SUMO

Documentation

From installation folder

- Example of simulations: /sumo/tests/complex/tutorial
- User Documentation : /sumo/docs/web/docs

From the web

- Developer Documentation: https://sumo.dlr.de/docs/Developer/
- User Documentation https://sumo.dlr.de/docs/index.html
- Tutorials: https://sumo.dlr.de/docs/Tutorials/index.html

Installation

- Follow this link (https://sumo.dlr.de/docs/Installing/index.html) to access detailed instructions for installing SUMO.
- To be able to run SUMO simulations from the command line or via a Python application, you should set SUMO_HOME as an environment variable
- To get the latest version, install SUMO by building the source code from the GitHub repository
    - https://github.com/eclipse-sumo/sumo

Installation from a GitHub repository (Linux)

1. sudo apt-get install cmake python3 g++ libxerces-c-dev libfox-1.6-dev libgdal-dev libproj-dev libgl2ps-dev swig default-jdk maven libeigen3-dev
2. Additional requirement
    a. sudo apt-get install python3-dev
    b. sudo apt install libavcodec-dev libavformat-dev libavfilter-dev
    c. sudo apt-get install libopenscenegraph-dev
    d. sudo apt-get install libgtest-dev
    e. sudo apt-get -y install gettext
3. cd /usr/local %% You can choose another directory
4. git clone --recursive https://github.com/eclipse/sumo
5. export SUMO_HOME="$PWD/sumo"
6. mkdir -p sumo/build/cmake-build && cd sumo/build/cmake-build
7. cmake ../..
8. make -j$(nproc)
9. Set SUMO path :
    a. cd && gedit .bashrc
    b. Add these lines at the end of the file
        SUMO_HOME="/usr/local/sumo:/usr/local/sumo/bin"
        export PATH="$PATH:$SUMO_HOME"
        c. Run: . .bashrc

SUMO Tools

1.  Netedit: graphical editor for networks and demand
2.  Sumo-gui: visualize simulations
3.  WebWizard: generate SUMO-compatible road networks and simulations from online map
4. Other tools
    a. Netconvert: Generates a network from node and edge files.
    b. randomTrips.py: Generates demand (traffic/route files) for a given network.

Create Simulation scenario

- A SUMO scenario requires at least the three following files:
  1. SUMO configuration file, with the extension .sumocfg;
  2. Network file, with extension .net.xml;
  3. Route file, with the extension .rou.xml
- All of these files are in XML text format
- It is possible to create a SUMO network either
  - by using NetEdit,
  - coding manually,
  - or with other tools such as street maps, the trip.py script, etc

Network Building(1/2)

- Component
  - Node : junctions that connect Edges
  - Edges: segment of road
  - Lanes: edge is composed of a set of lanes (At least one).
  - Connections: Describe how the incoming and outgoing edges of node are connected
- Process with Netedit :
  1. Create new network 

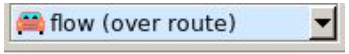
  2. Select Edit network elements
  3. Create edge:  –>  –> click on different places –> <ESC>
  4. Save  network with **.net.xml** file format

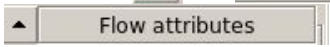
Network Building (2/2)

- Additional operation
  - Edit properties (name, location, etc.) of edge or node
    - Select Set inspect mode button 
    - Select Node or Edge then you edit properties that will appear on the left side
  - Edge operations :
    - Reverse edge
    - Add reverse direction
    - Split edge
  - Edit network shape
    - Set move mode 
    - Select and drag around geometry points on edge to alter its shape

Traffic demand generation (1/2)

- Components
  - Routes : sequence of edges (road segments)
  - Flow: set of repeated vehicles
  - Trip: vehicle movement
- Process with Netedit
  1. Open a saved network
  2. Select Edit Demand Elements 

  3. Create route : 
     a. Select **Create Route Mode**
     b. Select all or some edges –> click on Finish route creation
  4. Adding a vehicle
     a. Select Create Vehicle Mode  -> 

     b. Edit flow attributes 

     c. click over the route we have created, and a vehicle will appear at the beginning of the route
  5. Save (->) the demand elements (routes +vehicles) with **.rou.xml** file format

Traffic demand generation (2/2)

- Comments
  - The order of edge selection is important
  - Splitting the length of the road into multiple edges is necessary when an attribute such as speed or numLanes changes.
  - Generate the traffic demand using **Netedit** is difficult, so it is recommended to define route file by coding

Generate configure file Config file

To run simulation using sumo-gui have to create config file with the extension **.sumocfg**

1. From Netedit, once, we have created network and demand file click on save config file 
2. Or we can create the file manually

Run simulation

- From command line
  - sumo-gui -c path/to/config_file_name.sumocfg (If we want to visualize the simulation)
  - sumo -c path/to/config_file_name.sumocfg (without visualization)
- Using SUMO-gui
  1. Open sumo-gui 
  2. File → Open Simulation;
  3. Select the configuration file
  4. Set delay 

  5. Run the simulation 
- To run simulation another time: need to refresh 

Run simulation from Traci

```
import traci

traci.start(
    [
        "sumo-gui",
        "-c",
        "test.sumocfg",
    ]
)
i = 0
while i < 200:
    traci.simulationStep()
    i += 1
traci.close()
```

Create SUMO Simulation by coding (1/3)

1. Node file : hello.nod.xml

```
<nodes>
  <node id="1" x="-250.0" y="0.0" />
  <node id="2" x="+250.0" y="0.0" />
  <node id="3" x="+500.0" y="100.0" />
  <node id="4" x="+500.0" y="-100.0" />
</nodes>
```

2. Edge file : hello.edg.xml

```
<edges>
    <edge from="1" id="1to2" to="2" type="type3" />
    <edge from="2" id="2to3" to="3" type="type2" />
    <edge from="2" id="2to4" to="4" type="type3" />
</edges>
```

3. Edge type file (optional): hello.type.xml

```
<types>
    <type id="type3" priority="3" numLanes="3" />
    <type id="type2" priority="3" numLanes="2" />
</types>
```

Create SUMO Simulation by coding (2/3)

4. Generate network :
   - from command lane run this command

     netconvert --node-files hello.nod.xml --edge-files hello.edg.xml -t hello.type.xml -o **hello.net.xml**
   - A new file will be created named : hello.net.xml

5. route file : hello.rou.xml

```
<routes>
    <vType id="car" accel="1.0" decel="5.0"  length="2.0" maxSpeed="100.0" color="red" />
    <vType  id="bus" accel="1.0" decel="5.0" length="12.0" maxSpeed="50.0" color="white" />

    <route id="route0" edges="1to2 2to3" />
    <route id="route1" edges="1to2 2to4" />

    <flow id="f_0" route="route0"  begin="0.00" end="3600.00" vehsPerHour="3000" type="car"/>
    <flow id="f_1" route="route1"  begin="0.00" end="3600.00" vehsPerHour="2000" type="bus"/>
</routes>
```

Create SUMO Simulation by coding (3/3)

6. View setting file (optional) : viewsettings.xml

```
<viewsettings>
    <scheme>
        <vehicles vehicleQuality="2"/>   <!-- define  graphical representation of the vehicles  -->
        <edges  laneShowBorders="1"/>
    </scheme>
</viewsettings>
```

7. Config file

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
    <input>
        <net-file value="hello.net.xml" />
        <route-files value="hello.rou.xml" />
        <gui-settings-file value="viewsettings.xml"/>
    </input>
    <time>
        <begin value="0" />
        <end value="2000" />
        <!-- the simulation will begin at 0 and end at 2000 seconds. -->
    </time>
</configuration>
```

References

- https://github.com/nsaunier/CIV8740/blob/master/guide-sumo.md (Fr)
- https://github.com/nsaunier/CIV8740/blob/master/manual-sumo.md (Eng)
- https://sumo.dlr.de/docs/Tutorials/SUMOlympics.html (to do)