SUMO Tutorial

Jakob Erdmann

SUMO2017, Berlin

Knowledge for Tomorrow

Outline

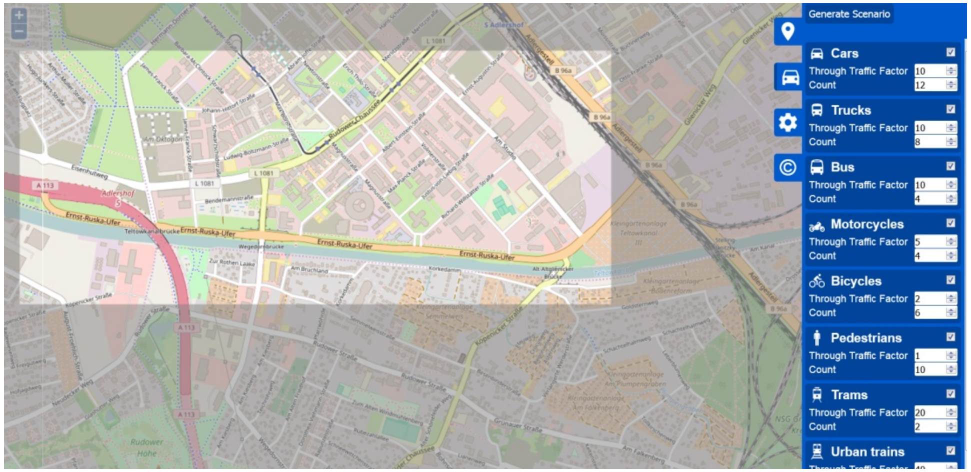
- Prerequisites
- 3-Click scenario generation with osmWebWizzard.py
- Fixing the network with Netedit
- Adapting and calibrating demand
- Modeling public transport and intermodal routing
- ParkingAreas
- Conclusion

Prerequisites

- SUMO 0.30.0 or latest development version sumo.dlr.de/wiki/Downloads
- Python: www.python.org/download/releases/2.7/
- Text Editor (i.e. notepad-plus-plus.org/)
- Data files: sumo.dlr.de/daily/sumo2017_tutorial.zip

osmWebWizzard

- Getting a basic scenario with tools/osmWebWizzard.py
  - Mode-specific network options
  - Random traffic
- Configure
  - Area
  - Traffic modes
  - Traffic volume
  - Fraction of through-traffic
- Generated files allow rebuilding and adapting the scenario
- Sample data in `adlershof_wizard`



Cars ☑
Through Traffic Factor 10
Count 12

Trucks ☑
Through Traffic Factor 10
Count 8

Bus ☑
Through Traffic Factor 10
Count 4

Motorcycles ☑
Through Traffic Factor 5
Count 4

Bicycles ☑
Through Traffic Factor 2
Count 6

Pedestrians ☑
Through Traffic Factor 1
Count 10

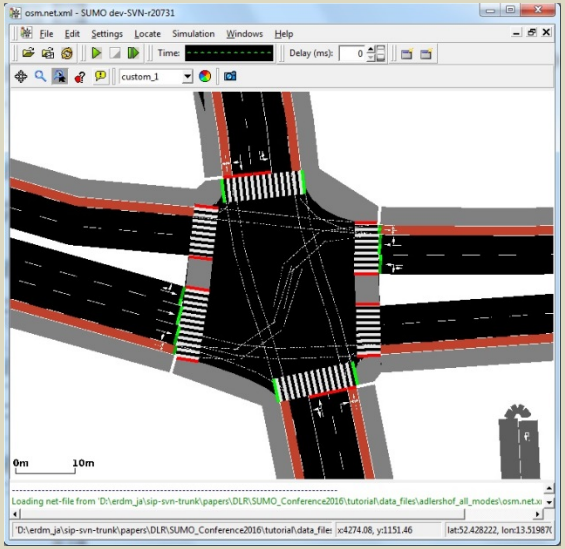
Trams ☑
Through Traffic Factor 20
Count 2

Urban trains ☑

The current state of intermodal junctions

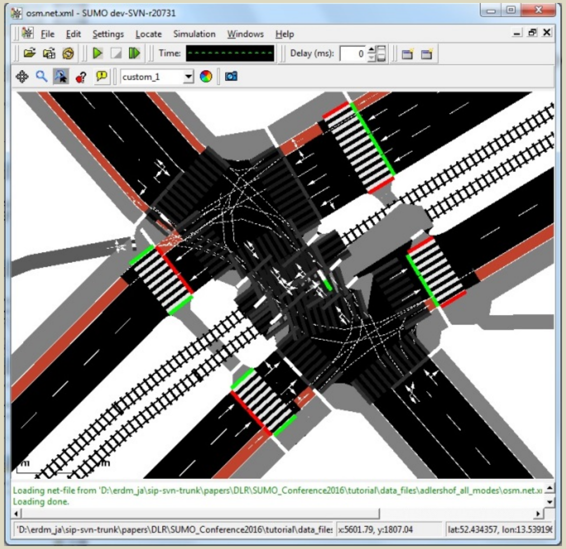
last-years slide

- The Good



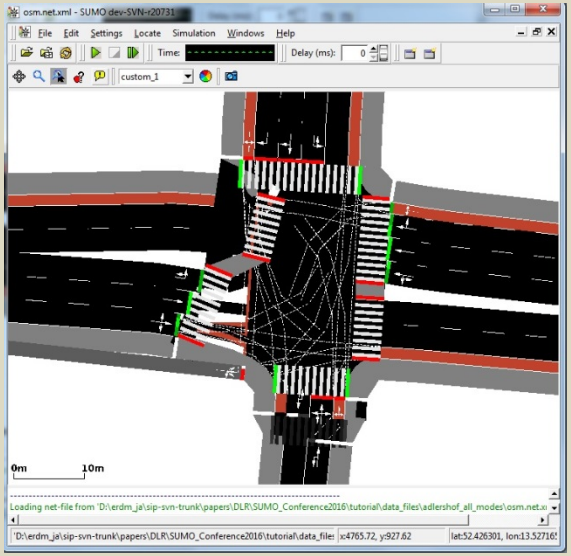
```
--------------------------------------------------------------------------------
Loading net-file from 'D:\erdm_ja\sip-svn-trunk\papers\DLR\SUMO_Conference2016\tutorial\data_files\adlershof_all_modes\osm.net.x
'D:\erdm_ja\sip-svn-trunk\papers\DLR\SUMO_Conference2016\tutorial\data_file  x:4274.08, y:1151.46  lat:52.428222, lon:13.51987
```

- The Bad



```
Loading net-file from 'D:\erdm_ja\sip-svn-trunk\papers\DLR\SUMO_Conference2016\tutorial\data_files\adlershof_all_modes\osm.net.x
Loading done.
'D:\erdm_ja\sip-svn-trunk\papers\DLR\SUMO_Conference2016\tutorial\data_file  x:5601.79, y:1807.04  lat:52.434357, lon:13.53919
```

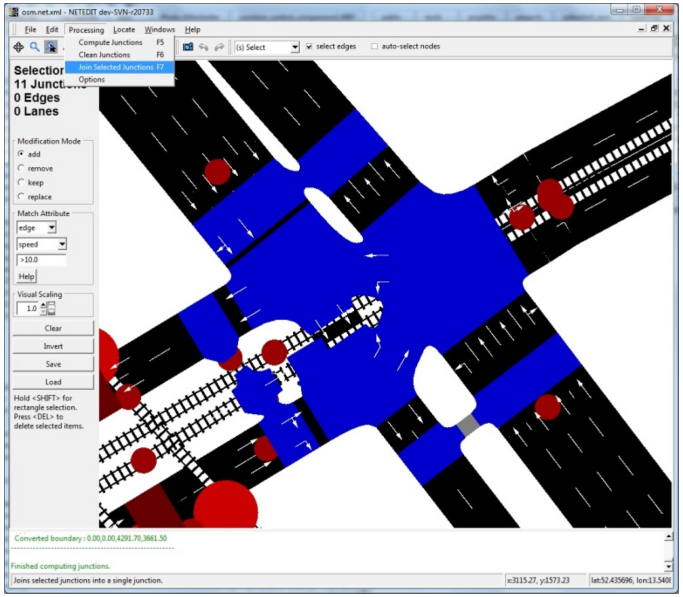
- The Ugly



```
--------------------------------------------------------------------------------
Loading net-file from 'D:\erdm_ja\sip-svn-trunk\papers\DLR\SUMO_Conference2016\tutorial\data_files\adlershof_all_modes\osm.net.x
'D:\erdm_ja\sip-svn-trunk\papers\DLR\SUMO_Conference2016\tutorial\data_file  x:4765.72, y:927.62  lat:52.426301, lon:13.52716
```

Fixing network problems

- Open network in NETEDIT (ctrl-t in SUMO-GUI)
- Crop network (selection mode)
  - Result in `adlershof_cut_network`
- Remove railways and waterways
  - (selection mode, filter, delete)
- Join complex junctions (select, F7)
  - Result in `adlershof_fixed_junctions`



(s) Select
☑ select edges
☐ auto-select nodes

Selection
11 Junctions
0 Edges
0 Lanes

Modification Mode
● add
○ remove
○ keep
○ replace

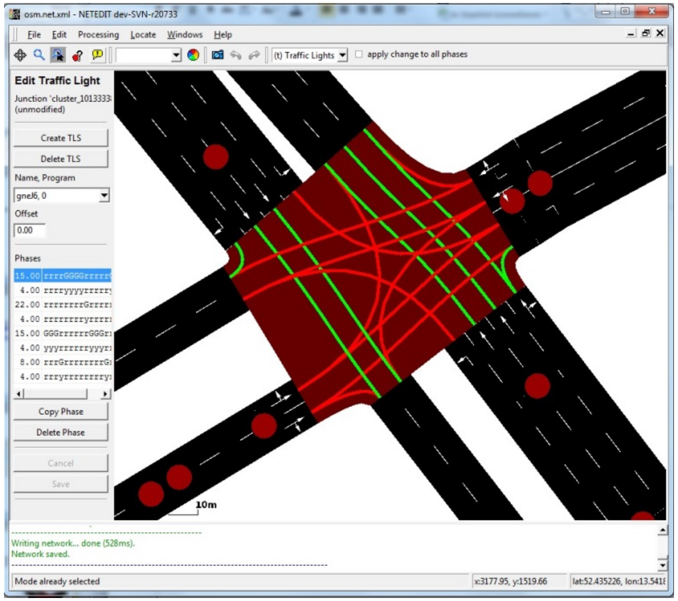
Match Attribute
edge
speed
>10.0

Visual Scaling
1.0

Hold <SHIFT> for rectangle selection. Press <DEL> to delete selected items.

```
Converted boundary : 0.00,0.00,4291.70,3661.50
-------------------------------------------------
Finished computing junctions.
```

Joins selected junctions into a single junction. x:3115.27, y:1573.23 lat:52.435696, lon:13.540



(t) Traffic Lights
☐ apply change to all phases

Edit Traffic Light

Junction 'cluster_1013333
(unmodified)

Name, Program
gneJ6, 0

Offset
0.00

Phases

```
15.00 rrrrGGGGrrrrr
 4.00 rrrryyyyrrrry
22.00 rrrrrrrrGrrrr
 4.00 rrrrrrrryrrrr
15.00 GGGrrrrrrGGGr
 4.00 yyyrrrrrryyyr
 8.00 rrrGrrrrrrrrG
 4.00 rrryrrrrrrrry
```

```
-------------------------------------------------
Writing network... done (528ms).
Network saved.
-------------------------------------------------
```

Mode already selected x:3177.95, y:1519.66 lat:52.435226, lon:13.541

Adapting Demand

- regenerate random demand (edges changed when patching network)
    - Run build.bat (calls randomTrips.py with preset parameters)
    - `--period 0.8` (increased demand)
    - `--speed-exponent 2` (more traffic on fast edges)
    - `--lanes` (more traffic on multi-lane roads)

- Additional traffic leads to deadlock at turn-arounds on short edges. Fix with NETEDIT
    - Result in `adlershof_demand`

Calibrating Demand

```
<routeProbe id="probe1" edge="-190083608#1" freq="60"
    file="calibrator_out.xml"/>

<calibrator id="cali1" lane="-190083608#1_0" pos="0"
    output="detector.xml"
    routeProbe="probe1">
    <route id="fallback" edges="-190083608#1"/>
    <flow begin="0" end="1800" route="fallback"
        vehsPerHour="3000" speed="20.0" departLane="best"
        departPos="180" color="blue"/>
    <flow begin="1800" end="3600" route="fallback"
        vehsPerHour="500" speed="5.0" departLane="best"
        departPos="180" color="red"/>
</calibrator>
```

Result in `adlershof_calibrate`

Public Transport (1)

- import bus stops from OSM
  - `netconvert -c osm.netccfg --pstop-output all_stops.add.xml`
    - `Run in folder 0_adlershof_webwizard`
  - Open *osm.net.xml* in NETEDIT and 'Load Additionals'
    - Stops that do not fit into the cropped network are discarded.
    - Save as *stops.add.xml*
      - Add attribute friendlyPos="true" to all stops (workaround for a Netedit bug discovered last week)
      - Fix stop lanes (_2 instead of _0) (workaround for a Netconvert bug discovered last week)

Public Transport (2)

- Define bus route - still not automatic)-:

```
<flow id="bus162" begin="0" end="1800" period="300"
     from="318210395" to="143308549#1" line="162">
    <stop busStop="306982602" until="130"/>
    <stop busStop="464396589" until="200"/>
</flow>
```

- Let persons decide whether to take the bus or walk
  - In *build.bat* add option --persontrips in the first line
  - `duarouter -t osm.pedestrian.trips.xml -o osm.pedestrian.rou.xml -n osm.net.xml --additional-files stops.add.xml,bus.add.xml --ignore-errors`

- Result in `adlershof_public_transport`

Parking Areas

- Define road-side parking and a car park

```
<parkingArea id="roadside" lane="143308555#2_1" startPos="3"
     endPos="57" roadsideCapacity="9"/>
<parkingArea id="carpark" lane="318210377#1_2" startPos="0"
     endPos="2">
   <space x="5289.90" y="1012.82" length="5" angle="315"/>
   …
</parkingArea>
```

- Define parking demand

```
<flow id="shopping" begin="0" end="1800" period="60"
     from="81639675#0" to="143308546#19">
    <stop parkingArea="roadside" duration="900"/>
</flow>
```

Parking Area Rerouting

- Define alternative parking area

```
<rerouter id="myRerouter" edges="143308555#2">
   <interval begin="0" end="3600">
      <parkingAreaReroute id="roadside"/>
      <parkingAreaReroute id="carpark"/>
   </interval>
</rerouter>
```

- Result in `adlershof_parkingArea`

Conclusion

- Use tools/osmWebWizzard.py to get a quick start
- Read the documentation / FAQ at http://sumo.dlr.de/wiki
- Report any bugs you find to sumo-user@lists.sourceforge.net
- Share your scenarios and results

- Talks to us. We are always looking for project partners! sumo@dlr.de