Multilayered Multiagent System for Traffic Simulation (Demonstration)

Rafik Hadfi
Department of Computer Science and Engineering
Nagoya Institute of Technology
Gokiso, Showa-ku, Nagoya, 466-8555 Japan
rafik.hadfi@nitech.ac.jp

Takayuki Ito
Department of Computer Science and Engineering
Nagoya Institute of Technology
Gokiso, Showa-ku, Nagoya, 466-8555 Japan
ito.takayuki@nitech.ac.jp

ABSTRACT

We propose a multilayered multiagent simulator that can simulate traffic in any urban environment on earth, subject to specific weather conditions. We adopt an agent-based approach for the behaviors of the vehicles and the drivers. We additionally propose a behavioral model to realistically emulate the driving behaviors of humans.

Keywords

Simulation; Multiagent Systems; Traffic Simulation; Geographic Information System; Mobility; Weather Simulation

1. INTRODUCTION

Multiagent systems have inspired an increasing number of researchers from different domains. Their goal is to model the real world in terms of autonomous agents that can purposely interact with their external environment [13]. An agent can basically gather information from the environment using sensors, while attempting to execute its objectives using effectors [12]. In this paper, we propose a multiagent simulator that simulates traffic in an urban area subject to particular weather conditions. An inherent feature of such complex system is its multilayered structure and the nested levels of detail that compose it. Hence, we will model our simulator in terms of multiple independent layers. Each layer processes a particular aspect of the simulation through the interactions of its elements. A layer will therefore be represented by a complex network of interacting agents that communicate within that layer and possibly with other layers. Multiagent traffic simulation has been extensively studied since traffic and transportation management require autonomic, collaborative, and reactive agents [1]. It is within this perspective that we propose to simulate traffic while adopting a multilayered multiagent architecture coupled to behavioral agent simulation. This is in fact a way to refine both microscopic and macroscopic aspects found in many traffic simulators. Additionally, the availability of data collected via sensors and mobile devices allows us to better model human behaviors. For instance, this allows for the analysis of driver behaviors and the underlying decision-making mechanisms [3, 9, 10, 11]. As a result, we can build large-scale simulations by embedding the social interactions and by elaborating fine-grained human behaviors [7]. The resulting multiagent social simulation [2] is well suited to any social context due to its ability to simulate pro-active behaviors, parallel computations [8], and dynamic micro scenarios [6, 4]. Our motivation is to build a simulator that can emulate traffic as well as any environmental factor that effects traffic flow, routing, and even CO_2 emission. Being able to simulate weather is a novel way to approach traffic simulation since it allows the reproduction of real-world scenarios like traffic congestion in natural disaster situations. Such simulator becomes a testbed for general-purpose computational intelligence and can be used for the benchmarking of routing algorithms.

In section 2, we provide the architecture of the simulator. In section 3, we cover the behavioral models. Finally, we discuss the result.

Appears in: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016), J. Thangarajah, K. Tuyls, C. Jonker, S. Marsella (eds.), May 9–13, 2016, Singapore.*

2. SIMULATOR ARCHITECTURE

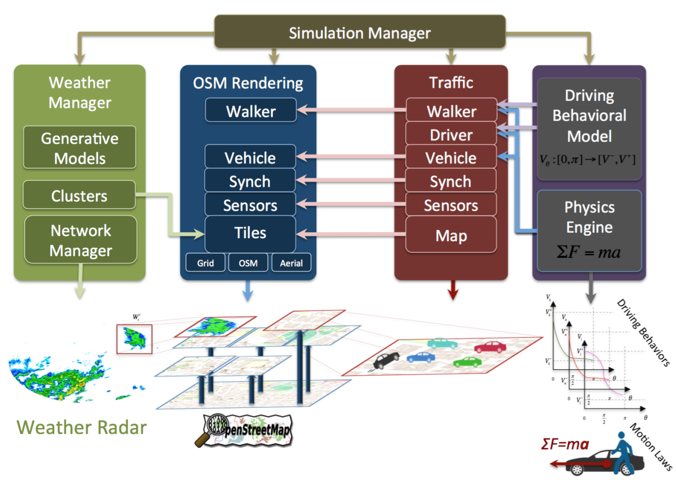

Figure 1: Architecture

The idea is to represent our complex system as a superposition of different layers. Each layer operates autonomously with its internal threads, agents, graphical objects, data, etc. The architecture of the simulator is illustrated in figure 1. We assume that the way vehicles navigate their space is not specific to one particular Geographic Information System (GIS). Particularly, There is an independence between the actual simulation (**Behavioral Models** and **Physical Engine**) and the corresponding GIS rendering. This is important when the renderer is complex, that is, when we are not only rendering vehicles, but weather data, pedestrians, snow, etc. This separation obeys the Dependency Inversion Principle (DIP) in the sense that the physical simulation acts as a high-level, abstract, or mathematical representation of the phenomena we are trying to simulate, while the GIS rendering is one possible way of rendering the traffic simulation. Next, we provide the components of the architecture.

2.1 Physics Engine and Behavioral Models

The **Physics Engine** component provides an approximate simulation of the physics that underly traffic, motion and the interactions between the agents (mostly collision detection). This component performs the simulation in real-time before rendering the result. The **Behavioral Models** component describe the scenarios that govern the motion of the vehicles and the behaviors of the drivers. The flow from the Physics Engine and Behavioral Models towards the OSM renderer is shown in figure 1 using the pink arrows.

2.2 Traffic Module

This module is composed of the 6 components. **Walkers** component relates to the pedestrians and their mobility generation and collision detection. **Drivers** contains routines related to driving, collision prediction, lights and lanes assessment. **Vehicles** is to modulate the acceleration, direction, location, friction, velocity, and breaking. **Synchronization** tier updates the states of the traffic lights. **Sensors** are the motion detectors and **Map** is the space where the motion is generated.

2.3 Weather Module

Weather simulation is the generation of precipitation as if it is detected by a Weather Surveillance Radar (WSR). Additionally, it is possible to download this data in real-time and render it directly onto the map.

2.4 Geographic Module

We adopt OpenStreetMap (OSM) [5] as a geographic referential. The rendering of the map relies on downloading and updating OSM tiles. Such tiles correspond to an area of the map and are loaded dynamically depending on where the simulation is being run. We note that there are two maps: a core map assigned to the physics engine, and a map to represent a real-world referential (OSM in our case). The core map is used to run the simulation in real-time so that the result is later rendered onto the second map (bottom pink arrow in figure 1).

3. BEHAVIORAL MODELING

The behavioral models govern the agents mobility and the actions allowed within the simulator. Herein, we mention two types of behavioral models: driving and vehicle mobility.

3.1 Driving

The main feature that reflects the driving behavior of a human is the velocity and the way it changes as function of the turns. In fact, turns are an important indicator of the driver's mastery of the steering wheel (with angle λ) and its physical effect on the vehicle. Thus, we can look at the turning angle θ comprised between the velocity and direction. For instance, driving in a straight line corresponds to $\theta = 0$ while a right turn corresponds to $\theta = \pi/2$. The minimal and maximal velocities of the vehicle are respectively $\vec{v}^-$ and $\vec{v}^+$, and define the driver's spectrum of physically allowed velocities. A behavioral model is a specific way of mapping θ to a velocity $\vec{v}_\theta$. Figure 2 shows three behavioral models. v_1 corresponds to what is perceived as a reckless driver since he barely decelerates when performing right turns. v_2 shows a conservative driver since he decelerates drastically when reaching right turns. v_3 shows a standard driver. A behavior is invariant beyond π (U-turn), which justifies the interval $[0, \pi]$.

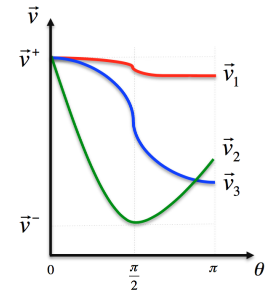

Figure 2: Behavioral Models $\vec{v}_1$, $\vec{v}_2$, and $\vec{v}_3$

3.2 Vehicle Mobility

Moving from point A to point B is reduced to a set of calculations that update the motive force $\vec{F}_m$, friction $\vec{F}_f$, acceleration $\vec{a}$, velocity $\vec{v}$, direction $\vec{d}$, and the location x. $\vec{d}$ is updated to $\alpha_{\vec{d}} + \frac{1}{L}(\|\vec{v}\| u_m) sin(\lambda \frac{\pi}{3}) p \frac{180}{\pi}$ with λ being the steering wheel angle, L the vehicle wheelbase, u_m the number of units per meter, p the simulation step, and $\alpha_{\vec{d}}$ is the angle of $\vec{d}$. $\vec{v}$ is updated to $\vec{d}\|\vec{v}\| + \vec{a}p$. Based on Newton's second law, $\vec{a}$ is set to $\frac{1}{m}(\vec{F}_m + \vec{F}_f)$. Updating all the forces requires updating the friction force whenever the driver breaks according to $\vec{F}_f \leftarrow -\vec{d}\vec{a}_{max}$ with $\vec{a}_{max}$ being the maximal acceleration. x is updated to $x + \vec{v} u_m p$.

4. DISCUSSION

The user can macroscopically switch between the views (Aerial view, OpenStreetMap view, Grid view) of the simulation by altering the opacity of the layer. It is possible to configure the simulations and specify the desired scenario by providing three XML files. The first file specifies the simulation geographic area. The second file specifies the driving behaviors of each driver by assigning a velocity to a turning angle λ. The third file specifies for each vehicle the starting point, the itinerary, and the driver's type. The separation between the layers allows the simulations to be scalable in the number of vehicles, despite the complexity of their behaviors. Furthermore, running the physical calculations within one monolithic component allows us to render all the results faster than if each layer had to separately perform complex physical computations. The simulator can additionally generate rich datasets corresponding to the vehicles physical processes (location, velocity, predictions), the drivers actions, and the weather effects of the overall traffic simulation.

REFERENCES

[1] B. Chen and H. Cheng. A review of the applications of agent technology in traffic and transportation systems. In *IEEE Transactions on Intelligent Transportation Systems*, pages 485 – 497, 2010.

[2] P. Davidsson. *Multi agent based simulation: beyond social simulation*. Springer, Berlin, Heidelberg, 2001.

[3] H. Dia. An agent-based approach to modelling driver route choice behaviour under the influence of real-time information. *Transportation Research Part C: Emerging Technologies*, 10(5-6):331–349, 2002.

[4] N. Gilbert. *Computer Simulation of Social Processes*. Number 6. Social Research Update, Department of Sociology, University of Surrey, UK, 1994.

[5] M. M. Haklay and P. Weber. Openstreetmap: User-generated street maps. *IEEE Pervasive Computing*, 7(4):12–18, Oct. 2008.

[6] A. Harding. *Microsimulation and Public Policy*. Elsevier, 1996.

[7] H. Hattori, Y. Nakajima, and S. Yamane. Massive multiagent-based urban traffic simulation with fine-grained behavior models. 2011.

[8] J. Misra. *Distributed Discrete-Event Simulation*, volume 18. ACM Computing Surveys (CSUR), 1986.

[9] K. Nagel and F. Marchal. Computational methods for multi-agent simulations of travel behaviour. In *Proceedings of International Association for Travel Behavior Research (IATBR)*, Lucerne, Switzerland, 2003.

[10] R. J. Rossetti, R. H. Bordini, A. L. Bazzan, S. Bampi, R. Liu, and D. Van Vliet. Using bdi agents to improve driver modelling in a commuter scenario. *Transportation Research Part C: Emerging Technologies*, 10(5-6):373–398, 2002.

[11] R. J. Rossetti, R. Liu, H. B. Cybis, and S. Bampi. A multi-agent demand model. In *Proceedings of the 13th Mini-Euro Conference and The 9th Meeting of the Euro Working Group Transportation*, pages 193–198, Bari, Italy, 2002.

[12] S. Russell and P. Norvig. *Artificial intelligence: a modern approach*. 1995.

[13] M. Wooldridge and N. R. Jennings. Intelligent agents: Theory and practice. *The knowledge engineering review*, 10(02):115–152, 1995.