Interface between proprietary controllers and SUMO

SUMO2014 Modeling Mobility with Open Data
May 15+16, 2014 - Berlin-Adlershof, Germany

Robbin Blokpoel
Senior researcher at Imtech Traffic & Infra,
robbin.blokpoel@imtech.com, +31 629409806

Contents

Introduction

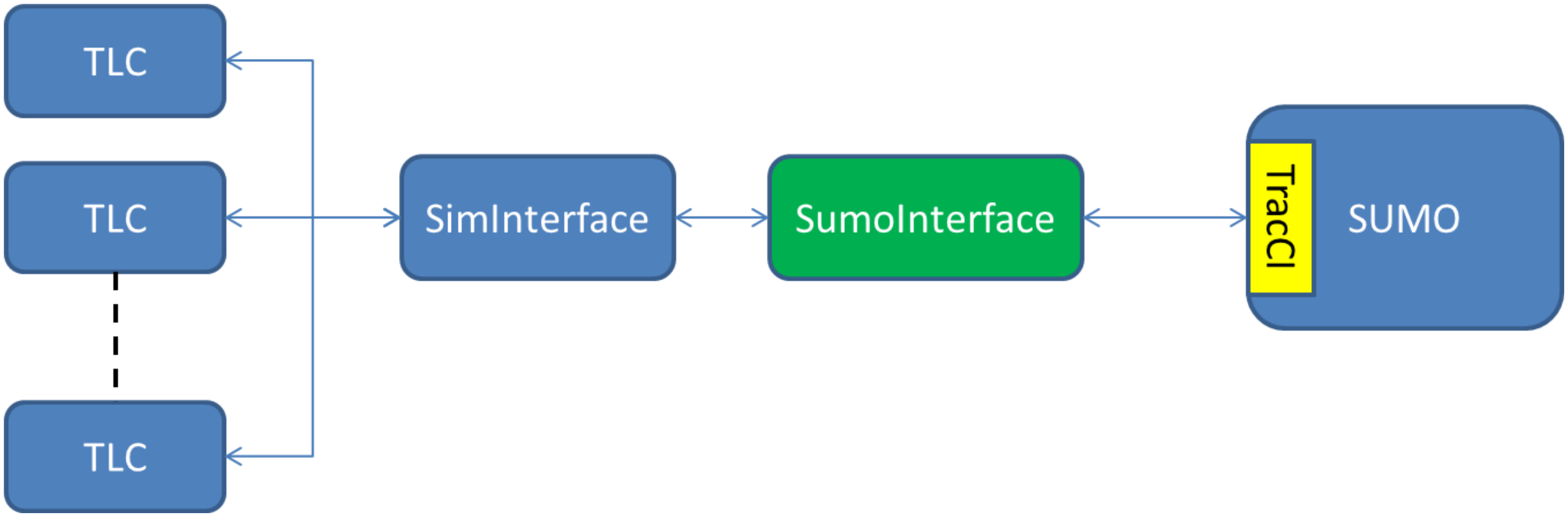
Why simulate controllers in SUMO?

- Simulation studies required before deployment
- Commercial packages costly
- Realism comparable to commercial alternatives
- Research requires full control:
  - ✓ TraCI interface, access to almost everything
  - ✓ Open source allows for quick extensions

Why external controllers?

- Real-world controllers give realistic results
- Copying them into SUMO source very complex:
  - Many different local rules
  - Dynamic behaviour according to complex algorithms

Architecture - components



Only the SumoInterface is a new component, the others are reused

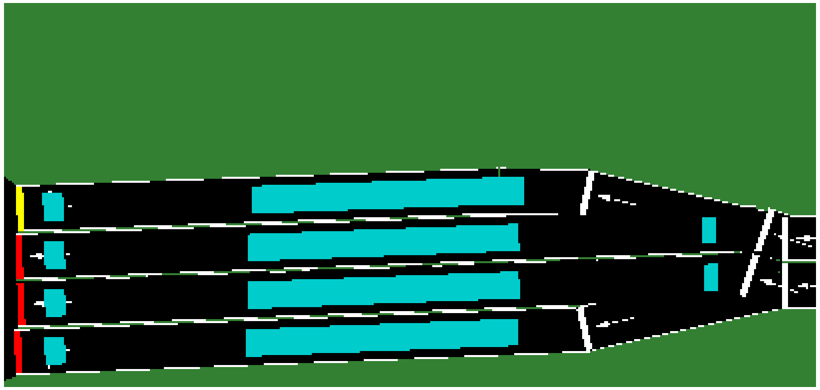
Architecture – process flow

1. Start up controller executables
2. Read signal group conversion file (see signal groups)
3. Start up SUMO
4. Request SUMO detector list

5. Execute a SUMO timestep (100ms)
6. Request SUMO detector status
7. Update detector status in SimInterface dll
8. Execute a TLC timestep through SimInterface
9. Request signalgroup status from SimInterface dll
10. Update signalgroup status in SUMO
11. Go back to 5

Detection – using E2 detector

- Addition of E2 (lane area) detector to Traci required
- Command 0x8E (get laneAreaDetector), variable 0x10 (number on loop)
- Long area detectors often used for vehicle actuated:
  - Vehicle leaves loop close to stopline, accurate moment for amber
  - Length of loop detects gaps and thus end of platoon



Detection – identification with TLC

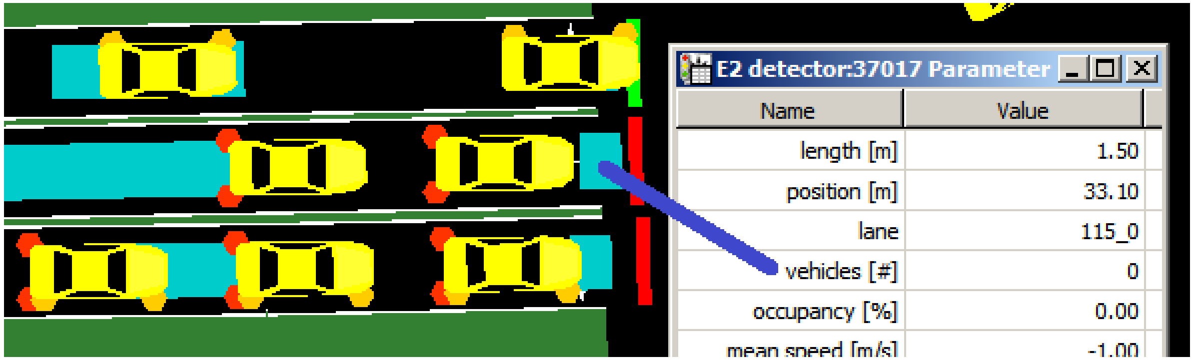
- In the TLC, detectors are simply numbered 0,1,2,…
- Skipping numbers possible on simulator side
- Conversion table detector "SG1_entry = controllerID 2" possible
- Previously used solution for Vissim was numbering convention:

  Intersection ID *1000 + detector number
- Logical names can be added as comment

```
<laneAreaDetector id="37000" lane="29_0" pos="40.394" length="1.5" freq="3600" file="NUL" /> <!--d011-->
<laneAreaDetector id="37001" lane="29_0" pos="7.963" length="20.0" freq="3600" file="NUL" /> <!--d012-->
<laneAreaDetector id="37002" lane="89_0" pos="38.517" length="1.468" freq="3600" file="NUL" /> <!--d021-->
<laneAreaDetector id="37003" lane="89_0" pos="0.996" length="25.0" freq="3600" file="NUL" /> <!--d022-->
<laneAreaDetector id="37005" lane="90_0" pos="40.202" length="1.5" freq="3600" file="NUL" /> <!--d031-->
```

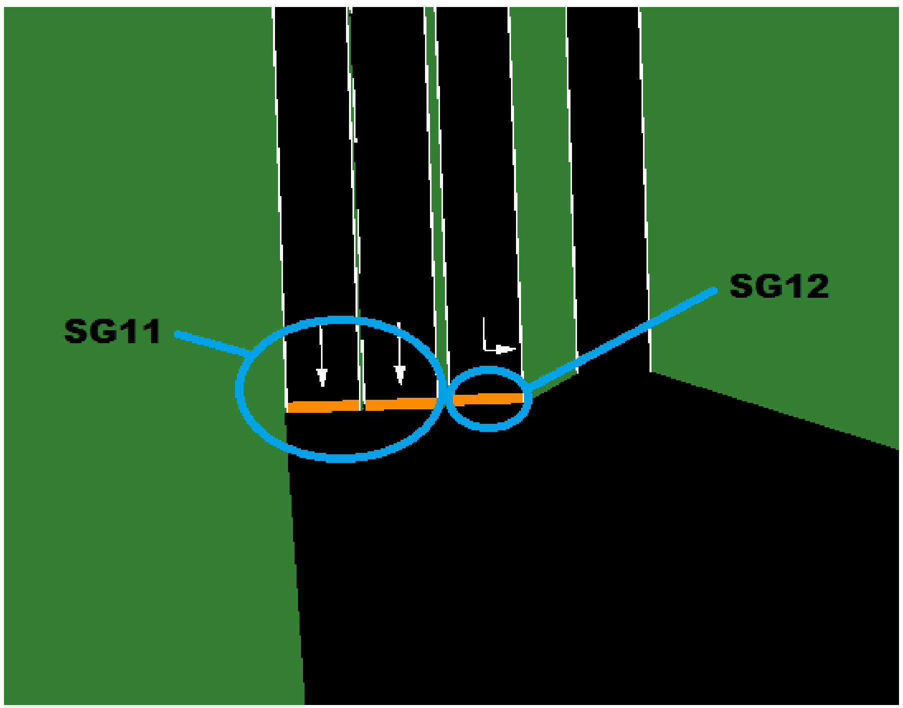
Detection – positioning and timing

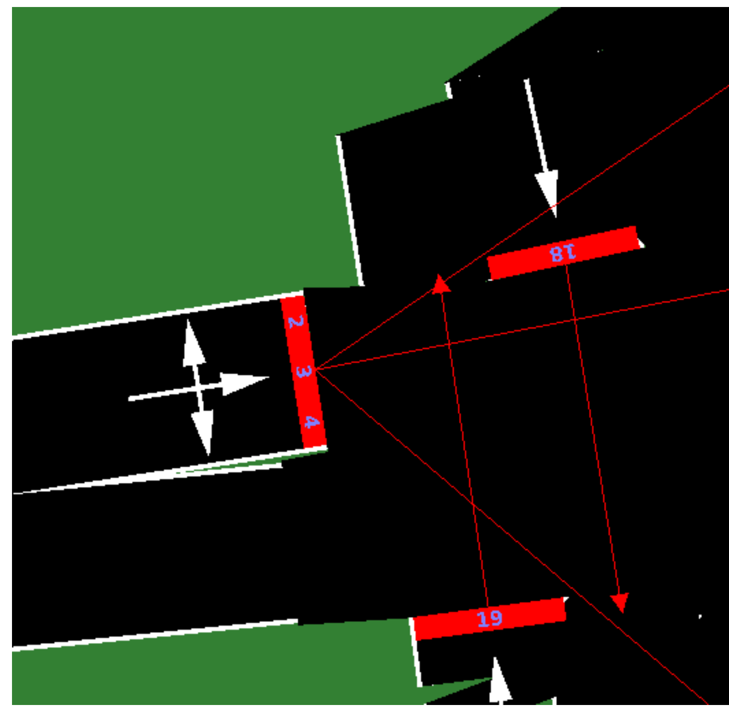
- Short fast vehicles require regular polling
  - Motorcycle of 2m length on 1m loop at 30 m/s occupies loop for 100ms
- Can be slower for urban situations
- Distance to stopline and default stopping distance important
  - SUMO 0.19.0 2.5m, 0.20.0 1.0m



Signal Groups - numbering

- Signal group, always green at the same time
- Signal head, per lane per direction





Signal Groups - conversion

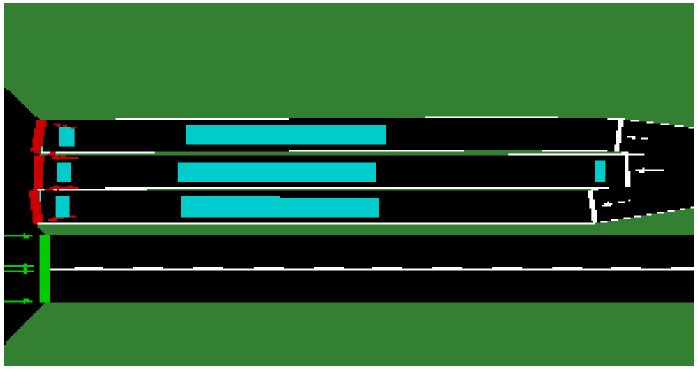
- Numbering convention not possible
- Conversion XML file required

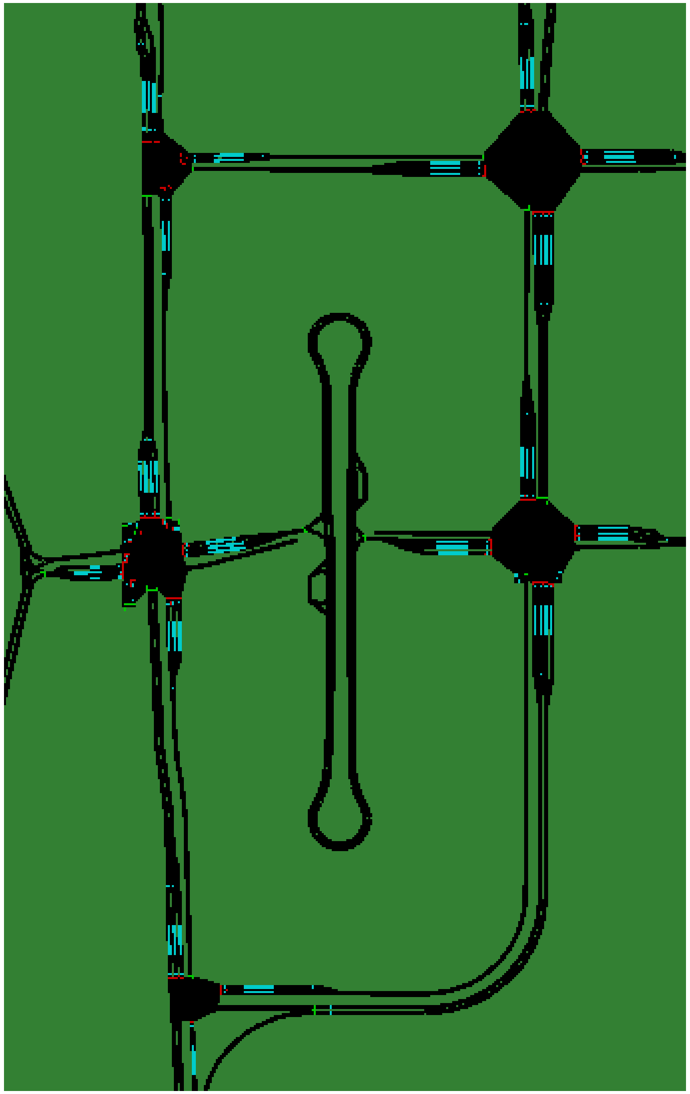
```
<intersection id ="37">
    <signalgroup id="37000" sumoSGs="7"/>
    <signalgroup id="37001" sumoSGs="19"/>
    .
    .
    <signalgroup id="37019" sumoSGs="24,12"/>
</intersection>
```

- SUMO state gGyYrRoO
- TLC state enum, flashing states alternate with O
- Red + amber = red in SUMO
- Flashing red = red
- Update every second command 0xC2, new state tuple 0x20

Simulation speed

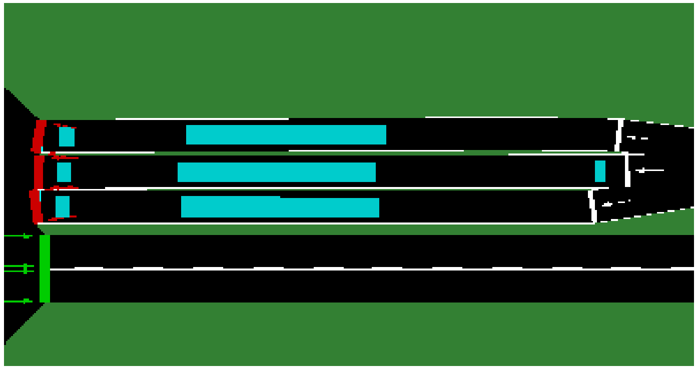
- Network with large amount (168) of detectors
- Core 2 duo 2.53 GHz
- Single Traci calls 1x real-time speed
- 1 call per intersection 2x real-time speed
- No calls for detection 50x real-time speed (Imflow limiting factor)

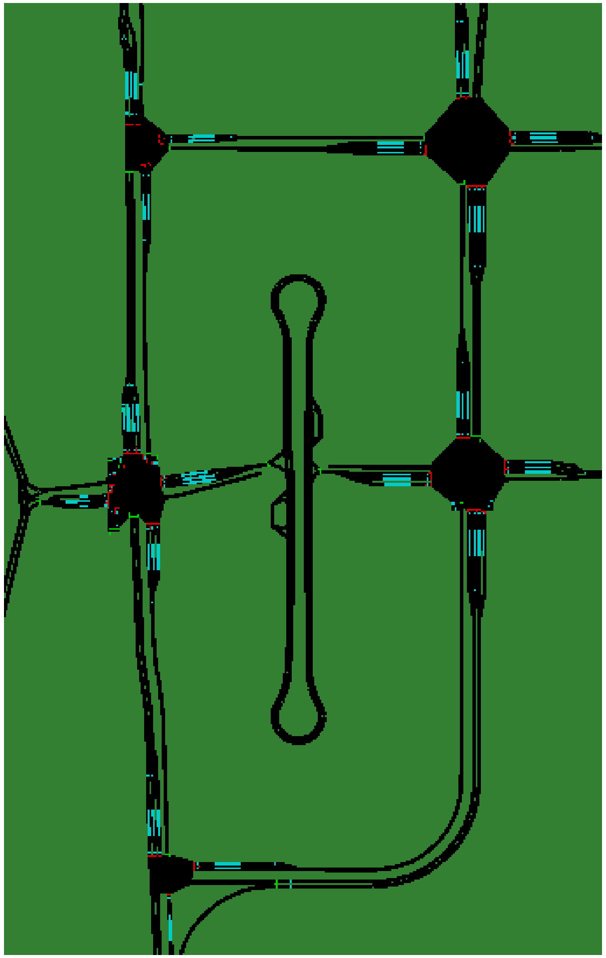




Comparison with Vissim – simulation scenario

- Pedestrians + bikes at 1 intersection
- 1500 vehicles per hour north-south
- Conflicting large streams at bottom intersection
- Demands created with duarouter
- Evaluation with MeMe/E3 detectors
- Position could be optimized





Comparison with Vissim - results

- Vehicle counts not accurate, only 35% measured in busy areas
- Free flow time acquired using a run with all “O”
- Could be distance divided by desired speed as well
- Pedestrian/bike delay 2.0 seconds higher than Vissim
- Vehicle delay 1.3 lower than Vissim
- Standard SUMO settings were used and no specific pedestrian model, this leaves room for improvement
- Poisson distribution adds realism

Conclusion

- Possibility to couple real-world controllers to SUMO allows commercial urban simulations with SUMO
- Detector and signal group translation
- Simulation speed needs to be addressed
- Results comparable to Vissim

Questions

?