SEMSim: A Distributed Architecture for Multi-scale Traffic Simulation

Yadong Xu
School of Computer Engineering
Nanyang Technological University
Singapore, 639798
Email: xuya0006@ntu.edu.sg

Heiko Aydt
TUM CREATE Ltd.
Singapore 637459
Email: heiko.aydt@tum-create.edu.sg

Michael Lees
School of Computer Engineering
Nanyang Technological University
Singapore, 639798
Email: mhlees@ntu.edu.sg

I. Introduction

With the fast urbanization of our modern society, transportation systems in cities are facing increasing problems such as congestion, collisions, and high levels of emissions. Researchers have been searching for solutions by investigating better urban planning and transportation policies, introducing new technologies such as Intelligent Transportation System (ITS), or introducing more environmentally friendly vehicles such as electric vehicles (EVs). Traffic modeling and simulation is one tool adopted by researchers for more than half a century [1] to help authorities assess new infrastructure design, and new policies without impacting real traffic.

Traffic simulation is typically classified according to the level of detail at which they represent traffic flow. These are macroscopic [2], mesoscopic [3], and microscopic [4]. In macroscopic simulation, traffic flow is modeled using fluid dynamics [5], [6] which analyzes the aggregate properties: speed, density, and flow, with respect to time and space. It is typically used for short-term traffic prediction and traffic control policy evaluation. Mesoscopic simulation considers a more detailed level of abstraction than macroscopic simulation. It models a combination of flow dynamics and simplified movement of vehicles [7]. Speed-density relationship and queuing theory are commonly used [3]. Mesoscopic simulation is often used for dynamic traffic demand modeling and real-time route planning. At the microscopic level, the detailed movement of each individual driver-vehicle-unit (DVU) in the street is modeled, characterized by car-following, lane-changing, and gap-acceptance behaviors [7]. Traffic flow patterns such as shock waves [6] are emergent phenomena from the microscopic behavioral descriptions. However, detailed microscopic models require large amount of calibration data and vast computational resources, which makes large-scale microscopic traffic simulation challenging [8]. Therefore, microscopic traffic simulation is suitable for offline evaluation of systems which impact individuals, such as ITS, for example. Cellular automata [9] is one type of microscopic model that describes the movement of DVUs in a discrete, cellular space, with basic localized rules instead of complicated mathematical equations. While being microscopic, the simplicity of the rules and assumptions about the space make this approach less computationally intensive.

Besides the aforementioned three approaches, researchers have recently extended traffic modeling and simulation into, what is termed, the nanoscopic level [10]. Instead of expressing the driver and the vehicle as a compound DVU, as in microscopic simulation, nanoscopic traffic models explicitly define the driver and the vehicle as unique entities. By modeling the traffic at this level of detail, questions regarding the effect of low-level components or behaviors on the overall traffic can be analyzed, such as the influence of risky driving, or EVs with different battery capacities, for instance.

Developing a simulation engine which attempts to capture this level of dynamics inevitably incurs high computational cost. If not done carefully, increasing the level of detail through nanoscopic modeling will only exacerbate this problem. In order to solve the performance issue of large-scale microscopic traffic simulation, work has been done from two main approaches: model construction and implementation. In terms of model construction, researchers built multi-model or hybrid traffic simulation aimed at capturing necessary details while omitting unnecessary details in traffic flow models [11]. Regarding implementation, work has generally focused on the use of parallel or distributed traffic simulation as a means of utilizing more computational power. One of the early works in this area is the parallel microscopic simulator PARAMICS [12]. Later, microscopic traffic simulator AIMSUN [13] and TRANSIMS [14] were also parallelized to increase the simulation speed. Discrete-event simulation has also been identified as an alternative to the traditional time-stepped method [4]. The simulation Dynameq is one excellent example of this [15].

The integration of electric vehicles into the traditional transportation landscape requires a great deal of effort and investment, whether it is to understand particular car design requirements or analyze the necessary infrastructure to support large numbers of electric vehicles. TUM CREATE Center for ElectroMobility Singapore[1] is a research program aimed at future transportation solutions for e-mobility in Singapore. At present, simulation is the only way for helping policy makers and car designers to make these very expensive decisions. In this paper we introduce the design of SEMSim (Scalable Electromobility Simulation), a nanoscopic traffic simulator

[1]http://www.tum-create.com.sg

designed for scalability and rapid part evaluation. We plan to utilize this simulation framework for helping understand the impact of various car and infrastructure design questions, and importantly the interaction between these two areas. For example, understanding how battery capacity, charging station placement and charging behavior impact the overall stability of the transportation system. To address issues regarding the performance of large-scale nanoscopic traffic simulation, SEMSim is designed to be a parallel discrete-event simulation capable of multi-resolution traffic simulation. In the remainder of this paper we outline the design of the simulation engine, in particular the discrete-event architecture, and also discuss issues regarding parallelization.

II. Discrete-Event Architecture

The main entity in nanoscopic traffic simulation is the *nano-agent*, which consists of a driver and a vehicle. The driver is realized by various model components that characterize the driver behavior. For example, behavioral components may include models for car-following and lane-changing. Similarly, the vehicle is realized by model components that characterize various vehicle parts. For example, in the context of electromobility, it is crucial to simulate the battery and major energy consuming parts, such as the air conditioning unit. In our model, a vehicle would thus be an orchestration of vehicle model components and, likewise, a driver would be an orchestration of driver behavior model components.

Microscopic and nanoscopic traffic simulation is typically realized in a time-stepped fashion. However, this has the disadvantage of deciding on a suitable time resolution. This is particularly difficult when considering nanoscopic models like those considered here. The various behavior and vehicle model components may necessitate arbitrary time scales. For this reason we consider a discrete-event approach. In general, each of the various model components are event generators and a nano-agent is thus a composition of event generators. A logical process (LP) may be responsible for executing and arbitrary number of agents. Events generated within an LP are organized in a single event queue. We distinguish different types of state variables, depending on the scope of visibility:

- Agent-based state variables: these variables represent states that are associated with an agent rather than a specific model component.
- Component-based state variables: these variables represent states that are specific to a particular model component. For example, a battery component will have a state of charge variable which indicates the amount of energy left.

Depending on the model components' functionality, they may need to be notified whenever a particular state variable is changing. For example, a routing behavior component may decide to re-calculate the route in case the battery charge drops below a certain threshold. For this purpose, it is necessary that the routing behavior component gets notified whenever the charging state of the battery component is changing. We

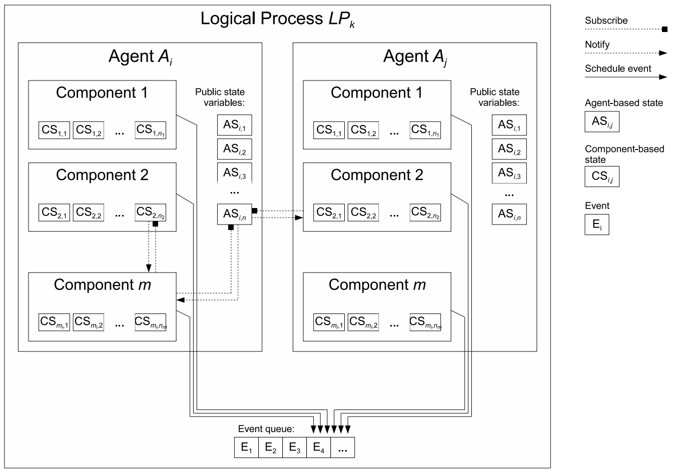

Fig. 1. SEMSim LP Architecture: an LP may contain an arbitrary number of agents. An agent is realized by an orchestration of model components for driver behaviors and vehicle components. Any model component can subscribe to change notifications published by state variables. In addition, every model component is an event generator. Each LP has its own event queue.

realize this notification functionality by using a publisher-subscriber pattern: state variables are state change publishers and model components can subscribe to state change notifications. It is important that an agent has access to some of the state variables of other agents. For example, in order to determine the preferred speed, a corresponding driving behavior component needs to be able to access the positions of other agents in its proximity.

Figure 1 illustrates the architecture of an LP in SEMSim.

III. Parallel and Distributed Simulation

Simulating a city-scale scenario with a realistic number of agents is computationally intensive. For example, in order to simulate the entire Singapore traffic network, it is necessary to consider many tens of thousands of vehicles and their drivers. Each of the DVUs comprises of a number of model components (or sub-models). Without parallelization and/or distribution of the simulation it would take too long to execute simulation scenarios in a reasonable period of time. An important issue for parallelization and distribution of our discrete-event simulation engine is the partitioning of agents into agent clusters and the mapping of agent clusters to LPs.

Inter-dependencies between agents and their components can cause a significant synchronization overhead. These dependencies are a result of two levels of causality in SEMSim: firstly, the state change subscriptions concerning the position of agents, and secondly, the causality between various events generated by each model component. Dependencies between agents can be assumed to be more prevalent if they are within each other's proximity. This is due to the fact that agents that are close to each other, within the traffic network, need to consider each other's location and velocity in order to determine their driving (e.g., preferred speed) strategy. The hierarchical structure of the road network can be assumed

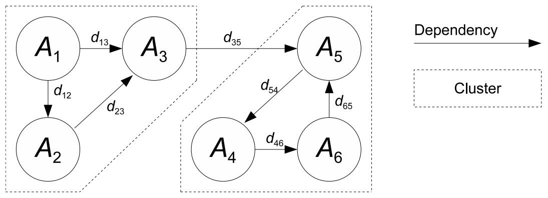

Fig. 2. Agent clusters and dependencies. A cluster may contain an arbitrary number of agents. Dependencies arise from state notification subscriptions between agents from different clusters.

to have significant impact on the inter-agent dependencies. For example, highways are usually connected to minor roads via ramps. Therefore, in order to determine the proximity of agents, it is necessary to take structural properties of the road network into consideration. As a result, agents which are close-by may not be within each other's proximity and thus not inter-dependent. Clustering of agents is illustrated in Figure 2.

In order to perform a parallel simulation efficiently, it is necessary to minimize the dependencies between various LPs. This can be achieved by optimal allocation of agents to clusters. We consider the cluster allocation vector $\mathbf{a}$:

$$\mathbf{a} = [a_1, a_2, a_3, \dots, a_M] \tag{1}$$

where M the total number of agents, N is the total number of clusters, and $1 \leq a_i \leq N$ indicates that the i-th agent is allocated to cluster a_i. The optimization problem can be described as a combinatorial problem that aims to determine the optimal $\mathbf{a}$ for which the total degree of dependencies D is minimized:

$$D = \sum_{i=1}^{N} \sum_{j=i+1}^{N} D(i,j) \tag{2}$$

where $D(i,j)$ is the number of dependencies between clusters i and j. The number of dependencies between two clusters can be determined by counting the number of state change subscriptions between the agents of cluster i and the agents of cluster j. However, minimizing dependencies is not sufficient. Load balancing is another issue that needs to be considered. Each LP should have approximately the same amount of work load. This can be done by considering another objective which aims to minimize the inequality L in load allocation:

$$L = \sum_{i=1}^{N} |S_i - \overline{S}| \tag{3}$$

where S_i is the size of cluster i (in terms of number of agents) and $\overline{S}$ is the average size of all clusters.

This multi-objective optimization problem is generally hard to solve for the large amount of agents which we will have to consider for a city-scale simulation. However, what makes this problem even more difficult is the dynamic nature of traffic simulation. The positions of the agents are constantly changing and thus requiring dynamic re-calculation of the allocation of agents. Apart from determining the optimal allocation of agents, another issue is thus the frequency of performing a re-calculation.

IV. Summary and Future Work

City-scale nanoscopic traffic simulation is a challenging problem that requires parallelization and distribution. In this paper, we have given an overview of the architecture for our nanoscopic traffic simulator SEMSim. For efficient parallel simulation, reducing the dependencies between the various LPs is crucial. We have specified a multi-objective optimization problem concerned with the allocation of agents to clusters. In our future work, we will implement a nanoscopic traffic simulation and devise methods to solve this problem dynamically. Given the difficulty of the problem, these methods will have to make use of domain-specific knowledge, such as information about the topology of the road network.

References

[1] M. Pursula, "Simulation of Traffic Systems - An Overview," *Journal of Geographic Information and Decision Analysis*, vol. 3, no. 1, pp. 1–8, 1999.

[2] M. Papageorgiou, I. Papamichail, A. Messmer, and Y. Wang, "Traffic Simulation with METANET," in *Fundamentals of Traffic Simulation*, ser. International Series in Operations Research & Management Science, J. Barceló, Ed. New York, NY: Springer New York, 2010, vol. 145, pp. 399–430.

[3] M. Ben-Akiva, H. N. Koutsopoulos, C. Antoniou, and R. Balakrishna, "Traffic simulation with dynamit," in *Fundamentals of Traffic Simulation*, ser. International Series in Operations Research & Management Science, J. Barceló, Ed. Springer New York, 2010, vol. 145, pp. 363–398.

[4] M. Ben-Akiva, H. N. Koutsopoulos, T. Toledo, Q. Yang, C. F. Choudhury, C. Antoniou, R. Balakrishna, and J. Barceló, "Traffic Simulation with MITSIMLab," in *Fundamentals of Traffic Simulation*, ser. International Series in Operations Research & Management Science, J. Barceló, Ed. New York, NY: Springer New York, 2010, vol. 145, pp. 233–268.

[5] M. J. Lighthill and G. B. Whitham, "On Kinematic Waves. II. A Theory of Traffic Flow on Long Crowded Roads," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 229, no. 1178, pp. 317–345, May 1955.

[6] P. I. Richards, "Shock Waves on the Highway," *Operations Research*, vol. 4, no. 1, pp. 42–51, 1956.

[7] J. Barceló, "Models, Traffic Models, Simulation, and Traffic Simulation," in *Fundamentals of Traffic Simulation*, ser. International Series in Operations Research & Management Science, J. Barceló, Ed. New York, NY: Springer New York, 2010, vol. 145, pp. 1–62.

[8] Y. Hollander and R. Liu, "The principles of calibrating traffic microsimulation models," *Transportation*, vol. 35, no. 3, pp. 347–362, Jan. 2008.

[9] K. Nagel, M. Schreckenberg, and Others, "A cellular automaton model for freeway traffic," *Journal de Physique I*, vol. 2, no. 12, pp. 2221–2229, 1992.

[10] D. Ni, "2DSIM: a prototype of nanoscopic traffic simulation," in *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE.* IEEE, 2003, pp. 47–52.

[11] R. Claes and T. Holvoet, "Multi-model traffic microsimulations," in *Proceedings of the 2009 Winter Simulation Conference (WSC).* Ieee, Dec. 2009, pp. 1113–1123.

[12] G. Cameron, "PARAMICS–Parallel Microscopic Simulation of Road Traffic," *The Journal of Supercomputing*, vol. 53, pp. 25–53, 1996.

[13] J. Barceló, J. L. Ferrer, and D. Garcia, "Microscopic traffic simulation for ATT systems analysis. a parallel computing version," *25th Aniversary of CRT*, pp. 1–16, 1998.

[14] K. Nagel and M. Rickert, "Parallel implementation of the TRANSIMS," *Parallel Computing*, vol. 27, pp. 1611–1639, 2001.

[15] M. Mahut and M. Florian, "Traffic simulation with dynameq," in *Fundamentals of Traffic Simulation*, ser. International Series in Operations Research & Management Science, J. Barceló, Ed. Springer New York, 2010, vol. 145, pp. 323–361.