Traffic Simulation with the TRASS Framework

Ulf Lotzmann

Institute of Information Systems Research, University of Koblenz,
Universitätsstraße 1, Koblenz 56070, Germany
ulf@uni-koblenz.de

In order to simulate route choice behavior of agents in complex and heterogeneous environments, we developed an agent-based TRAffic Simulation System (TRASS) suitable for traffic simulation with a wide range of freedom concerning agent design – particularly in terms of implementing agent behavior[1].

In TRASS, objects like traffic participants (e.g. cars, pedestrians, traffic lights) and topographical regions are regarded as agents.

The entities of the first kind of agent are classified into two subtypes. “Simple” agents are of the reactive type and feature properties such as shape (a freely definable set of circles), position, direction, velocity. They include custom code for reacting on requests by other agents or external events (triggered by timer or agents). A more complex type of agent extends the reactive type by a powerful sensor unit (a freely definable set of circular-sector-shaped perception areas) in combination with custom code for perception of the environment and implementation of cognitive behavior.

The second type of agent – the topographical region – embrace behavioral properties of the reactive agent type, but otherwise exhibit significant differences, as they are static (in terms of size and position) and polygon-shaped objects.

In principle, the customization of agent behavior has to be done by implementing Java methods, which are predefined by kernel interfaces. The proceeding is similar to the way of using other well-known simulation frameworks like Repast and Mason.

In the TRASS framework, the agent behavior is encapsulated within so-called strategy modules, which are Java objects that can be attached to respective agent objects.

In order for agents to navigate in complex environments, the agent-strategy design has to cover two distinct aspects. First, an agent must be able to interact within his physical environment directly. This includes recognition and classification of agents and topographic regions as well as conducting elementary actions (e.g. driving a curve, changing a lane) within this environment. Second, an agent needs some kind of reasoning to make strategic decisions with the aim to produce goal-driven behavior.

For this reason we use a two-layered approach. The first facet – recognition and basic action – is more of a technical nature and therefore packed into a so-called “robotics layer”. The techniques used in this layer, such as state machines, are quite similar to those used to control autonomous robots. The other layer – called “AI layer” – comprises all aspects subsumed under the keyword reasoning. This layer manages the robotics layer and can be seen as the link to the social simulation domain.

---

[1] Lotzmann, U.: Design and Implementation of a Framework for the Integrated Simulation of Traffic Participants of All Types. In: Bruzzone, A. G. et al (eds.): Proceedings of the International Mediterranean Modelling Multiconference, Barcelona (2006)