Multi-model Based Simulation Platform for Urban Traffic Simulation

Yuu Nakajima, Shohei Yamane, and Hiromitsu Hattori

Department of Social Informatics, Kyoto University
Yoshida-Honmachi, Sakyo-ku, Kyoto, 606-8501, Japan
{nkjm,hatto}@i.kyoto-u.ac.jp
yamane@ai.soc.i.kyoto-u.ac.jp

Abstract. Multiagent-based simulations are regarded as a useful technology for analyzing complex social systems; for example, traffic in a city. Traffic in a city has various aspects such as route planning on the road network and driving operations on a certain road. Both types of human behavior are being studied separately by specialists in their respective domains. We believe that traffic simulation platforms should integrate the various paradigms underlying agent decision making and the target environment. We focus on urban traffic as the target problem and attempt to realize a multiagent simulation platform based on the multi-model approach. While traffic flow simulations using simple agents are popular in the traffic domain, it has been recognized that driving behavior simulations with sophisticated agents are also beneficial. However, there is no software platform that can integrate traffic simulators dealing with different aspects of urban traffic. In this paper, we propose a traffic simulation platform that can execute citywide traffic simulations that take account of the aspects of route selection on a road network and driving behavior on individual roads. The proposed simulation platform enables the multiple aspects of city traffic to be reproduced while still retaining scalability.

1 Introduction

Multiagent-based simulations are increasingly seen as the most attractive approach to reproducing and analyzing diverse social systems including autonomous and heterogeneous decision making entities, *i.e.*, humans [5]. The multiagent-based simulation is a paradigm that can reproduce macroscopic complex phenomena through localized interactions among heterogeneous agents. Multiagent-based simulations have been applied in various fields in the city, examples include traffic planning, rescues, and pandemic responses[1,8,4]. Although numerous attempts have been made to conduct multiagent-based simulations in various domains, no study has fully captured and analyzed social systems from various aspects.

The challenge tackled in this paper is a massive urban traffic simulation platform based on the multi-model approach to agent decision making and the target

N. Desai, A. Liu, and M. Winikoff (Eds.): PRIMA 2010, LNAI 7057, pp. 228–241, 2012.

environment in a city. Traffic, which is one of the most complex systems in modern society, is a highly suitable target for our research because vehicular traffic is a phenomenon that includes various aspects: route selection and driving behavior.

While traffic flow simulations using simple agents are popular in the traffic domain, it has been recognized that driving behavior simulations with sophisticated agents provide many additional benefits for analyzing the relation between local driving behaviors and global traffic flow in a city. However, no published software platform can integrate traffic simulators dealing with different aspects of urban traffic. We design an architecture and develop a framework to integrate multiple simulators founded on different paradigms. The proposed platform provides a collaborative environment to experts who traditionally use different simulators in different domains. We also propose a traffic simulation platform that can execute citywide traffic simulations that include the aspects of route selection aspect and driving behavior.

More specifically, this paper has three goals:

1. Design multiagent simulation platform based on multi-model of a city
   The phenomena that occur in a city cannot be captured with a single model. For realizing traffic simulations of a whole city, it is required that the platform enable us to integrate different aspects of agent decision making and the target environment in the relevant areas. It is also necessary that the platform take advantage of multiple models proposed in different works.
2. Implement urban traffic simulation capturing various aspects of a city and agents
   We develop an urban traffic simulator based on the proposed architecture. In this platform, we focus on two aspects of the traffic domain in a city: global route processing and local driving behavior.
3. Evaluate platform performance
   We investigate the potential of the developed platform to realize more realistic urban traffic simulations. We verify that the simulation platform enables the introduction of multiple aspects of traffic while still retaining scalability. We also conduct an experiment that demonstrates how the number of agents impacts simulation results such as traffic flow.

The remainder of this paper is as follows. Section 2 describes our approach to designing the multi-model traffic simulator platform. Section 3 shows the implementation of the platform. Section 4 describe an analysis of the platform's performance and Section 5 demonstrates the effects of the number of agents.

2 Architecture

We consider that agents in the traffic simulation should be covered by flexible combinations of various decision-making models. This is because agents face various situations and make decisions according to their current situations while they move around the city. In addition, the simulation has to include traffic

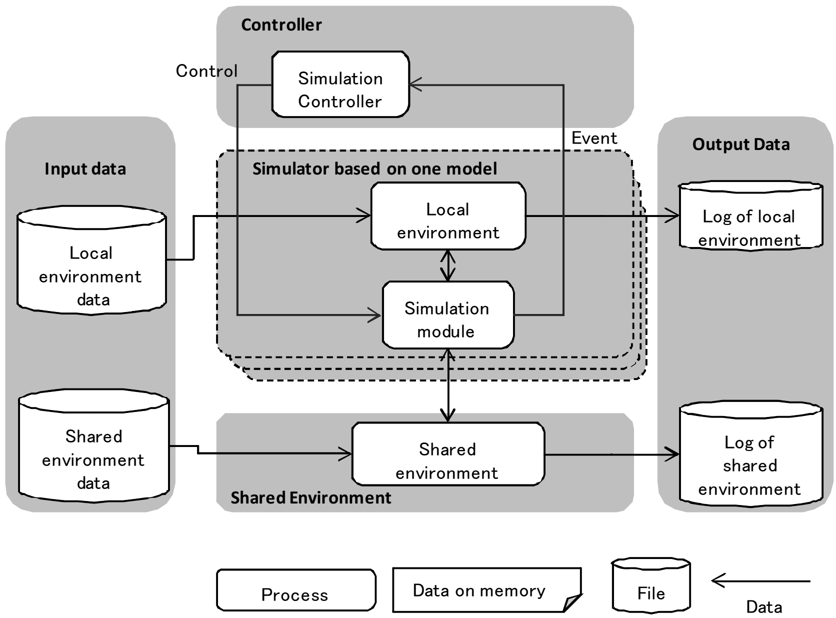

Fig. 1. Architecture for Multi-model Simulation Platform

systems such as traffic control systems and car navigation systems. The platform must integrate various aspects of the city environment.

Figure 1 shows the architecture proposed in this paper. This architecture includes multiple simulators and each simulator captures a specialized aspect of the traffic domain (*e.g.* route selection aspect and driving behavior aspect).

Settings unique to the environment covered by each simulator and the environment settings shared by the simulators are input. When the result of a certain simulator influences another simulator, the result is stored in the shared environment. On the other hand, information that is unique to one simulator cannot be accessed by other simulators. Such data is accumulated in the corresponding local environment.

Simulation controller should manage the simulation processes in order to combine the multiple simulators. The controller requests simulators to calculate the state of the next step. Basically, the simulators receive a request to output a result for the next time step. When an event that should be sent to another simulator occurs in the calculation, the event is sent to that simulator through the simulation controller.

When all simulations finish, the logs of local environments and the logs of the shared environment are written to external files.

Some platforms that combine multiple simulators have been proposed, but these platforms mainly focus on use in a distributed environment [11].

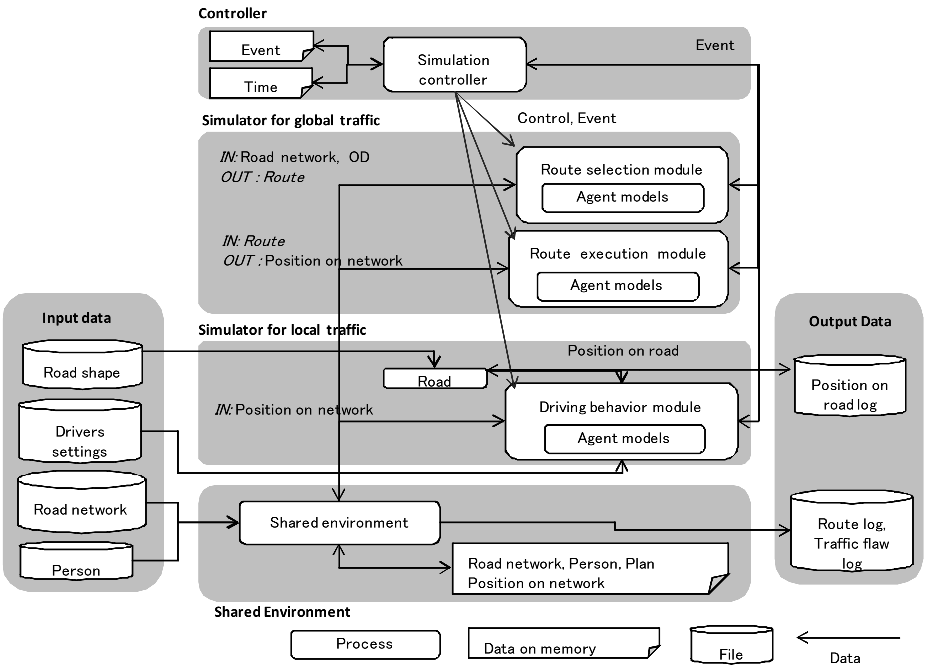

Fig. 2. System Diagram of Platform

3 Implementation

Previous traffic simulation research consists of either route selection on a road network or local driving behavior on single roads. Research on route selection has lead to the modeling of decision processes and route utility functions. Research on local driving behavior has considered the observation of and responses to road geometry, signals, and surrounding cars. There are gaps between the global traffic flow based on route selection and local traffic flows based on driving behavior.

Nagel and his colleagues worked on global traffic flow in a city with multiagent-based traffic simulators based on the queue model [1]. However, their approach fails to support realistic driving behavior simulations on particular roads. This is because details of the road structure (*e.g.*, the width of lanes) or surrounding environment including neighboring vehicles cannot be represented, so that the simulated driving behavior fails to consider such local factors.

We assume that there is some interaction between local driving behavior and global route selection. What we need to do is to analyze how local driving behavior impacts citywide traffic patterns. Therefore, the simulation platform must be able to incorporate both driving behavior models and route selection models.

We implemented a traffic simulation platform on the proposed architecture. Figure 2 depicts the system diagram of the platform. We used the open source traffic simulation tool kit MATSim[1] to create the platform. We select MATSim because it has been applied to various traffic simulations and its source codes are completely open [7,2,3]. The global traffic simulation part of our platform is mainly owe to MATSim.

In the following sections, each module is described precisely.

3.1 Simulator for Global Traffic

Route Selection Module. The route selection module reads road network data and OD (Origin-Destination) data of agents from the shared environment. Road network data mainly describes the structure of the road network while the OD data consists of tuples of the starting point and the destination point of each agent.

The road network has travel times of each link; we use either initial default values or the results of the traffic flow simulation of the previous day. The route selection module calculates the average trip time of each road based on the traffic information of the previous day.

In the route selection module, an agent is regarded as the entity performing route selection. The agent selects the route that has minimum cost considering map information and the average trip time of each road. A route plan consists of paths, mode choice, daily activity, and so on.

This module outputs the routes selected by the agents to the shared environment.

Route Execution Module. The route execution module deals with abstracted road networks, not two-dimensional spaces. The route execution module is implemented for handling a queue-based simulator; that is, the road network is represented as a network of FIFO (First-In, First-Out) queues. Each agent moves over this queue-network between queues according to its scheduled routing plan given vacancies in the next queue. Traffic flows in this platform are composed of agent transfers between queues.

The route execution module reads the route plan of each driver agent from the shared environment. In the route execution module, the agent is regarded as the plan executor.

The road network is abstracted as a network consisting of nodes and links. The agent acquires location information on the basis of nodes and links. A road node pops a driver agent from the waiting queue and pushes it onto the running queue of the next road link, if the running queue on the next road link has enough space.

The route execution module writes agent positions, using node and link descriptions, to the shared environment.

[1] MATSim (Multiagent Transport Simulation Toolkit: (`http://sourceforge.net/projects/matsim/`) is an open source toolkit developed by the Technical University Berlin and the Swiss Federal Institute of Technology Zurich for conducting large-scale agent-based traffic simulations. Revision 7476 is used in this paper.

3.2 Simulator for Local Traffic

Driving Behavior Module. In order to achieve traffic simulations that cover the driving behavior level, we add a driving behavior module. In the driving behavior module, the agent is regarded as a virtual driver and vehicle. They move in a two-dimensional space rather than the abstract road network.

The driving behavior module starts calculating driving behavior when an agent enters a link in the route execution module. The module reads agent ID and road ID from the shared environment and gets details of the road's structure and surrounding environment including neighboring vehicles from the road module in the local environment.

Data that is used by only one simulator must be accumulated in the local environment for the simulator. Other simulators do not use specific road details such as width and slope, but deal with more abstract data such as transit time or link loads. Accordingly, these elements are stored in the road module of the local environment.

The execution process of agents in the driving behavior module is summarized as follows.

1. Observation
   Controller requests the driving behavior module to determine the next operation. At first, the driver agent demands information on the surrounding environment, *i.e.*, sensor data. He observes state of own car, surrounding cars, and the roads in the immediate vicinity.
2. Recognition
   Drivers may not be able to recognize all observed information. This step filters the observed information based on the driver's characteristic. For example, an aged driver is unable to mentally map the surrounding traffic situation as quickly as a young driver.
3. Decision
   Driver agents decide which driving behavior should be executed next considering the recognized information. They determine their acceleration/brake/steering operations.
4. Execution
   The driver agents execute the acceleration/brake/steering operations. This involves not only setting the accelerator/brake/steering values directly but also the execution of sequential acts such as changing lanes. The driver agent has own vehicle module which holds car specifications, such as size, maximum speed, car type and so on. The vehicle module converts the operations set by the driver agent into direction and acceleration/deceleration values.
5. Update location
   Vehicle module calculates the vehicle's next state, such as its speed, velocity, and direction, based on the driving operation. Vehicle module updates the location information for the road module by accumulating the positions of vehicles in the local environment.

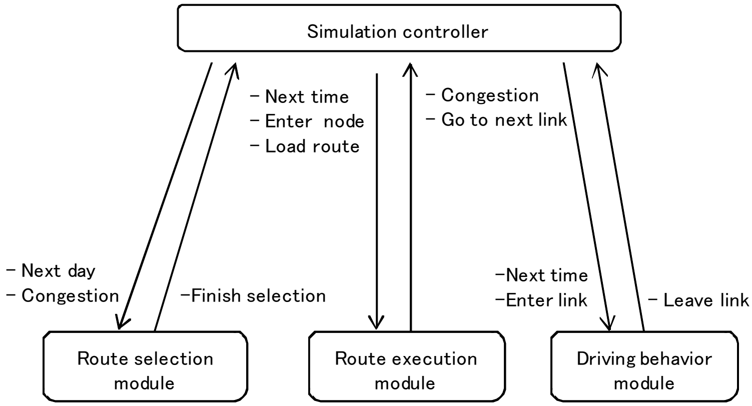

Fig. 3. Message Control Provided by Simulation Controller

Using the location information, the driving behavior module checks whether the driver agent should be transferred to the next link or not. The result is then reported to the route execution module via the simulation controller.

3.3 Simulation Controller

The simulation controller administers the entire simulation process. Simulator communication is based on message passing. At the beginning of a city traffic simulation, the route selection module is called to create a route from starting point to goal point for each agent. After that, the traffic simulation is started. The route execution module is called every second to calculate the route traces of agents on the abstracted road network. The driving behavior module can be called on shorter periods, such as 0.1 seconds.

Figure 3 shows how the simulators work together by sending messages.

- When a simulation is started, the controller requests the route selection module to calculate a route from starting point to goal point ("Next Day"). When congestion occurs on an intersection, the route selection module receives a "Congestion" message from the route execution module and rerouting is begun. The route selection module returns "Finish selection" message. After that, the controller sends "Load route" to the route execution module which triggers the module into reloading the appropriate routes.
- When the route execution module receives "Next time" message, the module calculates the state expected at the next time step. If the route execution module receives "Enter node" message which is raised by the "Leave link" message sent by the driving behavior module, the route execution module registers the agent mentioned in the message as an object to calculate the route trace of the agent on the road network. The agents on the route execution module decide the next link toward their goals and send "Go to next link" messages with agent ID and road ID to the simulation controller.

- When the driving behavior module receives "Next time" message, the module calculates the state expected at the next time step. If the driving behavior module receives "Enter link" message which is raised from "Go to next link" from the route execution module, the driving behavior module registers the agent mentioned in the message as an object to calculate its driving behavior. The driver agents in the driving behavior module check whether they have reached the end of the link or not. If they have arrived at the end of the link, "Leave link" messages are sent to the route execution module via the simulation controller.

In this manner, our platform for traffic simulations can integrate the simulators that reflect different aspects of driving in a city, *i.e.*, global route planning-execution and local reactive behavior.

3.4 Shared Environment

The shared environment manages data shared by agents on different simulators. This technique allows transitions in the data to be handled. At the step of time t, all agents read data at time t and decide actions for time t. At the end of the step, the shared environment fixes the data for time $t+1$. In doing so, the simulators do not need to consider the order in which agents are processed.

In general, several simulators may access the shared environment simultaneously, so we need to implement the lock and rollback functions for the shared environment. At present, the shared environment does not have facilities for lock and rollback because these agents on the simulator do not write to the shared environment simultaneously and so do not cause conflicts in terms of the results of actions in our traffic simulation[2]. If the actions of the agents cause a conflict, for example the agents intend to occupy the same spatial position at the same time step in the driving behavior module; the shared environment rollbacks the data and requests the agents to recalculate. With the conflict in mind, they recast their operations at time t all over again.

When other simulation modules are added in this platform, the simulation modules have to implement the interfaces that support event control and data sharing, which are defined by the simulation controller and the shared environment.

4 Performance Analyses

It is important to achieve adequate scalability as well as the ability to handle multiple aspects of traffic. This is because traffic is a phenomenon that emerges from the mass actions of agents.

[2] In the driving behavior module, driver agents can recognize surrounding agents and they move only a short distance from one time step to the next because the time offset is small. Therefore, agents should not collide with each other.

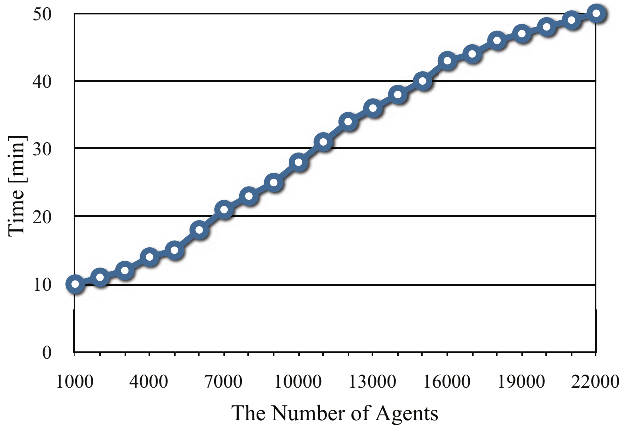

Fig. 4. Computation Time

For example, Paruchuri *et al.* reproduced some traffic situations with around 30 vehicles [9]. Increasing the number of vehicles yielded different results. Agent-based auction simulations were executed in [12], this research indicated that the simulation results were affected by the number of agents.

The challenge tackled in this paper is to achieve massive urban traffic simulations based on the multi-model simulator. In this section, we show that the implemented simulation platform has sufficient scalability. This is because there is a trade-off between the scale of multiagent-based simulations and the diversity of traffic models (decision making of agent and the target environment) in terms of the computation time.

In this experiment, we generated 100 ODs (origin-destination) by pairing two randomly selected points from 25 main intersections within an area that represents the heart of the city of Kyoto (2km x 2km square with 1700 links). For simplicity, all agents used the same route selection model and the same driving behavior model. The simulation time was 2 hours. We ran our experiments on a desktop computer with a Core2Duo 2.53 GHz CPU and 3GB of main memory.

Figure 4 plots the computation time versus the number of agents. As you can see, the computation time is directly proportional to the number of agents. In fact, with the largest number of agents (22,000), the computation time is around 50 minutes.

5 Effect of the Number of Agents

As shown above, we implemented a traffic simulation platform and in this section, we experimentally confirm that our platform has the ability to reproduce actual urban traffic created by a large number of agents. As an example, we investigated how the number of agents impacted city traffic.

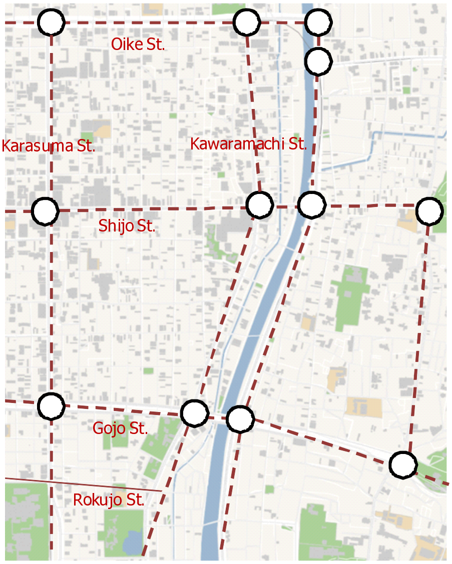

Fig. 5. Simulation Target Area

5.1 Settings

We conducted simulations with 8000 vehicle agents, each of which was assigned an OD selected from 36 types of ODs. We prepared an OD set considering two types of traffic, *i.e.*, traffic in the central part of the target area of the experiment and traffic through the central area. The simulation period was set to 90 minutes and the simulation was iterated 50 times following [10].

Figure 5 shows the simulation target area, which is the central part in the city of Kyoto. Circled points are big intersections. Agents mainly depart from and arrive at these big intersections. The dashed red lines are main streets. We applied the road network data, including all road links in Kyoto city, prepared for commercial programs. Figure 6 shows a screen shot of a simulation experiment. Red rectangles are simulated vehicles.

The aim of this experiment was to investigate how the number of agents influences global traffic flow via agents' route selection.

5.2 Execution

In this platform, multiagent-based urban traffic simulations are conducted with agents who can make decisions on both global route planning-execution and local driving operation. An agent has functions to interact with both simulation modules so that it can determine the most suitable route to the destination and run on that route while expressing its preferred driving behavior (accelerating, braking, lane-changing) given the surrounding environment. The agents decide their behavior according to the assigned models.

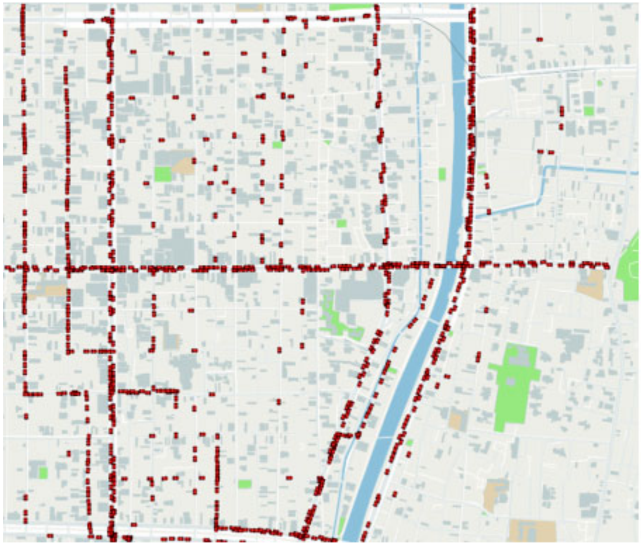

Fig. 6. Simulations of the Traffic in the heart of Kyoto city

Within a simulation, the agents iteratively execute the day-to-day re-planning process which consists of route-planning, traffic flow simulation, and scoring. The traffic flow simulation is calculated every second. The details of the process are as follows:

1. At the initial step, a set of initial plans (routes) is generated based on free speed travel times in the route selection module.
2. The traffic simulation is run using the generated plans in the route execution module and the driving behavior module.
3. Each agent calculates the score of his/her plan based on the performance identified by the simulation at end of the day in the route selection module.
4. In the route selection module, some of the population (10% is used in this paper) explore new plans based on the updated travel times resulting from the last simulation. The remaining agents use the previously executed plan.
5. Step 2 to step 4 must be iterated many times before the optimized demand can be identified.

In this paper, we iterated steps 2 to 4 over 50 days and the length of step 2 was 90 minutes.

5.3 Results

We investigate how the number of agents affects the outcome of the simulation, such as visible traffic flows. In order to analyze the effect of the number of agents, we changed only the number of agents; from 2,000 to 12,000 in steps of 2,000.

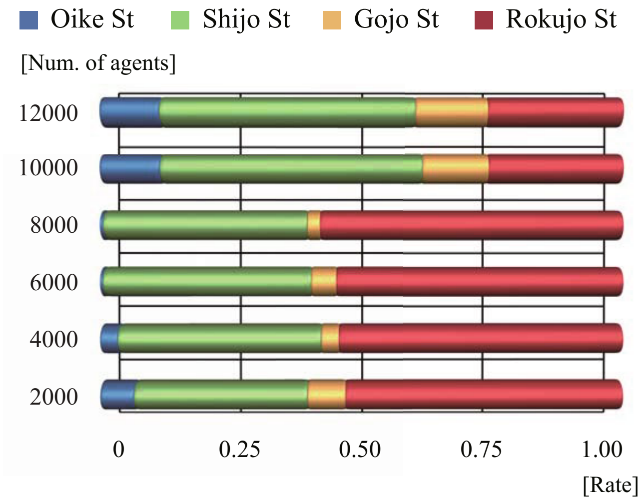

Fig. 7. Impact of the Number of Vehicle Agents: Traffic Share Rate of Four Streets

We counted the number of vehicles that drove through four streets (Oike St., Shijo St., Gojo St., and Rokujo St.) from Karasuma St. to Kawaramachi St. (accordingly, we did not count vehicles which changed their route in the middle of the streets). Rokujo St. is relatively narrow and the three other streets are main streets in the city.

Figure 7 shows the traffic share rates of these four streets in the result of simulation iteration 50. Starting from the left, each column lists the share rates of Oike St., Shijo St., Gojo St. and Rokujo St. Because Rokujo St. is rather a short route between Kawaramachi St. from Karasuma St. (see Figure 5), the share rate of Rokujo St. was high. As shown in the figure, this situation, traffic flows are biased to Rokujo St., is unchanged regardless of the number of agents. However, the share rate of Rokujo St. is reduced at agent numbers of 8000 and 1,0000, while the rates of Oike St., Shijo St and Gojo St. are increased. These results presumably mean that Rokujo St. becomes full and the agents avoid it by selecting other routes including the three other streets even though the routes are longer than routes through Rokujo St.

The important point is that these results are obtained by only changing of the number of the agents. These results indicate that traffic modality patterns do depend on simulation scale. Thanks to the scalability of our simulation platform, we can capture the effect of volume of agents on the city traffic.

6 Conclusion

Multiagent-based simulations yield multiagent societies that well reproduce human societies, and so are seen as an excellent tool for analyzing the real world.

Although numerous attempts have been made to conduct multiagent-based simulations in the traffic domain, it has, up to now, been impossible to reproduce and analyze the traffic from various aspects.

Existing research on city traffic falls into two camps; research focused on global route selection and research focused on local driving behavior. However, these two behaviors clearly affect each other. Phenomena that occur in a city cannot be captured with single model.

For realizing city-wide traffic simulations, the different aspects of agent decision making and the target environment must be integrated. Toward our objective, we developed a wide-area traffic simulation platform based on the multi-model approach that enables us to execute social simulations from various aspects of city traffic.

Our contributions are as follows.

1. Designed multiagent simulation platform based on multi-model approach
   For realizing city-wide traffic simulations, we designed a multi-model platform for urban traffic simulations that can take account of the different aspects of the decision making of agents and the target environment. The platform allows us to take advantage of the multiple models proposed in related works.
2. Implemented urban traffic simulation capturing various aspects
   We developed an urban traffic simulator based on the proposed architecture. This integrated simulator includes two models; route processing and driving behavior.
3. Evaluate platform performance
   We evaluated the scalability of the platform. As a simulation example, we examined how the number of agents impacts simulation results such as traffic flow.

One future direction of this study is to create more sophisticated behavior models. It is clear that human drivers have very diverse driving behaviors with complicated decision making processes. We are going to use participatory modeling methodologies to extract more realistic driving behavior models [6].

Acknowledgment. This work was supported by Panasonic Corp. - Kyoto University Joint Research: Crowd Navigation for Region EMS Considering Individual Behaviors and Preferences and Kyoto University Global COE Program: Informatics Education and Research Center for Knowledge-Circulating Society.

References

1. Balmer, M., Cetin, N., Nagel, K., Raney, B.: Towards truly agent-based traffic and mobility simulations. In: The 3rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS-2004), pp. 60–67 (2004)
2. Balmer, M., Meister, K., Rieser, M., Nagel, K., Axhausen, K.W.: Agent-based simulation of travel demand: Structure and computational performance of matsimt. In: The 2nd TRB Conference on Innovations in Travel Modeling (2008)

3. Balmer, M., Rieser, M., Meister, K., Charypar, D., Lefebvre, N., Nagel, K.: MATSim-T: Architecture and Simulation Times. In: Multi-Agent Systems for Traffic and Transportation Engineering, pp. 57–78. IGI Global (2009)
4. Deguchi, H., Kanatani, Y., Kaneda, T., Koyama, Y., Ichikawa, M., Tanuma, H.: Social simulation design for pandemic protection. In: The First World Congress on Social Simulation (WCSS-2006), vol. 1, pp. 21–28 (2006)
5. Epstein, J., Axtell, R.: Growing Artificial Societies: Social Science from the Bottom Up. MIT Press (1996)
6. Hattori, H., Nakajima, Y., Ishida, T.: Learning from humans: Agent modeling with individual human behaviors. IEEE Transactions on Systems, Man, and Cybernetics, Part A 41(1), 1–9 (2011)
7. Illenberger, J., Flotterod, G., Nagel, K.: Enhancing matsim with capabilities of within-day re-planning. In: The IEEE Intelligent Transportation Systems Conference, pp. 94–99 (2007)
8. Kitano, H., Tadokor, S., Noda, H., Matsubara, I., Takahasi, T., Shinjou, A., Shimada, S.: Robocup rescue: search and rescue in large-scale disasters as a domain for autonomous agents research. In: The IEEE Conference on Systems, Men, and Cybernetics, Tokyo, vol. VI, pp. 739–743 (October 1999), `citeseer.ist.psu.edu/kitano99robocup.html`
9. Paruchuri, P., Pullalarevu, A.R., Karlapalem, K.: Multi agent simulation of unorganized traffic. In: The 1st International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2002), pp. 176–183 (2002)
10. Raney, B., Nagel, K.: Iterative route planning for large-scale modular transportation simulations. Future Generation Computer Systems 20(7), 1101–1118 (2004)
11. Scerri, D., Hickmott, S., Padgham, L., Drogoul, A.: An Architecture for Modular Distributed Simulation with Agent-Based Models. In: Ninth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2010), pp. 541–548 (2010)
12. Yamamoto, G., Tai, H., Mizuta, H.: A Platform for Massive Agent-Based Simulation and Its Evaluation. In: Jamali, N., Scerri, P., Sugawara, T. (eds.) MMAS 2006, LSMAS 2006, and CCMMS 2007. LNCS (LNAI), vol. 5043, pp. 1–12. Springer, Heidelberg (2008)