Using SUMO

(Simulation of Urban MObility)
Slides
What is it?

- SUMO is an open source package which allows you to simulate traffic.
- It gives you a lot of control so that you may fine tune the behavior of the roads and vehicles.
Terms

edge

node

node
Building a simulation

1. Nodes
2. Edges
3. Network (product of nodes and edges)
4. Vehicles
5. Routes
Nodes

- id - unique identifier
- x, y - distance from the origin in meters
- These definitions go in a file with a .nod.xml extension

- Example:

```xml
<nodes>
  <node id="1" x="-250.0" y="0.0" />  
  <node id="2" x="+25050.0" y="0.0" />  
</nodes>
```
Building a simulation

1. Nodes
2. Edges
3. Network (product of nodes and edges)
4. Vehicles
5. Routes
Edges

- id - unique identifier
- from - node car comes from
- to - node car goes to
- These definitions go in a file with a .edg.xml extension

- Example:

```xml
<edges>
  <edge from="1" id="1to2" to="2"/>
  <edge from="2" id="out" to="3"/>
</edges>
```
Building a simulation

1. Nodes
2. Edges
3. Network (product of nodes and edges)
4. Vehicles
5. Routes
Network

- A definition of nodes and edges that can be recognized by other programs (SUMO)
- Netconvert
  - A tool included in the SUMO installation which takes in node and edge xml files and generates a network definition.
  - Example usage from the command line:

```bash
netconvert --node-files=hello.nod.xml --edge-files=hello.edg.xml --output-file=hello.net.xml
```
Building a simulation

1. Nodes
2. Edges
3. Network (product of nodes and edges)
4. Vehicles
5. Routes
Defining Vehicles

- accel - acceleration of the car (m/s^2)
- decel - deceleration of the car (m/s^2)
- id - unique identifier
- length - car length in meters
- maxspeed - car’s maximum speed (m/s^2)
- sigma - driver imperfection (between 0 and 1)

Example:

```
<vType accel="1.0" decel="5.0" id="Car" length="2.0" maxSpeed="100.0" sigma="0.0" />
```
Instantiating a Vehicle

- id - unique identifier
- depart - time in milliseconds before car is spawned
- type - id of a defined vehicle
- route - path taken

- Example:

  `<vehicle depart="1" id="veh0" route="route0" type="Car" />`
Building a simulation

1. Nodes
2. Edges
3. Network (product of nodes and edges)
4. Vehicles
5. Routes
Routes

- id - unique identifier
- edges - a list of edges, separated by spaces, which define the route the vehicle will take

Example:

<route id="route0" edges="edge1 edge2"/>
Building a simulation

1. Nodes
2. Edges
3. Network (product of nodes and edges)
4. Vehicles
5. Routes
Running the simulation

- **sumo-gui**
  - A tool included in SUMO to create a visualization for your simulation
  - All you need to do is provide it with a config file which gives the paths to your routes, network, etc. and it does the rest
Road Types

- You can define different types of edges each with their own attributes in a file with a .typ.xml extension.
- **priority** - the priority of that road based on traffic regulations. Defined by an integer.
- **numLanes** - number of lanes.
- **speed** - max speed on that edge.

- Example:

  ```xml
  <type id="a" priority="3" numLanes="3" speed="13.889"/>
  ```
DEMO 2
Using OpenStreetMap to create networks

- It’s as simple as choosing an area on a map; No XML to define
- Allows you to model real-world locations with ease
Getting the data

- Goto https://www.openstreetmap.org/ to select the area of the world you would like to model
Convert osm data to a network

- We will use netconvert from the commandline with an additional flag

```
netconvert --osm-files file.osm.xml -o file.net.xml
```
Getting polygons from the osm data

- The data from OpenStreetMap also provides figures in that area which allow for a more realistic simulation

```
polyconvert --net-file file.net.xml --osm-files file.osm -o file.poly.xml
```
netconvert --osm-files map.osm -o map.net.xml
polyconvert --net-file map.net.xml --osm-files map.osm --type-file typemap.xml -o map.poly.xml
python /home/kunal/sumo-0.22.0/tools/trip/randomTrips.py -n map.net.xml -e 100 -l
python /home/kunal/sumo-0.22.0/tools/trip/randomTrips.py -n map.net.xml -r map.rou.xml -e 100 -l
Using TraCI

- TraCI is a traffic control interface
- It allows you to alter your simulation in real time using a python script which receives information from the simulation
DEMO 4
Output Analysis
Vehicle Positions

- You can receive the location of every vehicle at a user-specified interval of the simulation.
- This can be useful for detecting traffic jams or any other slowdowns.
Emission

- You can receive data on the amount of car emissions per vehicle
- This would be very useful in making predictions for future amounts of car emissions
- Can help individuals or companies quantify the impact of their travel on the environment
Trip Info

- Data on the departure and arrival time of each vehicle, number of stops, etc...
- Useful for taxi companies in calculating the efficiency of routes taken
More Info

http://sumo.dlr.de/wiki/Simulation_of_Urban_MObility_-_Wiki
Code for Demos

https://github.com/bigolu/sumo-presentation