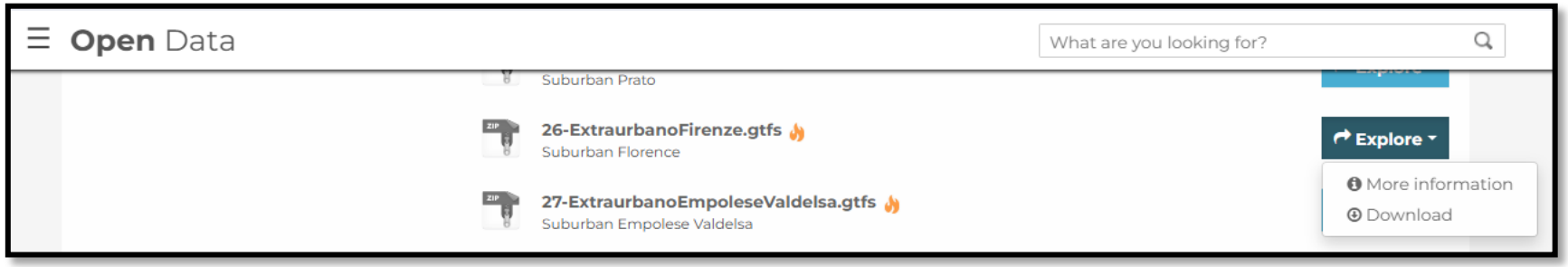
OSM Web Wizard



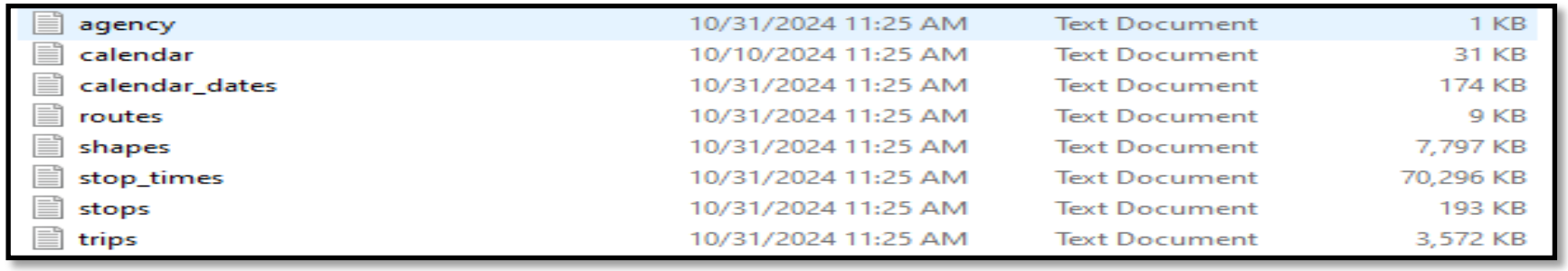


gtfs2pt.py

1. Download the GTFS file of the concerned region from dedicated website.



2. https://dati.toscana.it/dataset/rt-oraritb



3. python gtfs\gtfs2pt.py –n myNet.net.xml --gtfs Florence-GTFS.zip --date 20241204 --repair --modes bus --vtype-output pt_vtypes.xml

gtfs2pt.py

4. Output:

1. gtfs_pt_stops.add.xml

```
<busStop id="5487_70030269.0" lane="704745532_0" startPos="0.00" endPos="20.00" friendlyPos="true" name="Borghini"/>
<busStop id="5487_70030269.1" lane="598396138_0" startPos="0.00" endPos="20.00" friendlyPos="true" name="Mille Deposito At"/>
<busStop id="5487_70030269.2" lane="24860987#1_0" startPos="22.62" endPos="42.62" friendlyPos="true" name="Mille Marconi"/>
```

2. gtfs_pt_vehicles.add.xml

```
<vehicle id="5487_70305001.0" route="5487_70300547" type="bus" depart="17570" line="5487_70300547">
    <param key="gtfs.route_name" value="30"/>
    <param key="gtfs.trip_headsign" value="Leopolda Porta Al Prato"/>
</vehicle>
```

3. pt_vtypes.xml

```
<additional xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://sumo.dlr.de/xsd/additional_file.xsd">
   <vType id="bus" vClass="bus"/>
</additional>
```

randomtrips.py

1. Python script already present in Sumo home folder.

2. python tools/**randomTrips.py** -n net.net.xml -o veh.trips.xml -r vehroutes.xml -b 0 -e 3600

```
<trip id="0_bus_80:0" type="pt_bus" depart="3.0" departLane="best" from="-24854471#6" to="609700904">
    <stop busStop="-5820311128" duration="30"/>
    <stop busStop="-5770323939" duration="30"/>
    <stop busStop="4548975676" duration="30"/>
</trip>
<trip id="1_bus_80:1" type="pt_bus" depart="15.0" departLane="best" from="452879254#0" to="24854471#6">
    <stop busStop="3577393533" duration="30"/>
    <stop busStop="5749924522" duration="30"/>
    <stop busStop="5770323938" duration="30"/>
    <stop busStop="5770323935" duration="30"/>
</trip>
```

od2trips

- od2trips imports O/D-matrices and splits them into single vehicle trips.
- Od2trips -v --taz-files TAZ.xml --vtype bus --prefix pedestrian --od-matrix-files od.mtx --persontrips true --persontrips.modes "public" -o output.odtrips.xml
- **Purpose:** Conversion of O/D-matrices to single vehicle trips

```
$O;D3
* da A
7.00 9.00
* Fatt
1.00
*
* TPS Transport Planning Service S.r.l. Perugia PG
* 05.02.21
          1           1 44.386
          1           2  3.167
          1           3  0.770
          1           4  6.382
          1           5  1.166
          1           6  3.153
          1           7 70.112
          1           8  4.337
          1           9  6.173
          1          10  2.838
          1          11 11.189
          1          12  0.663
          1          13  0.515
```

OD Matrix

od2trips

- **Input (mandatory):**
  - O/D-Matrix
  - a set of districts

```
<tazs xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http
    <taz id="1" color="51,128,255" edges="-1033426333 -1033426334 -1087069687#0 -1087069687#1 -
    <taz id="10" color="51,128,255" edges="-113005153 -113005158#0 -113005158#1 -27370886#0 -27
    <taz id="11" color="51,128,255" edges="-393390393#0 -393390393#1 -478264202#0 -478264202#1
    <taz id="118" color="51,128,255" edges="-1108592425 -181315437#0 -181315437#1 -181315437#2
    <taz id="119" color="51,128,255" edges="-1074107232 -1079983579#0 -1079983579#1 -1079983579
```

Set of Districts

- **Output:** A list of person trip definitions

```
<person id="pedestrian269" depart="25202.83" departPos="random">
    <personTrip from="689946172" to="246189772" fromTaz="1" toTaz="7" arrivalPos="random" modes="public"/>
</person>
```

Person Trips

activitygen

- Produces trip-based mobility meant to represent the concatenation of trips required to satisfy the sequence of activities in a personal plan.
- Based on the population statistics of an area.
- activitygen -n myNet.net.xml --stat-file activitygen.stat.xml --output-file trips.xml --seed 1 --random

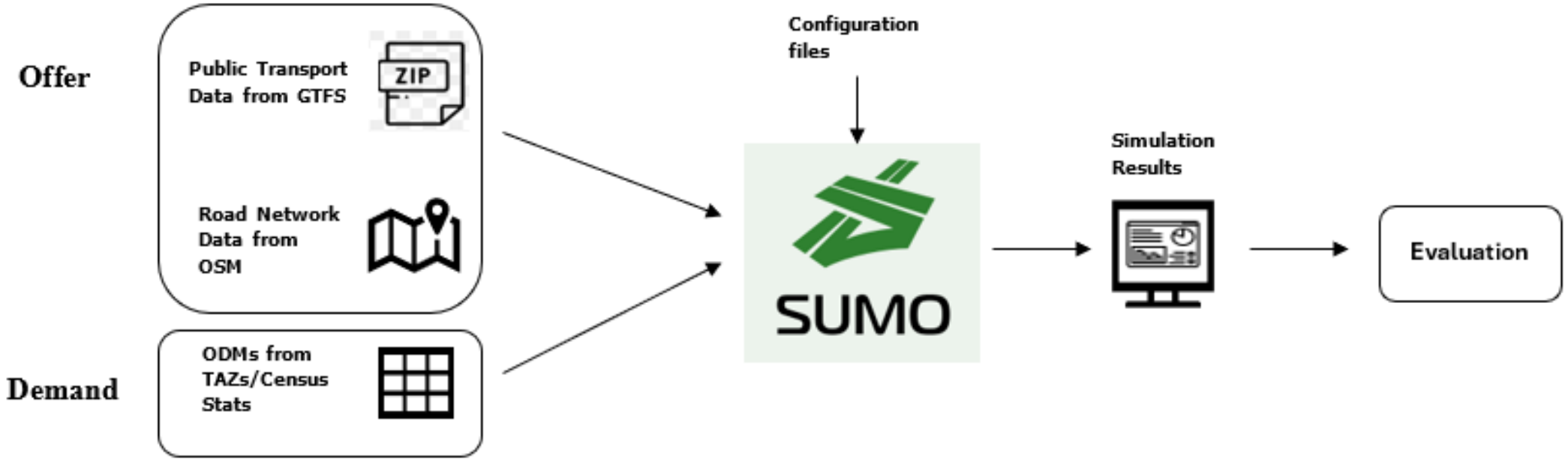
```
<city>

    <!-- Define Population Size -->
    <general inhabitants="8" households="5" childrenAgeLimit="19" retirementAgeLimit="66" carRate="0" unemploymentRate="0.05" footDistanceLimit="250" incomingTraffic="1" outgoingTraffic="1"/>
    <parameters carPreference="0" meanTimePerKmInCity="6" freeTimeActivityRate="0.15" uniformRandomTraffic="0.20" departureVariation="300"/> <!-- Simulate trips for 1000 people -->

    <population>
        <bracket beginAge="0" endAge="30" peopleNbr="2" />
        <bracket beginAge="30" endAge="60" peopleNbr="4" />
        <bracket beginAge="60" endAge="90" peopleNbr="2" />
    </population>

    <workHours>
        <opening hour="106" proportion="0.5" />
        <closing hour="2755" proportion="0.5" />
    </workHours>

    <cityGates>
        <entrance edge="173036829#0" pos="3.35" incoming="1" outgoing="1" />
        <entrance edge="1135788009#1" pos="181.24" incoming="1" outgoing="1" />
    </cityGates>
    <schools>
        <school edge="-31727571#1" pos="19.93" beginAge="20" endAge="70" capacity="50" opening="200" closing="2500" />
    </schools>
```

Stats File

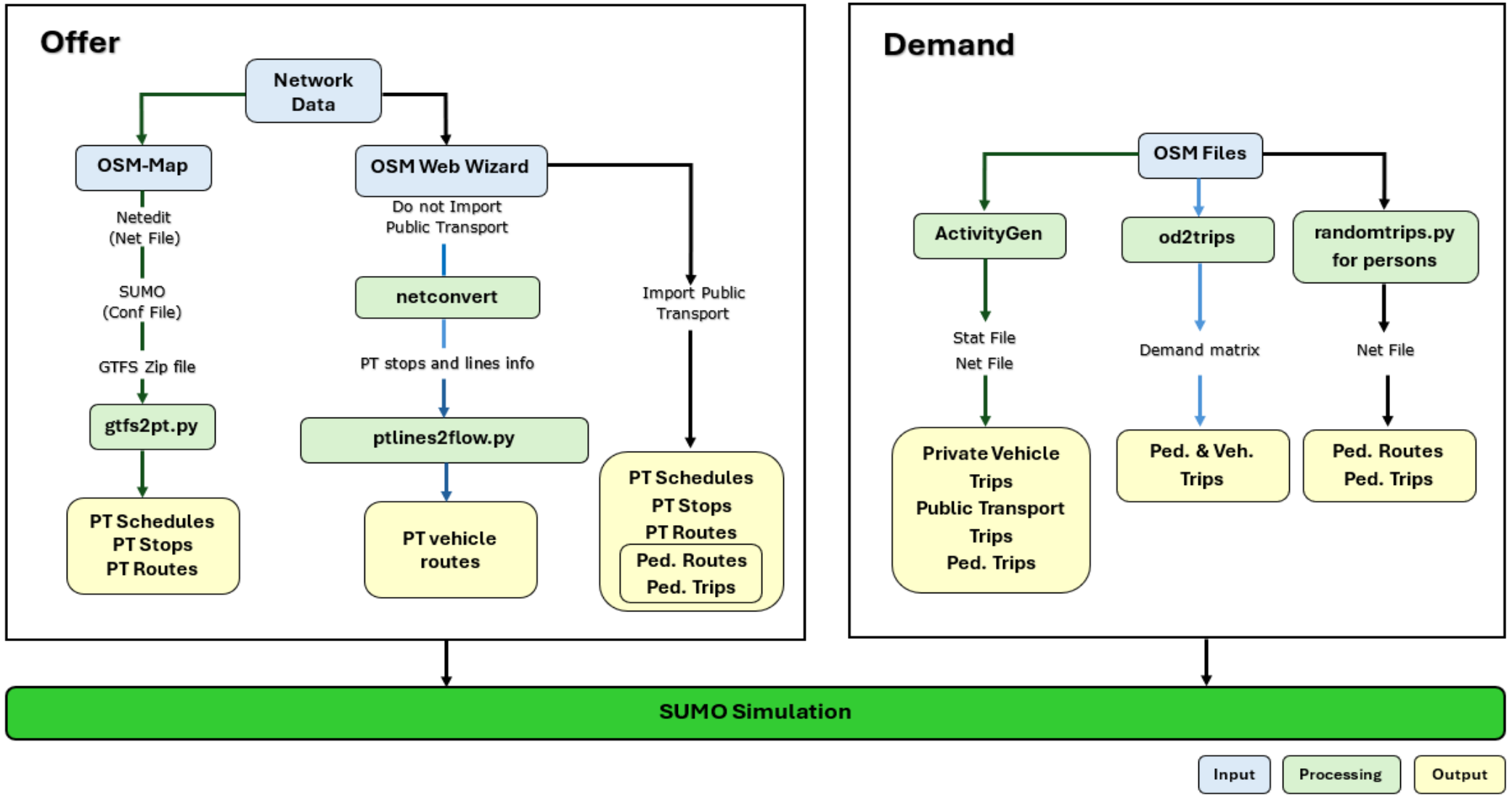
Data Representation in SUMO

- **Methods to extract road network:**
  1. OSM Web Wizard
  2. OSM map
- **Methods to extract public transit network:**
  1. OSM Web Wizard ("import public transport" option selected)
  2. GTFS (gtfs2pt.py tool)
  3. ptlines2flow.py tool followed by netconvert
  4. randomtrips.py tool (with "--vehicles" option)
- **Methods to extract person trips:**
  1. OSM Web Wizard ("Pedestrian" option selected)
  2. randomtrips.py tool (with "--persontrips" option)
  3. Od2trips tool
  4. ActivityGen tool

General WorkFlow

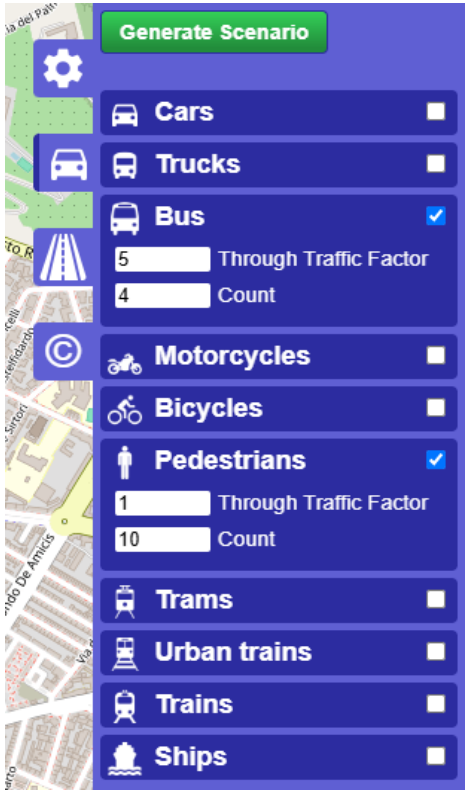


Detailed WorkFlow



Public Transport Offer

1. **OSM Web Wizard → Import Public Transport → <u>Simulation 1</u>**
   - Input:
     - OSM Web Wizard
   - Processing:
     - Select options and click generate scenario
   - Output:
     - Net file, SUMO conf. file, PT routes and stops file from OSM.
     - Route file for persons from OSM.
   - Simulation Results:
     - End Time: 7003 sec
     - Bus Stops in simulation area: 88
     - PT lines in simulation area: 55
     - Vehicles: 522
     - Pedestrians: 1460
       - Ride-takers: 19



Note: No commands have been used.

Simulation 1 Analysis

sumo *-c osm.sumocfg* --statistic-output stats.xml --summary summary.xml --human-readable-time true

```
Simulation ended at time: 7003.00
Reason: All vehicles have left the simulation.
Performance:
 Duration: 7.46s
 Real time factor: 938.866
 UPS: 45203.512535
 UPS-Persons: 167122.402467
Vehicles:
 Inserted: 522
 Running: 0
 Waiting: 0
 Teleports: 38 (Collisions: 1, Wrong Lane: 38)
 Emergency Braking: 12
Persons:
 Inserted: 1396
 Running: 0
 Jammed: 232
Statistics (avg of 522):
 RouteLength: 1942.94
 Speed: 3.36
 Duration: 645.92
 WaitingTime: 135.71
 TimeLoss: 433.26
 DepartDelay: 18.60
Pedestrian Statistics (avg of 1467 walks):
 RouteLength: 1013.11
 Duration: 845.23
 TimeLoss: 111.07
Ride Statistics (avg of 19 rides):
 WaitingTime: 129.63
 RouteLength: 408.83
 Duration: 78.05
 Bus: 19

DijkstraRouter answered 57953 queries and explored 9.47 edges on average.
DijkstraRouter spent 0.11s answering queries (0.00ms on average).
DijkstraRouter answered 192 queries and explored 598.81 edges on average.
DijkstraRouter spent 0.06s answering queries (0.31ms on average).
```

Public Transport Offer

2. OSM Web Wizard → Do not Import Public Transport → ptlines2flow.py script → Simulation 2

- Input: OSM Web Wizard
- Processing:

Offer

- **netconvert** --osm-files osm.xml **-o net.net.xml** --osm.stop-output.length **20** **--ptstop-output additional.xml** **--ptline-output ptlines.xml**
- python tools/**ptlines2flows.py** -n net.net.xml -s additional.xml -l ptlines.xml **-o pt_routes.rou.xml** -p 600 --use-osm-routes

Demand

- python tools/**randomTrips.py** -n net.net.xml -o pass.trips.xml -r persontrips.xml -b 0 -e 3600 **--personrides true**

- **Simulation Results:**
  - End Time: 4023 sec
  - Bus Stops in simulation area: 90
  - PT lines in simulation area: 57
  - Vehicles: 522
  - Pedestrians: 1460
    - Ride-takers: none

Simulation 2 Analysis

sumo *-c osm.sumocfg* --statistic-output stats.xml --summary summary.xml --human-readable-time true

```
Simulation ended at time: 4023.00
Reason: All vehicles have left the simulation.
Performance:
 Duration: 1.85s
 Real time factor: 2178.13
 UPS: 14940.985382
 UPS-Persons: 183723.876557
Vehicles:
 Inserted: 72
 Running: 0
 Waiting: 0
 Teleports: 33 (Jam: 3, Wrong Lane: 30)
 Emergency Braking: 1
Persons:
 Inserted: 346
 Running: 0
 Jammed: 169
Statistics (avg of 72):
 RouteLength: 1230.73
 Speed: 3.93
 Duration: 383.28
 WaitingTime: 193.69
 TimeLoss: 239.28
 DepartDelay: 0.07
Pedestrian Statistics (avg of 347 walks):
 RouteLength: 879.69
 Duration: 977.83
 TimeLoss: 344.45
```

Public Transport Offer

4. OSM Map → GTFS File → SUMO Configuration File → gtfs2pt.py script → Simulation 3

- Input: OSM Map
- Processing:
  - Use Netedit to generate Net and SUMO configuration file.
  - Use gtfs2pt.py to generate PT routes and stops file

Offer

- Download gtfs zip file.
- python gtfs\gtfs2pt.py –n myNet.net.xml --gtfs Florence-GTFS.zip --date 20241023 --repair --modes bus --vtype-output pt_vtypes.xml
  - GTFS routes and stops file is generated

Demand

- Demand can be generated by various methods:
  - **Method 1:**
    - **od2trips** -n districts.taz.xml -d routes2od_file.od.xml -z routes2od_file.add.xml -o passenger_trips.xml --persontrips --persontrips.modes --begin 0 --end 3600 --spread.uniform --vtype bus
  - **Method 2 (Activity based demand generation):**
    - **activitygen** -n myNet.net.xml --stat-file activitygen.stat.xml --output-file trips.xml --seed 1

- Output: Net file, SUMO conf. file, PT routes and stops file, passenger trips.

Simulation 3 Analysis (Method 1)

sumo *-c osm.sumocfg* --statistic-output stats.xml --summary summary.xml --human-readable-time true

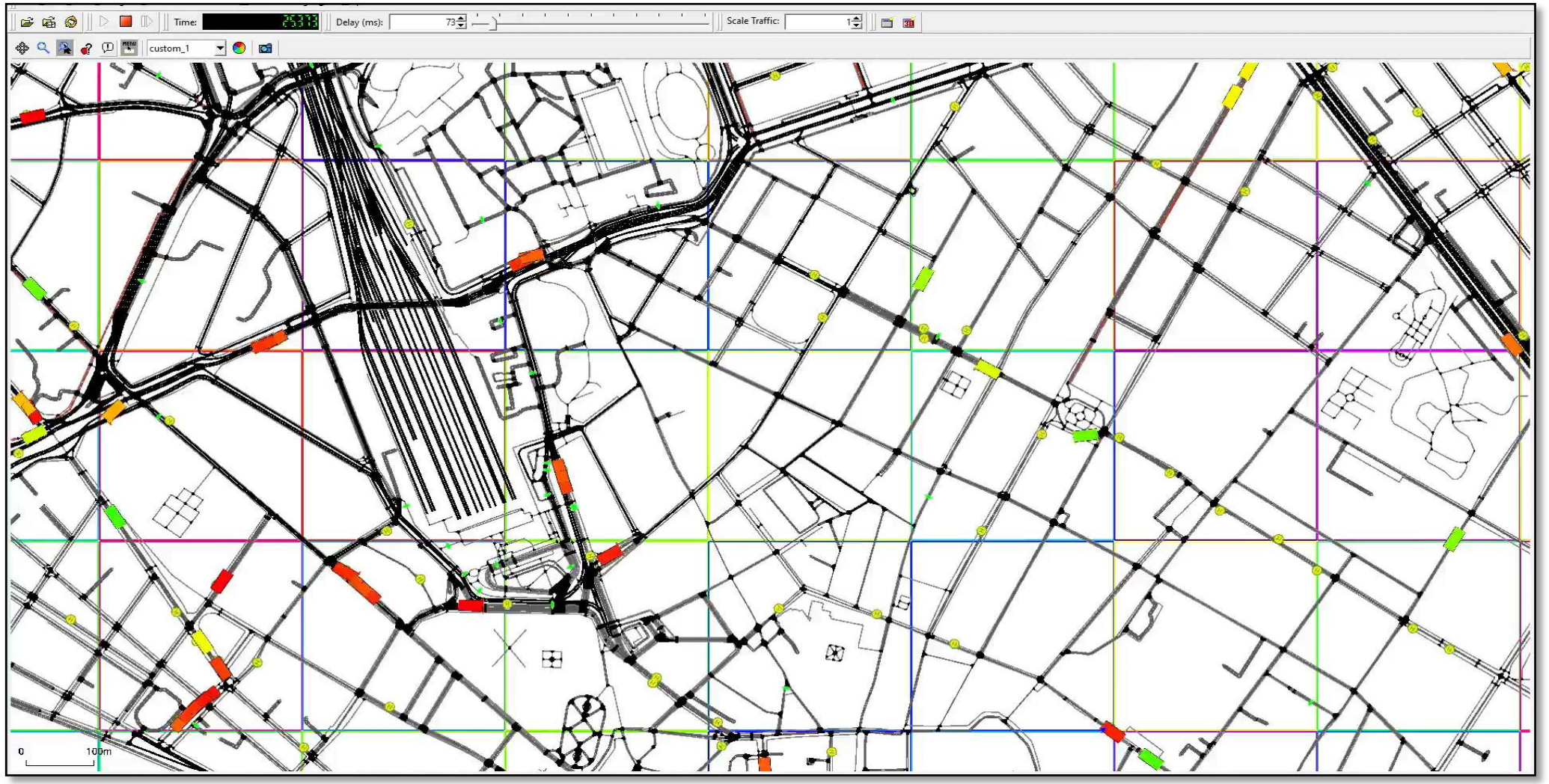
```
Simulation ended at time: 86944.00
Reason: All vehicles have left the simulation.
Performance:
 Duration: 10.13s
 Real time factor: 8585.37
 UPS: 70750.370297
 UPS-Persons: 1258.121852
Vehicles:
 Inserted: 3503
 Running: 0
 Waiting: 0
Persons:
 Inserted: 9
 Running: 0
 Jammed: 6
Statistics (avg of 3503):
 RouteLength: 1991.04
 Speed: 10.04
 Duration: 204.53
 WaitingTime: 19.54
 TimeLoss: 77.79
 DepartDelay: 0.84
Pedestrian Statistics (avg of 9 walks):
 RouteLength: 1283.94
 Duration: 1415.67
 TimeLoss: 504.33
```

Simulation 3 Analysis (Method 2)

sumo *-c osm.sumocfg* --statistic-output stats.xml --summary summary.xml --human-readable-time true

```
Simulation version 1.19.0 started with time: 0.00.
Warning: No connection between edge '1135788009#1' and edge '1135788009#1' found.
Warning: Vehicle 'bl1b1:1' has no valid route from edge '1135788009#1' to stop edge '1135788009#1'.
Warning: No connection between edge '1135788009#1' and edge '-23348473#4' found.
Warning: Vehicle 'bl1b1:1' has no valid route from edge '1135788009#1' to stop edge '-23348473#4'.
Warning: No connection between edge '1135788009#1' and edge '-350214396#0' found.
Warning: No route for vehicle 'carIn1:1' found.
Warning: No connection between edge '1135788009#1' and edge '1135788009#1' found.
Warning: Vehicle 'bl1b3:1' has no valid route from edge '1135788009#1' to stop edge '1135788009#1'.
Warning: No connection between edge '1135788009#1' and edge '-23348473#4' found.
Warning: Vehicle 'bl1b3:1' has no valid route from edge '1135788009#1' to stop edge '-23348473#4'.
Warning: Teleporting vehicle 'carIn1:1'; waited too long (wrong lane), lane='1135788009#1_0', time=539.00.
Warning: Vehicle 'carIn1:1' ends teleporting on edge '-350214396#0', time=539.00.
Warning: No connection between edge '1135788009#1' and edge '1135788009#1' found.
Warning: Vehicle 'bl1b5:1' has no valid route from edge '1135788009#1' to stop edge '1135788009#1'.
Warning: No connection between edge '1135788009#1' and edge '-23348473#4' found.
Warning: Vehicle 'bl1b5:1' has no valid route from edge '1135788009#1' to stop edge '-23348473#4'.
Warning: No connection between edge '1135788009#1' and edge '1135788009#1' found.
Warning: Vehicle 'bl1b7:1' has no valid route from edge '1135788009#1' to stop edge '1135788009#1'.
Warning: No connection between edge '1135788009#1' and edge '-23348473#4' found.
Warning: Vehicle 'bl1b7:1' has no valid route from edge '1135788009#1' to stop edge '-23348473#4'.
Warning: Teleporting vehicle 'bl1b1:1'; waited too long (wrong lane), lane='1135788009#1_0', time=881.00.
Warning: Vehicle 'bl1b1:1' ends teleporting on edge '1135788009#1', time=881.00.
Warning: Teleporting vehicle 'bl1b3:1'; waited too long (wrong lane), lane='1135788009#1_0', time=1189.00.
Warning: Vehicle 'bl1b3:1' ends teleporting on edge '1135788009#1', time=1189.00.
Warning: Teleporting vehicle 'bl1b1:1'; waited too long (wrong lane), lane='1135788009#1_0', time=1496.00.
Warning: Vehicle 'bl1b1:1' ends teleporting on edge '-23348473#4', time=1496.00.
Warning: Teleporting vehicle 'bl1b5:1'; waited too long (wrong lane), lane='1135788009#1_0', time=1821.00.
Warning: Vehicle 'bl1b5:1' ends teleporting on edge '1135788009#1', time=1821.00.
Warning: Teleporting vehicle 'bl1b7:1'; waited too long (wrong lane), lane='1135788009#1_0', time=2129.00.
Warning: Vehicle 'bl1b7:1' ends teleporting on edge '1135788009#1', time=2129.00.
Warning: Teleporting vehicle 'bl1b3:1'; waited too long (wrong lane), lane='1135788009#1_0', time=2436.00.
Warning: Vehicle 'bl1b3:1' ends teleporting on edge '-23348473#4', time=2436.00.
Warning: Teleporting vehicle 'bl1b5:1'; waited too long (wrong lane), lane='1135788009#1_0', time=2744.00.
Warning: Vehicle 'bl1b5:1' ends teleporting on edge '-23348473#4', time=2744.00.
Warning: Teleporting vehicle 'bl1b7:1'; waited too long (wrong lane), lane='1135788009#1_0', time=3052.00.
Warning: Vehicle 'bl1b7:1' ends teleporting on edge '-23348473#4', time=3052.00.
Warning: No connection between edge '1135788009#1' and edge '-350214396#0' found.
Warning: No route for vehicle 'carIn1:2' found.
Warning: Teleporting vehicle 'carIn1:2'; waited too long (wrong lane), lane='1135788009#1_0', time=86639.00.
Warning: Vehicle 'carIn1:2' ends teleporting on edge '-350214396#0', time=86639.00.
Simulation ended at time: 86646.00
Reason: All vehicles have left the simulation.
Statistics (avg of 10):
 RouteLength: 4382.71
 Speed: 4.96
 Duration: 1194.70
 WaitingTime: 744.20
 TimeLoss: 872.97
 DepartDelay: 0.00
```

Simulation

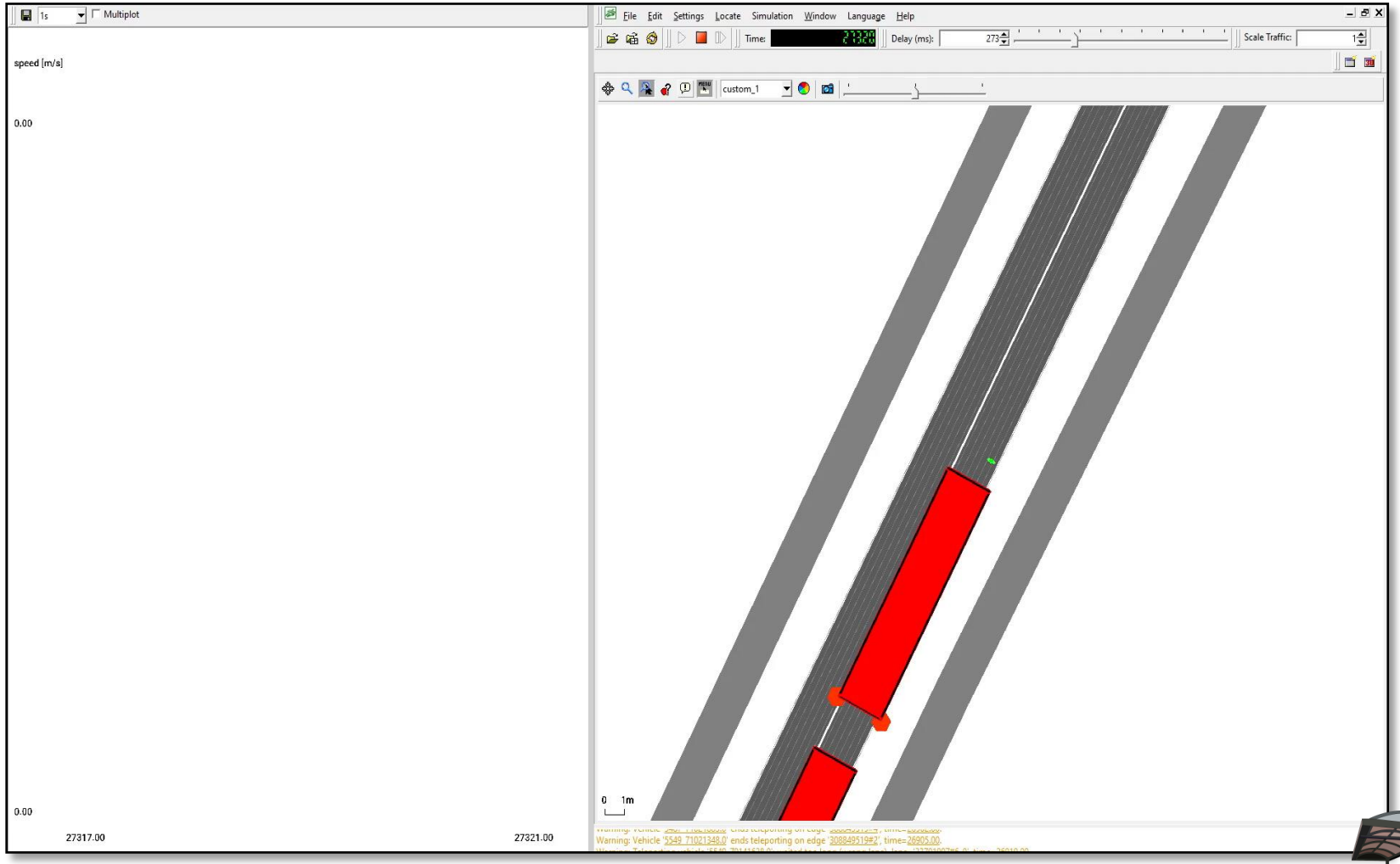


Results

```
<statistics xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://sumo.dlr.de/xsd/statistic_file.xsd">
    <performance begin="06:56:40" end="09:00:00" duration="02:03:20" clockBegin="00:13:39.52" clockEnd="00:15:48.52" clockDuration="00:02:09" traciDuration="00:00:10.06"
                 realTimeFactor="57.36" vehicleUpdatesPerSecond="18473.99" personUpdatesPerSecond="1542.31"/>
    <vehicles loaded="13175" inserted="2048" running="504" waiting="3"/>
    <teleports total="207" jam="121" yield="36" wrongLane="50"/>
    <safety collisions="16" emergencyStops="1" emergencyBraking="1"/>
    <persons loaded="122" running="0" jammed="30"/>
    <personTeleports total="0" abortWait="0" wrongDest="0"/>
    <vehicleTripStatistics count="1544" routeLength="4252.03" speed="4.21" duration="1094.35" waitingTime="335.88" timeLoss="615.97"
                           departDelay="7.81" departDelayWaiting="-1.00" totalTravelTime="19:13:21:23" totalDepartDelay="03:20:57"/>
    <pedestrianStatistics number="263" routeLength="346.79" duration="336.34" timeLoss="86.65"/>
    <rideStatistics number="254" waitingTime="240.27" routeLength="1037.84" duration="194.70" bus="254" train="0" taxi="0" bike="0" aborted="0"/>
    <transportStatistics number="0"/>
</statistics>
```

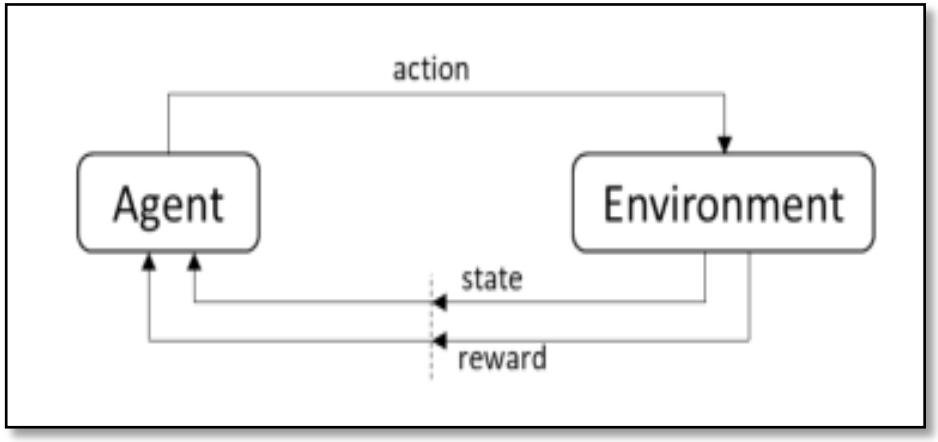
Problems & Solutions

1. **Problems:**
   - Persons Jamming
   - Persons walking on lanes except sidewalks
   - Vehicle jamming, yielding, going to wrong lanes and emergency braking.
2. **Solutions:**
   - Disallow pedestrians to walk on lanes
   - Allow pedestrians walking only on sidewalks.
   - Use netedit to manually make necessary changes in the network.



Reinforcement Learning

- A goal-oriented type of ML technique that guides an autonomous ***agent*** to maximize the agent's ***reward*** throughout repetitions of computational procedures in a certain ***environment***.
- It provides partially stochastic, rule-based modelling outcomes.
- It derives optimal or near-optimal solutions without explicit modelling or formulation of the problem and instructions.
- The key advantage: The ability to learn how to solve an optimal control problem without any external supervision.
- This self-learning ability of the learning agent is the key characteristic that defines the intelligent of the system.
- Traditional RL model has 5 components i.e., agent, action, environment, state and reward.



Reinforcement Learning

- Environment: Mobility simulation in SUMO
- Agents: Persons, buses
- A set of agents interact with and explore an environment, based on which they optimize their behavior to achieve a goal.
- In RL, the interaction with the environment is discretized in steps t.
- For every step, each agent receives information about the current state of the environment $\boldsymbol{s_t}$.
- Based on $\boldsymbol{s_t}$, each agent selects an action $\mathbf{a_t}$ to perform in the environment based on a policy that is learnt over time.
- The current policy **p** determines the behavior and the decisions made by the agents.
- The agents learn this mapping through the exploration of the environment.
- The learning mechanism is based on the reward $\mathbf{r_t}$ received from the environment after performing the action $\mathbf{a_t}$.
- Each agent's goal is to maximize its cumulative reward in the long run.

Summary

- **Input Configuration Files**
  - **OSM files**
    - Road network
    - PoIs
    - Polygons
  - **GTFS files** (gtfs2pt.py)
    - PT routes
    - PT stops
    - PT vehicle types
  - **ODMs from TAZs** (od2trips)
    - OD matrix in "O" or Amitran format
  - **duarouter**
    - Demand (person routes)
- **Output Simulation**
  - **Integration in SUMO**
    - Veh routes + Persons routes
    - Sumo_tripinfo
- **Reinforcement Learning**
  - **Multi-agent environment setting**
    - Optimize KPIs
      - People on bus
      - People waiting on stop
      - Travel Time
      - Waiting time
      - Walking speed

Any Questions?
Thank you!