SUMO Workshop

2/5: Simulation Requirements and Issues

Deutsches Zentrum
für Luft- und Raumfahrt e.V.
in der Helmholtz-Gemeinschaft

Current and future Developments

Up to V1.0

- TraCI-Rework
- Multi- and Intermodality
- Debugging microsim

Beyond

- Opening networks for on-line changes
- Opening car-following and lane-changing models
- Verification / adaptation

SUMO Workshop

1/5/1: Reworking TraCI

Reworking TraCI

Removing old SUMO string ID to integer ID mapping

Reasons

- Internal mapping requires pre-parsing of all input files (slow, double code, additional memory usage)
- Not all types were supported; adding new ones requires additional mappings (containers, parsers, access)
- Extending types requires changes in two places
- Highly inflexible by assigning all vehicles a-priori to equipped/not equipped using a random number

Nice side-effects

- More accessed types
- More access possibilities

Any problems? Which?

Reworking TraCI
Vehicle Manipulation

Up-to-now

- Quite dirty in-line manipulations of vehicle speeds or latera movements
  - Not following code style
  - Adding additional code even if not used

Wanted changes

- Manipulations of vehicles are done by adding manipulation “devices”
  - Only there if needed
- Consolidation of MAX_SPEED, SLOWDOWN, etc. (does not work anyhow)

Discussion

SUMO Workshop

1/5/2: Simulation and Models

Deutsches Zentrum
für Luft- und Raumfahrt e.V.
in der Helmholtz-Gemeinschaft

Simulation State
Increasing Complexity

Extensions and resulting problems

- Junction-internal lanes
  - Jamming place on intersections
  - Unverified right-of-way rules
- Complex traffic lights
- Subsecond time-steps
  - New, vehicle-based right-of-way computation suffers from abolition of a scheduler (no deadlock solving mechanism)
- Running gag: lane-changing
  - Always posing several ways to improve
  - But too complex and implicite already; should be reworked

Simulation State
Increasing Complexity

Points to Discuss

- Current approach:
  - We have several versions here and there. Hack everything into SUMO, examine how it works…
  - … as a result the SVN version will be probably quite odd at some time
  - Try to catch problems by more tests
- Testing
  - We have UnitTests and TextTest tests. Both are inappropriate for multi-agent systems
  - Any ideas on how to test a microscopic traffic simulation? Sure, “evaluation”. But what about single agent’s model complexity against the whole simulation’s behavior?

Simulation State
Models

Some generic questions

- What is assumed to be the best solution
  - Hacking own models to make the simulation run, or
  - stick to published models, assuring their correct implementation, even if the simulation is not working perfectly

  (clean scientific work vs. a running simulation?)
- Any comments on documentation on models? Any ideas how to improve?