Intelligent transportation systems

Miklós Máté

2016 September 29

VANET Simulations

- Vehicular Ad-hoc NETwork
- We want to examine wireless communication among vehicles in road traffic
- This may include motorcycles, bicycles, pedestrians, trains, buses etc.
- The main difference compared to MANET is the movement model
  - Movement is a challenge: find new paths fast and often
  - Sensors: random walk, Brownian motion
  - Cars move much faster than typical sensors (if they move at all)
  - Their movement pattern is more restricted: they follow the road network (except tank)

- What is a simulation?
  - We create a model of the subject
  - Numerical computations on the abstract model
- Why?
  - Experiments are expensive
  - Experiments may be infeasible

Simulating road traffic

- Goal: road planning, city planning, tuning traffic light programs, *examine inter-vehicle communication*
- Granularity
  - Microscopic: vehicles are modeled individually with their paths, acceleration etc.
  - Macroscopic: a flow model for vehicle densities (testing the capacity of the road network)
  - Mesoscopic: somewhere in between, e.g. modeling groups of vehicles
- Output: data dump (positions, accelerations, damage), statistics, visualization
- Examples: MATSim, SUMO, PTV VisSim

SUMO

- Simulation of Urban MObility
- Developed by DLR: Forschungszentrum der Bundesrepublik Deutschland für Luft- und Raumfahrt
- Institute of Transportation Systems (Institut für Verkehrsystemtechnik), Berlin
- Open Source (GPLv3), supported platforms: Windows, Linux, (macOS)
- Current version 0.27.1
- Microscopic simulation
- It has a graphical user interface, but it can be run in batch mode from command line
- http://sumo.dlr.de/
- http://sumo.dlr.de/wiki/Main_Page

SUMO

- Multi-lane roads
- Programmable traffic lights
- Public transport and pedestrians
- Detects collisions (by default things don't collide)
- Not just cities, one can simulate forklifts in a warehouse
- TraCI: Traffic Control Interface
  - SUMO listens on a TCP port, an external program can connect and control the simulation
  - There is a precisely specified network protocol (not XML, library for C++, Java, Python)
  - Step simulation
  - Query properties and states: parts of the road network, vehicles, traffic lights, GUI
  - Intervene: add/remove/change vehicles/roads, control traffic lights, control what the GUI shows

SUMO

- Creating road network for the simulation
  - NETCONVERT: import network from a variety of formats
    - SUMO native descriptions: .nod.xml, .edg.xml, .con.xml, .typ.xml, .tll.xml → .net.xml
    - Map data
    - File formats of other simulators (pl. Vissim, MATSim)
  - NETGENERATE: generate abstract shapes (grid, spider, random)
  - POLYCONVERT: imports geometric shapes from the map that are not part of the road network (e.g. outlines of buildings)

- Generating vehicle movements
  - Theory: observe real traffic, import the data
  - Trip: start and end points; Route: exact path including lanes
  - MAROUTER: shortest path routing based on O/D matrix
  - ACTIVITYGEN + DUAROUTER: generate population density, generate trips, generate routes
  - JTRROUTER: given trip list, random walk with given turn probabilities in intersections
  - DFROUTER: routes based on vehicle density measurements
  - randomTrips.py: random trips, optionally calling DUAROUTER

- NETEDIT: graphical network editor
  - public since 0.25
  - Not integrated into the main GUI
- MESO: mesoscopic simulation
  - Public since 0.26
  - Uses a queueing model instead of moving the vehicles
  - They promise it to be 100x faster than the default simulator
 - Needs the same input and generates the same output

OpenStreetMap

- Community-created map
- NOT copied from other maps, but their own cartographic measurements (GPS), the maps of cartographic companies are copyrighted!
- Worldwide community
- Landmarks are not just the paying ones (like in Google Maps) but everything the community deemed interesting
- The database is freely downloadable and usable with attribution
- The local community creates it for itself, so the place marks are only available in the local language
- http://www.openstreetmap.org
- https://blog.openstreetmap.org/2012/01/17/google-ip-vandalizing-openstreetmap/

Demo!
(if it works)

- Vehicles in Network Simulation
- Simulate wireless communication among vehicles
- SUMO and OMNeT++ integrated with the TraCI interface
- Current version is 4.4
- http://veins.car2x.org/

Veins: OMNeT++

- Generic framework for event-based simulations
- Open source, freely available for academic use, OMNEST for commercial use
- Supported platforms: Windows, Linux, macOS
- Started as a Hungarian project (Varga András, BME-HIT)
- Current version: 5.0 (I still use 4.6)
- The simulator itself and the simple models are C++
- NED (NEtwork Description) language for constructing combined models and networks
- Theoretically not only computer network simulators are possible
- Module, Gate, Channel, Message, Packet, Network
- Eclipse-based integrated development environment
- Simulations can be controlled and inspected with GUI
- IDE has data collection, processing, visualizing capabilities
- Horribly slow
- `https://omnetpp.org/`
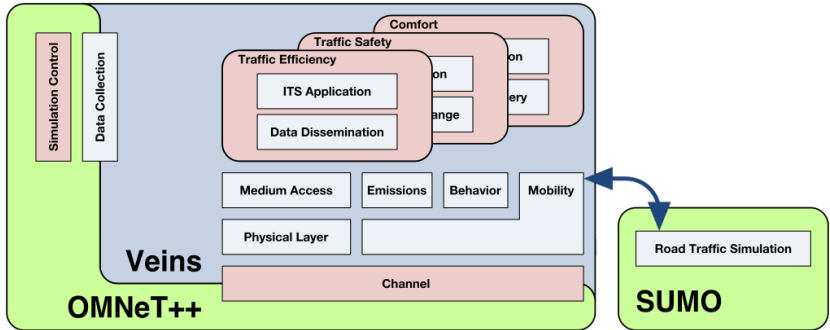
Veins

- This is a model library in OMNeT++
- First SUMO has to start with the desired map+routes in TraCI mode, then the Veins model can connect to it
- The simulation is controlled via the OMNeT++ interface, and the vehicles can be observed in the SUMO GUI
- It uses its own models for the network equipments instead of e.g. INET or MIXIM
- Focuses on IEEE 802.11p és IEEE 1609.4 DSRC/WAVE systems
- It's OMNet++ so it's easily expandable

PTV Vissim
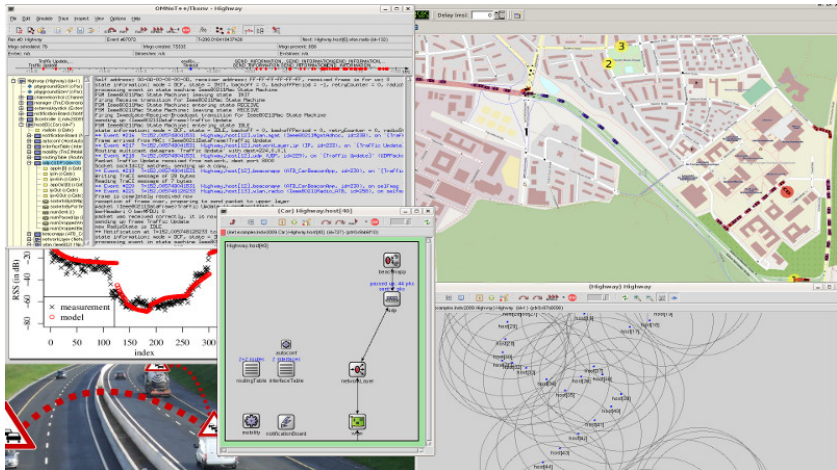
- This is the market-leading traffic simulator
- Planung Transport Verkehr AG (Karlsruhe), "Verkehr In Städten - SIMulationsmodell"
- Costs a lot of money, only supports Windows
- Based on the marketing it knows everything imaginable, but I couldn't try it out
- `http://vision-traffic.ptvgroup.com/en-uk/products/ptv-vissim/`
- Two demonstration videos
  - `https://www.youtube.com/watch?v=OtYby7QnyAE`
  - `https://www.youtube.com/watch?v=Ju9BbC2914I`