Proceedings of the 2007 IEEE
Intelligent Transportation Systems Conference
Seattle, WA, USA, Sept. 30 - Oct. 3, 2007

A Software Architecture for Artificial Transportation Systems – Principles and Framework

Jinyuan Li, Shuming Tang, *Member, IEEE*, Xiqin Wang, Fei-Yue Wang, *Fellow, IEEE*

Abstract—**Artificial Transportation Systems(ATS) is an extension of the traffic micro-simulation, which integrates the transportation system with other urban systems, such as logistic systems, social and economic systems, etc., to behave as a coordinated tool for transportation analysis, evaluation, decision-making, and training. Research, especially implemental work, on ATS is far from enough. In this paper, a software architecture for ATS is proposed. Four principles of object-oriented software engineering, considerations for computational experiments and parallel systems, issues of expansibility and cooperative development for the software design and implementation, as well as a software framework, which explains the functional structure of ATS are addressed. Furthermore, the rationale behind the framework is explained, with emphasis on the discussion about transportation scenarios, the agent characteristics of travelers, and the spread and flow of information in ATS.**

I. Introduction

TRAFFIC simulation is an area which has been developing rapidly with the advance of information and computer technology for the recent decades. It has become a powerful tool that assists people in traffic planning, control, evaluation, etc., and is attracting more and more interest. Micro-simulation, as a major category of traffic simulation, tracks every single traveler in traffic networks. It is able to capture detailed traffic behavior and effects, such as lane changing, and is considered more accurate to reality. TRANSIMS (TRansportation ANalysis and SIMulation System) [1] is a software package of this category, in which the simulation is originated by travelers' activities. Micro-simulation of traffic can generate abundant information about each traveler and help people a lot in transportation issues, in spite of a normally intensive computational effort.

Manuscript received August 15, 2007. This work was supported in part by the MOST Grant 2006CB705500, NNSFC Grants 60334020, 60621001, CAS Grant 2F05N01, 2006 CAS Presidential Special Scholarship Grant, SDAS Research Grant 2005006, and KLCSIS OPR Grant 20050101.

Jinyuan Li is with the Key Laboratory of Complex Systems and Intelligence Science of the Institute of Automation of Chinese Academy of Sciences, Beijing 100080, China, and currently a Ph.D. candidate of the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China (e-mail: lijinyuan00@mails.tsinghua.edu.cn).

Shuming Tang is with the Key Laboratory of Complex Systems and Intelligence Science of the Institute of Automation of Chinese Academy of Sciences, Beijing 100080, China and the Institute of Automation of Shandong Academy of Sciences, Jinan 250014, China (phone: 86-531-8260-5491; fax: 86-531-8296-2259; e-mail: sharron@ieee.org)

Xiqin Wang is with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China (e-mail: wangxq_ee@tsinghua.edu.cn).

Fei-Yue Wang is with the Key Laboratory of Complex Systems and Intelligence Science of the Institute of Automation of Chinese Academy of Sciences, Beijing 100080, China and the University of Arizona, Tucson 85721, USA (e-mail: feiyue@sie.arizona.edu).

However, the traditional micro-simulation is not the optimum choice. The huge number of travelers involved in a metropolitan transportation system makes it more stochastic and complex than expected. Travelers are free to choose or alter their travel plans and routes in most cases and their choices may be affected by weather, social and economic information, legal and regulation information, etc. The transportation system itself also has intimate connections with other urban systems, such as the logistical system, the infrastructure system as well as the ecological and resource systems. We believe that only if the micro-simulation of transportation encompasses the characteristics of humans and the combined effects with other urban systems could it be reasonable and effective enough to use. This motivates the development of Artificial Transportation Systems (ATS) proposed by Wang and Tang in [2].

It deserves to point out that there have been other models or approaches proposed in the past that adopt an integrated perspective of transportation systems like the one presented here. For instance, in the first, "trip generation" step of the classical four-step model[3], socio-economic and/ or land use characteristics of zones are taken into account to determine the level of aggregate demand for trips originating in, and attracted to, each study zone. ATS differs from the four-step model (and other similar models) in terms of integration in two important ways. First, the potential impacts on transportation systems of others systems are considered for each individual in ATS, rather than as a macroscopic factor that affects aggregate traffic demand in the four-step model. Second, the possible impacts of other urban systems considered in ATS are more comprehensive than that in the four-step model, without a commitment to socio-economic and/ or land use characteristics only.

The objective of ATS is to design and implement a software, which can not only simulate the real traffic, but also "grow" traffic behavior and effects, using the theories and methods of artificial societies and complex systems, agent modeling technology, and parallel and distributed computing technology. However, the research in both ATS-related theories and the design and implementation issues is far from enough. Concepts and a theoretical framework for ATS were proposed in [2] and [4], basic approaches in ATS were studied in [5], and a multi-agent architecture for ATS based on Petri net modeling was presented in [6]. But as for implementation, only [7] provides an example based on P2P computing and yet it focuses mainly on the aspect of computing. Therefore, thorough consideration for the software design and implementation of ATS is still in need. In this paper, a software architecture for ATS is proposed. The focus is on four principles of object-oriented software engineering, considerations for computational experiments and parallel systems, and issues of expansibility and cooperative development for the software design and implementation, and on a software framework, which explains the functional structure of ATS.

ATS is a large-scale software in terms of implementation. Therefore, four major principles of object-oriented software engineering are adopted for the design and implementation of ATS: layering, modularity, iteration, and object-orientation [8]. Besides, computational experiments and parallel systems are two major ways that ATS works and thus need to be considered carefully during the design phase. Moreover, ATS will be developed step by step and many researchers will be involved in the implementation process due to its large scale and current immaturity. So the issue of expansibility needs to be emphasized in order to keep the implementation process consistent. Interfaces among software modules need to be clearly defined beforehand to support the cooperative development.

The proposed software framework is a layered structure, which consists of four layers: application layer, function layer, representation layer, and resource layer. The function layer and the representation layer are further divided into a few modules. They will perform a variety of functions or represent different concepts. Such a structure is somewhat similar to that of other micro-simulation softwares such as TRANSIMS [8]. However, the partition of modules and the rationale behind them are different. Autonomy, sociality, learning, proactive, and mobility [9] are the essential characteristics of agents. In this paper, these characteristics of a traveler as an agent are emphasized through an individual behavior module. The representation and flow of information in ATS are also emphasized in the proposed framework, because traveler agents cannot "live" in ATS without sufficient information when their autonomy, learning, and proactivity are enhanced.

The rest of this paper is organized as follows. In section II, the four principles, the two considerations, and the two issues mentioned above for the software design and implementation are presented. The software framework is described in detail in section III, along with a block diagram for the framework. Finally, in section IV, conclusions are drawn and future work for the implementation issues of ATS is discussed.

II. Principles, Considerations, and Issues

A. Four Principles of object-oriented Software Engineering

Since ATS is a large-scale software in terms of implementation, four major principles of object-oriented software engineering are adopted for the design and implementation: layering, modularity, iteration, and object-oriented [8].

1) Layering: Layering divides a software into several layers (four in this work). Each layer uses the layer below it and serves the one above it. Layering can enhance the reusability of software components. Various components can be integrated in a clear way through layering.

2) Modularity: Different functional components are implemented as different modules. Each module provides services through a specific public interface. And the interface separates the implementation of the module from its services. Modularity reduces the coupling between different software components, makes the software easier to develop and maintain, and enhances the reusability, the expansibility, and the portability of the software codes.

3) Iteration: The software is developed in an iterative way. An iterative cycle contains three phases: design, implementation, and test. Thus, an executable or releasable version of the software can be obtained at the end of every iterative cycle. Iterative development can help us find defects in design during an earlier stage, reduce the risk of redesigning everything, reduce the cost of debugging, and boost the development process.

4)Object-oriented: Object-oriented programming languages(such as C++) provide the mechanisms of abstraction/capsulation, inheritance, and polymorphism that help us map concepts and models in the real world to classes and/or objects in the software domain. Object-oriented programming reduces the complexity of the software, makes the software easier to develop and maintain, and enhances the reusability and the expansibility of the software codes. Besides, traveler agents can be easily implemented as objects.

Objects, in our opinion, are appropriate for the representation of agents in ATS. However, it deserves to point out that more contemporary programming techniques like service-oriented, aspect-oriented, and even agent-oriented approaches had been proposed after the object-oriented method. These techniques can also be used in the implementation of ATS, but it is not a necessity. Take the agent-oriented approach as an example. Agents in ATS are individual travelers, rather than software components. Therefore, the adoption of the agent-oriented software engineering is not direct. Nevertheless, the investigation into the application of the techniques mentioned above in ATS can be a worthwhile work in the future.

B. Considerations for Computational Experiments and Parallel Systems

Computational experiments and parallel systems are two major ways that ATS works and thus need to be considered carefully during the design phase.

1) Computational Experiments: It means that ATS will be used as an experimental platform for studying various transportation problems, thus overcoming the difficulty of conducting experiments in real transportations systems. Accelerated tests for pressure, limit, failures, or disasters can be run on ATS repeatedly. Therefore, ATS has to be able to restore a transportation scenario quickly, readily, and accurately. Here, transportation scenario means the initial condition for ATS to start a computational experiment and will be explained in more detail in a later part of this paper. Hence, states of ATS need to be stored and loaded dynamically to restore a transportation scenario. Besides, scripts or configuration files are supposed to be supported in ATS to facilitate the construction of a specific scenario.

2) Parallel Systems: It means that ATS and a real transportation system can run in parallel and exchange information in real time. Ideas and methods developed in adaptive control can then be employed effectively in such a framework of parallel systems for decision-making in the real transportation system. Hence, ATS is supposed to be able to connect with real transportation facilities, such as vehicle detectors, traffic signal controllers, or surveillance devices in management centers, in real time. This connection can be established by Ethernet or RS232 on personal computers, because ATS can be run on PCs. Besides, human operators can participate in the execution of ATS, controlling a vehicle through a keyboard and observing the status through a monitor. In this way, human characteristics can be added in ATS directly.

C. Issues of Expansibility and Cooperative Development

As reviewed above, not enough effort has been devoted so far to ATS, especially its implementation issues. The implementation will advance step by step as more effort will be made. During this process, many researchers will take part and ATS will be developed in a cooperative way. Thus, more attention is supposed to be paid on the issues of expansibility and cooperative development.

1) Expansibility: ATS contains the transportation sub-system, the logistical sub-system, the social and economic sub-system, etc. The fundamental sub-systems, such as the transportation sub-system, will be implemented firstly and the other sub-systems will be integrated one by one. The implementation of functional modules will be the same. Thus, the dependency between different modules is supposed to be reduced as more as possible in order to enhance the expansibility and the consistency of ATS. The dependence needs to be recorded and archived if necessary.

2) Cooperative Development: ATS can not be implemented by one person due to its large scale. The issue of cooperation becomes important when many people are involved in the development. A good cooperative development is facilitated by two approaches. One is through third-party source code management software. The other is to define specific and clear interfaces among modules before coding. The interface protocols need to be recorded and archived for test and integration.

III. Framework

A. Layers

As mentioned above, the proposed framework is a layered structure, consisting of the following four layers:

1) Application Layer: It only contains a graphical user interface that provides a uniform interface for users to employ the various functions in the function layer. This interface can be used to set up transportation scenarios, run ATS, observe the animation of micro-simulation and emergent traffic behavior, and quantitatively analyze the artificial transportation.

2) Function Layer: It contains all the functional modules in ATS, including population generation, travel plan generation, traffic micro-simulation, analysis of simulation results, etc. It provides the application layer with services to invoke its functions.

3) Representation Layer: All physical concepts and models such as travelers, vehicles, roads, etc, are abstracted in this layer. These concepts and models are implemented as objects, including data and methods, which are manipulated by the functional modules.

4) Resource Layer: It only contains a data base for storing data including travelers, vehicles, roads, information, analysis results, etc.

The block diagram of the above framework is depicted in Fig. 1.

B. Modules

The function layer and the representation layer are the main parts of the proposed framework. The modules contained in these two layers and rationale behind them will be explained in detail as follows.

Modules in the Function layer

1) Environment Generation and Change: It generates environments where travelers live and travel and also changes the environments while ATS is running. It contains four sub-modules.

a) Traffic Network Generation: It generates roads, intersections, signals, and vehicle detectors. The attributes of roads and intersections include geometry, lengths, widths, lane assignments, lane marks, etc.

b) Other Environment Generation: It generates weather, guidance facilities (such as VMS and traffic radio), establishments, etc.

c) Weather Change: It changes weather while ATS is running. Weather can affect roads surface properties. For example, roads will be slipperier in a rain and deceleration rates of vehicles will be lowered accordingly.

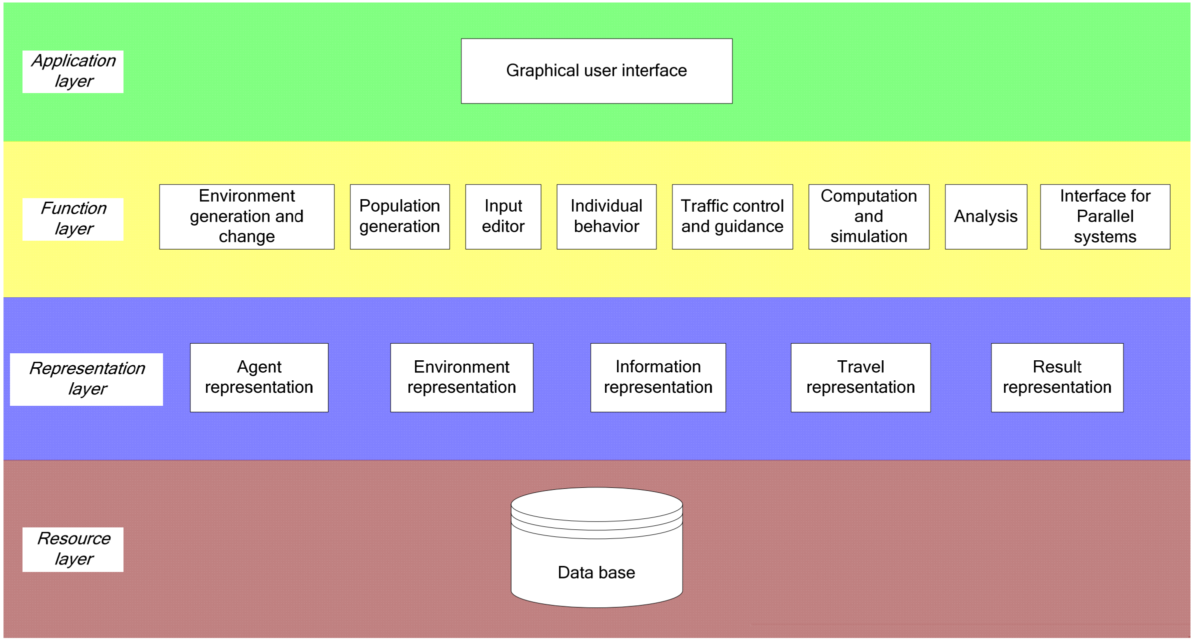

Fig. 1. Block Diagram of ATS Software.

e) Road Change: It can change roads surface properties or block a road if user need.

2) Population Generation: This module is used to generate travelers that are distributed in a traffic network before running ATS. It can be further divided into two sub-modules.

a) Individual Attribute Generation: It generates travelers with different attributes, which consist of gender, age, occupation, driving preference (such as cautious, middle, or aggressive), vehicle type, length, performance, etc.

b) Population Distribution: It distributes the generated population over the traffic network. Some may be in buildings or establishments beside roads and the others may be driving or waiting on roads. The vehicles on roads need to be assigned velocities that are consistent with those of their leading or following vehicles.

3) Input Editor: This module can be used to edit attributes of objects and transportation scenarios. It can be further divided into two sub-modules.

a) Object Attribute Editor: Using this sub-module, users can edit the attributes of all traffic objects in ATS, add or delete objects, and store or load configuration files of objects.

b) Transportation Scenario Editor: Using this sub-module, users can edit transportation scenarios and store or load configuration files of transportation scenarios.

Transportation scenario is designed for computational experiments in ATS. It is the initial condition of a computational experiment. Since a computational experiment can be repeated, a transportation scenario needs to be recordable. Furthermore, it is supposed to be stored or loaded as configuration files if needed.

A transportation scenario may be the set of the attributes of all the objects in ATS or a subset of this set. The reason is that the ultimate goal of ATS is to grow and cultivate traffic behavior rather than simply simulate or replay traffic. Therefore, a transportation scenario, as the initial condition of a computational experiment, is supposed to be the set of the key factors intrinsic to the experiment, but not contain any irrelevant factors. However, what factors are intrinsic to a computational experiment is a research topic that needs to be further studied.

4) Individual Behavior: It implements the traffic behavior of individual travelers (pedestrians, vehicles with a driver, or automated vehicles). This is the key module in the whole framework, in that the behaviors of agents, which are the key notions in ATS, are implemented in this module. Four sub-modules are needed.

a) Information Acquisition: Information is one of the foundations that guarantee travelers to live and travel in ATS. Information that possibly affects travelers include weather, legal and regulation information, logistic information, social and economic information, traffic guidance information, traffic signals, self-status(such as position, velocity, gas volume, etc.), other vehicles' status, etc. The information may affect one's travel in four ways.

1) The traveler determines his/her travel plan according to weather, legal and regulation information, logistic information, and social and economic information.

2) Traffic guidance information helps him/her choose a route.

3) He/She decides how to change his/her kinetic status (such as direction, velocity, acceleration/deceleration, whether to brake/stop, etc.) according to the ambient environment, other vehicles' status and his/her self-status.

4) He/She adjusts his/her plan (such as cancel the plan or take other vehicles) according to traffic guidance information, his/her self-status, etc.

In ATS, travelers are supposed to acquire information autonomously rather than through a uniform dispatching mechanism. This is because travelers in ATS are agents and they can perceive themselves, other agents, and ambient environments. Besides, the spread and flow of information in ATS has to obey some rules that accord with the reality. For example, VMS, traffic signals, and lane marks information can only be acquired by travelers when they get close to them.

This sub-module reflects the characteristics of sociality and learning of agents.

b) Decision and Learning: Travelers need to decide and learn after they acquire information. Decision means they need to decide how to move in the next time step. For example, car following, lane changing, overtaking, turning, decelerating, and braking are all implemented in this sub-module. Learning, which is one of the agent characteristics of travelers, means that travelers are supposed to improve their knowledge bases and rules of decision-making according to the acquired information.

It needs to point out that each traveler has his/ her own decision and learning sub-module. It means that distinct travelers possibly have totally different knowledge bases and rules of decision-making, which are determined by their distinctive personalities.

This sub-module reflects all the five characteristics of agents, namely autonomy, sociality, learning, proactive, and mobility.

c) Travel Plan Generation and Adjustment: The plan generation consists of the determination of origin-destination, schedules, and travel modes of travelers. A traveler may adjust or even cancel his/her plan during the travel due to the information he/she acquired. For example, a traveler who has planned to play tennis outside probably will cancel the plan due to the weather information that there will be a rain.

This sub-module reflects the characteristics of autonomy, proactive, and mobility of agents.

d) Route Choice and Adjustment: A traveler needs to choose a route for his/her travel according to traffic guidance information and may adjust the route during the travel. For example, a traveler who is going shopping probably will adjust his/her pre-planned route due to the guidance information that a street in front is oversaturated.

This sub-module also reflects the characteristics of autonomy, proactive, and mobility of agents.

5) Traffic Control and Guidance: Vehicle detectors, traffic signals, and traffic guidance are implemented in this module. It can be further divided into three sub-modules.

a) Signal Control: It includes control algorithms that generate signals.

b) Vehicle Detection: How vehicle detectors are set up is determined by signal control algorithms.

c) Traffic Guidance: It includes VMS, traffic radio, etc.

6) Computation and Micro-simulation: It contains a parallel and distributed computing platform and a micro-simulator.

a) Parallel and Distributed Computing Platform: It is the computing environment of ATS and provides a message-passing mechanism for computer nodes to exchange information while ATS is running. It is supposed to support the addition and deletion of computer nodes, dynamic balancing of computational loads, and diagnostic outputs to users. This platform can be established using existing parallel toolbox or libraries.

b) Micro-simulator: Travelers are actually moved by this sub-module. Travelers' intentions on how to move are generated in their own decision and learning sub-module, while they are taken into action in this centralized sub-module. Therefore, the micro-simulator is only supposed to update the positions of travelers according to physical laws, which is simpler than other micro-simulators. The reason behind this is that travelers are agents in ATS that can decide what to do autonomously.

7) Analysis: This module displays the running status of ATS and analyzes the results. Four sub-modules are needed.

a) GIS: It is used to search, display, and edit geographical information. Existing third-party GIS toolbox or libraries can be used to develop this sub-module.

b) Statistical analysis: The effectiveness of ATS is measured by this sub-module. It can be created using existing third-party statistical toolbox or libraries.

c) Animation: It provides animated display of vehicle movement and traffic signals while ATS is running.

d) Plotting: It plots analysis results. Scatter plot, line plot, histogram, etc, are supposed to be supported.

8) Interfaces for Parallel Systems: ATS will communicate with real transportation systems through these interfaces. They serve as decoders when receiving instructions from other systems and invoke functions in ATS, and as encoders when sending instructions to other systems.

Modules in the Representation Layer

Physical concepts and models are represented as objects, which consist of data (or attributes) and methods, through object-oriented programming languages in this layer.

Different modules in the function layer need different views of the objects. For instance, the micro-simulator needs the performance characteristics (the maximal acceleration, the maximal velocity, etc.) of a vehicle, but does not care about its type (a private car or a bus), while the statistical analysis does. Therefore, objects are supposed to be represented in layers. The attributes that are needed by several functional modules of an object are represented as its basic attributes while the ones that are needed specifically by a single functional module are represented as its additional or supplementary attributes. The methods of the objects are treated in the same way. In this manner, the overheads of manipulating objects by functional modules can be reduced and the efficiency of the system can be improved.

The objects representation can be divided into five categories, each corresponding to a module in the representation layer in Fig. 1.

1) Agent Representation: Agents include pedestrians, vehicles with a driver, and automated vehicles. So, travelers and vehicles are represented in this module.

2) Environment Representation: This module represents the common traffic environments, including networks, establishments, and signals (traffic signals and vehicle detection signals), as well as the "environment" for computational experiments -- transportation scenarios.

3) Information Representation: The various information in ATS is represented in this module, including data and time, weather, traffic guidance information (traffic radio, VMS, etc.), legal and regulation information, logistic information, and social and economic information.

4) Travel Representation: It includes plan representation and route representation.

5) Result Representation: Effectiveness measurements (traffic flow rate, delay, number of stops, etc.) and analysis results (the mean, variance, and other statistics of the concerned measurements) are represented in this module.

IV. Conclusion and Future Work

This paper proposes a software architecture for Artificial Transportation Systems (ATS). The focus is on four principles of object-oriented software engineering, considerations for computational experiments and parallel systems, and issues of expansibility and cooperative development for the software design and implementation, and on a software framework, which explains the functional structure of ATS.

The proposed software framework is a layered structure, consisting of four layers: application layer, function layer, representation layer, and resource layer. The function layer and the representation layer, each of which contains several functional modules, are the main parts of this framework. Functions of the modules and the sub-modules included in each module are presented concisely. The rationale behind the module choice is explained in detail, with emphasis on the discussion about transportation scenarios, the agent characteristics of travelers, and the spread and flow of information in ATS. However, no detailed implemental concerns are involved.

This work only serves as a rough description of the requirements for an ATS prototype, which can be a preparative effort before the actual implementation. Much work needs to be done to make the proposed framework solid enough to implement. Detailed functions and interfaces need to be defined and third-party tools that assist in the implementation of ATS need to be chosen. The definition of transportation scenarios also needs to be further studied theoretically or experimentally.

References

[1] C. L. Barrett, et al, "Transportation Analysis Simulation System (TRANSIMS): Overview," Los Alamos National Lab., Los Alamos, NM Rep. LA-1658, 1999.

[2] W. G. Lu. (1998, October). Urban Transport Models: A Review. working paper 039 [Online]. Available: http://www.btre.gov.au/info.aspx?ResourceId=27&NodeId=23

[3] F.-Y. Wang and S. M. Tang, "Concept and Framework of Artificial Transportation Systems," *J. Complex Systems and Complexity Science*, vol. 1, no. 2, pp. 52-59, 2004.

[4] F.-Y. Wang and S. M. Tang, "A Framework for Artificial Transportation Systems: From Computer Simulations to Computational Experiments," in *Proc. 8th Int. IEEE Conf. Intelligent Transportation Systems*, Vienna, Austria, 2005, pp. 1130-1134.

[5] S. M. Tang, "The Preliminary Study for Approaches in Artificial Transportation Systems," Ph.D. dissertation, Institute of Automation, Chinese Academy of Sciences, 2005.

[6] F. He, Q. H. Miao, etc, "Modeling and Analysis of Artificial Transportation System Based on Multi-agent Technology," in *Proc. 9th Int. IEEE Conf. Intelligent Transportation Systems*, Toronto, Canada, 2006, pp. 1120-1124.

[7] Q. H. Miao, F.-Y. Wang, and S. M. Tang, "An Implementation of Artificial Transportation Systems based on P2P Computing," in *Proc. IEEE Int. Conf. Vehicular Electronics*, Shanghai, China, 2006, pp. 93-97.

[8] K. P. Berkbigler, B. W. Bush, and J. F. Davis, "TRANSIMS Software Architecture for IOC-1," Los Alamos National Lab., Los Alamos, NM Rep. LA-UR-97-1242, 1997.

[9] S. M. Tang, "A Preliminary Investigation on Artificial Transportation Systems," *Journal of System Simulation,* vol. 17, no. 3, pp. 704-709, 2005.