Hierarchical Modeling and Simulation Environment for Intelligent Transportation Systems

Jong-Keun Lee, Ye-Hwan Lim and Sung-Do Chi

SIMULATION 2004; 80; 61
DOI: 10.1177/0037549704042860

Hierarchical Modeling and Simulation Environment for Intelligent Transportation Systems

Jong-Keun Lee
Ye-Hwan Lim
Sung-Do Chi
Intelligent Systems Research Laboratory
Department of Computer Engineering
Hangkong University
Koyang-si, Hwajeon-dong 200-1, Seoul 411-791, Korea
leejk@mail.hangkong.ac.kr

This article presents a hierarchical and modular traffic simulation environment for intelligent transportation systems (ITS). A conventional traffic simulation model is classified as microscopic or macroscopic; however, abstraction-level, vehicle-level, cell-level, road-level, and street-level traffic models can be coherently integrated and developed. Such an abstraction process is important for flexible traffic analysis since it reduces the complexity of a model, retaining its validity relative to the modeling objectives and experimental condition. To do this, the authors have proposed the four-layered approach: (1) system entity structure/model base, (2) model abstraction, (3) traffic modeling, and (4) ITS simulation systems layer. A proposed methodology has been successfully applied for building the advanced traveler information system and the advanced traffic management system. The abstraction method of a road network in traffic modeling and simulation will be widely used because it is possible to express and analyze various requirements of the traffic analyst hierarchically and structurally.

Keywords: DEVS formalism, SES/MB, traffic simulation, model abstraction method, ATIS: The Next Generation, I^3D^2 Transportation Simulation System

1. Introduction

Intelligent transportation system (ITS) is an umbrella term. It covers the application of a wide variety of computer, communication, positioning, sensing, control, and other information-related technologies. Major categories of ITS include traffic management systems and traveler information systems [1, 2]. Currently, simulation has become a tool commonly used to understand the characteristics of a traffic system and to select an appropriate design for ITS. Thus, modeling and simulation techniques provide a convenient way to evaluate the alternative signal control strategies at the operational level of advanced traffic management systems (ATMS) and to generate the simulation-based forecasting information for advanced traveler information systems (ATIS) [2, 3]. Traffic simulation models can be typically classified as either microscopic or macroscopic. In microscopic simulation models, individual vehicles are studied, and attention is focused on their performance in the context of the whole traffic network system. Thus, this approach has usually been adopted for the simulation of relatively small or simple systems. On the other hand, the macroscopic approach can be well used to simulate the large network or complex traffic system. This approach uses simplified models of roads and intersections [4]. Discrete event modeling techniques can be suitably employed to describe macroscopic models. However, by allowing the level of abstraction that is only possible in the hierarchical modeling and simulation environment, both microscopic and macroscopic levels of modeling can be accomplished. This abstraction process is critical for designing the traffic simulation system by reducing the complexity of the model while retaining its validity relative to the modeling objectives and experimental conditions. Moreover, discrete event modeling and the simulation environment support an integrated simulation of traffic network, vehicles, and other traffic management systems. For example, the simulated congestion dynamics of the traffic network may be broadcast to the driver model of each vehicle. The driver model is able to change his or her previously decided path upon receiving the newly forecasted traffic information during the simulation. Also, the simulation result can be reflected to the signal light control during the simulation. Therefore, various components such as traffic network, vehicles

SIMULATION, Vol. 80, Issue 2, February 2004 61-76

DOI: 10.1177/0037549704042860

with drivers, and traffic management systems (i.e., signal lights, sensors, etc.) can be independently modeled and coherently coupled to describe realistic traffic dynamics. Such an integrated simulation environment is not possible in conventional software environments.

This article presents a hierarchical and modular traffic simulation environment for ITS. To do this, we have proposed the four-layered approach on the basis of the object-oriented programming environment: (1) system entity structure/model base, (2) model abstraction, (3) traffic modeling, and (4) ITS simulation systems layers. A developed S/W tool for ATIS, called "ATIS: The Next Generation," supports not only conventional graphics-based traffic information but also the following advanced features: macroscopic traffic modeling and simulation, dynamic path searching, virtual driving, and virtual investigation. Another S/W tool for ATMS, called the "I^3D^2 Transportation Simulation System," has been also successfully developed for supporting an intelligent, interactive, integrated, and distributed microscopic transportation simulation environment.

The remainder of this article is organized as follows. We first briefly review the background on conventional traffic simulation systems. Then, we propose a four-layered design methodology for the transportation modeling and simulation systems. This is followed by two case studies—ATIS: The Next Generation and the I^3D^2 Transportation Simulation System.

2. Background on Traffic Simulation Systems

Traffic modeling in the simulation environment is adopted to analyze the traffic congestion of the given traffic network. The congestion on each road should be changed depending on the current traffic status, such as the number of vehicles, control strategy of the signal light, traffic accidents, and scheduled traffic events (i.e., road construction, parade, etc.).

Traditionally, traffic simulation models were developed independently for different facilities (e.g., freeways, urban streets, etc.). Therefore, a wide variety of simulation models exists for various applications. Most of these models were developed for evaluation and real-time support of ATIS operation and traffic prediction (e.g., DYNASMART [5], DYNAMIT [6]). Most of the old-generation models (e.g., NETSIM [7] and earlier versions of AIMSUN2 [8]), however, do not represent vehicle paths [1]. Analytic flow models such as FREFLOW, TRANSYT-7F, or HCS do not track individual vehicle movements and do not support sufficient detail for analyzing ATMS and ATIS studies [9]. Although NETSIM and FRESIM [10] do provide microscopic performance models, these microscopic models cannot effectively meet the requirements of evaluating integrated ATIS and ATMS applications at an operational level because of limited representation of travel behavior or inflexibility in modeling more advanced surveillance and control systems.

A new generation of traffic simulation models has been developed for ITS applications. Examples are AUTOS [11] and THOREAU [9]. AUTOS is a macroscopic traffic model developed for ATMS applications at traffic management centers, such as the testing of signal optimization, emergency vehicle response management, and human factors. THOREAU is a microscopic model developed for ITS evaluation. However, it has a very long running time. PARAMICS [12, 13] is also a recently developed microscopic traffic simulation and visualization tool for ITS analysis (i.e., a suite of high-performance software tools for microscopic traffic simulation). Other microscopic simulation models are also under development for modeling automated highway systems.

While all of these models have all been successfully applied in particular studies, a common shortcoming is the relatively limited range of applications. Some of them are designed for particular applications and are useful only for specific purposes, while others do not support advance surveillance and control systems or integrated networks. No model has the integrated component and functionality required for evaluating dynamic control and route guidance strategies on general networks [1].

Recently, Deshpande et al. [14] have developed a hybrid traffic modeling S/W environment, called SHIFT [14, 15], which allows specifications of both continuous-time and discrete event dynamics. This research proposed the formal semantics of the model and was successfully applied to automated highway systems. However, it is not clear that it could support advanced modeling features, such as hierarchical and modular modeling, distributed simulation, and model abstraction capabilities. The advantages of these capabilities—such as reduction in model development time, support in reusing a database of models, and aid in model verification and validation—are well accepted [15, 16]. In ITS studies especially, these capabilities provide a convenient means to construct and analyze complex traffic network systems. By employing these advantages, we have proposed the four-layered design methodology for ITS simulation.

3. Traffic Modeling and Simulation Approach

On the basis of the object-oriented programming environment, we have proposed the traffic modeling and simulation methodology, as shown in Figure 1. The features of the ITS simulation system can be understood better by organizing them within a set of layers that characterizes its software design structure. Layer 1 provides a hierarchical, modular, and distributed modeling and simulation environment on which the system is built. The properties of this lowest layer make it possible to realize similar properties at the higher layers. The next layer, model abstraction, can be framed in terms of its ability to reduce the complexity of a model. Microscopic and macroscopic traffic modeling can be accomplished within layer 3. ITS simulation environment can be finally achieved in layer 4.

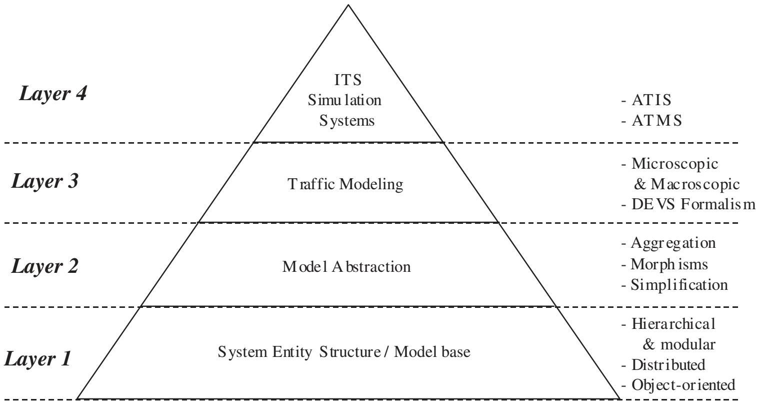


Figure 1. Layered approach for designing transportation simulation systems

4. Layer 1: System Entity Structure/Model Base

The system entity structure/model base framework was proposed by Zeigler as a step toward marrying the dynamics-based formalism of simulation with the symbolic formalism of artificial intelligence. It consists of two components: a *system entity structure* (SES) and a *model base* (MB) [17-21].

The *system entity structure*, declarative in character, represents knowledge of decomposition, component taxonomies, coupling specification, and constraints. The *entities* of SES refer to conceptual components of reality for which models may reside in the MB. An entity may have several *aspects*, denoting the decomposition and having several entities. An entity may also have several *specializations*, each representing a classification of the possible variants of the entity. The *pruning* operation extracts a substructure of the SES by selecting one aspect and/or one specialization for each entity in the SES. Hierarchical simulation models can be constructed by applying the transform function to *pruned entity structures* (PESs) in working memory.

As a concrete example of SES, consider Figure 2. The traffic network system is divided basically into the NODE model, which includes the concept of an intersection, and the LINK model, which includes the concept of roads. The LINK model is divided into the MICRO model, which is a microscopic road model, and the MACRO model, which is the macroscopic road model according to the abstraction relation of the model in section 5. The MACRO model is abstracted to CELL, ROAD, and STREET models and divided through the aggregation method, which is the abstraction method in space notion. In this way, all possible entities and configurations for the traffic network system can be easily represented within the SES. Applying the pruning operation, we can extract a specific traffic network configuration.

The *model base* contains models that are procedural in character. They are expressed in discrete event system specification (DEVS) formalism. Also, they are a theoretically well-grounded means of expressing modular discrete event simulation models. The DEVS formalism is materialized by formal programming languages, such as C++ and Java, for modelers to access easily (see www.acims.arizona.edu). Traffic models in this article were materialized by DeSim (C++ for DEVS simulation), which Hankuk Hangkong University built according to the DEVS formalism.

As explained before, Figure 3 explains the SES/MB environment. The PES is created by pruning the entity structure files in the system entity structure. The PES is compounded with each model in the model base by the transform command and creates a working model for simulation. Working models are combined with DEVS abstraction simulators and simulation starts. Detailed descriptions on SES/MB are available in Zeigler [17] and Zeigler, Kim, and Praehofer [21].

5. Layer 2: Model Abstraction

Abstraction is a key to model construction for simulation [21-24]. *Abstraction*, as a process, refers to a method or algorithm applied to a model to simplify its complexity while preserving its validity. The amount of detail in a model can be taken as the “product” of its scope and resolution. *Scope* refers to how much of the real world is represented; *resolution* refers to the number of variables in the model and their precision or granularity. Given a fixed amount of resources and a model complexity that exceeds this limit, there is a trade-off between scope and resolution. We may be able to represent some aspects of a system accurately, but only a few components will be presentable. Or, we may be able to provide a comprehensive view on the system but only at a relatively low resolution. Such an abstraction process

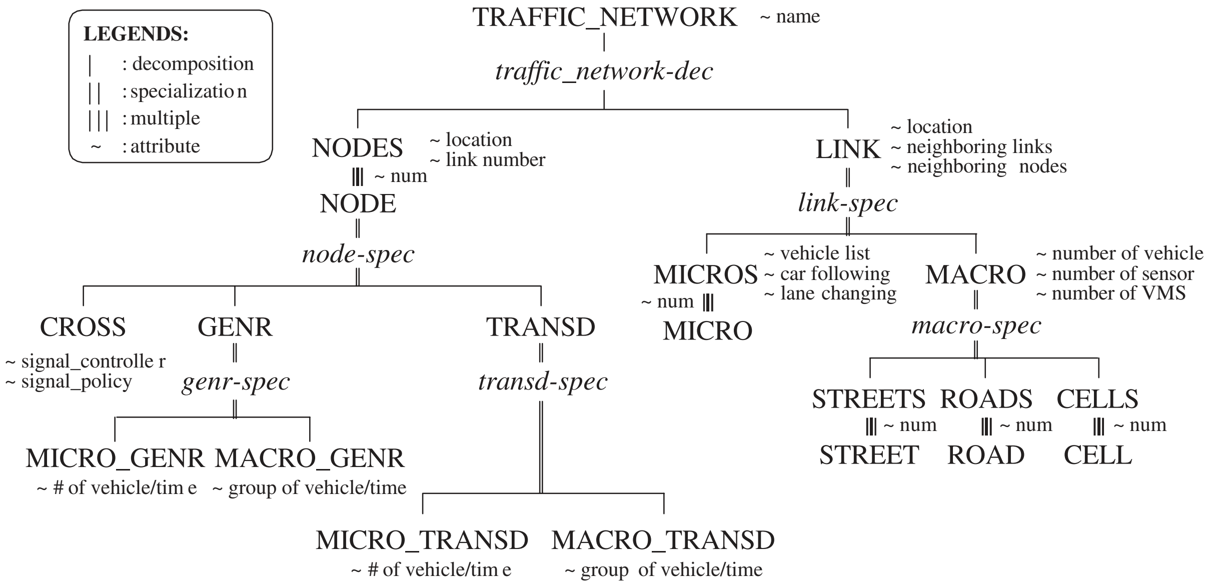

LEGENDS:
| : decomposition
|| : specialization
||| : multiple
~ : attribute
TRAFFIC_NETWORK ~ name
traffic_network-dec
NODES ~ location ~ link number
~ num
NODE
node-spec
LINK ~ location ~ neighboring links ~ neighboring nodes
link-spec
MICROS ~ vehicle list ~ car following ~ lane changing
~ num
MICRO
MACRO ~ number of vehicle ~ number of sensor ~ number of VMS
macro-spec
STREETS ROADS CELLS
~ num
STREET ROAD CELL
CROSS
~ signal_controller ~ signal_policy
GENR
genr-spec
TRANSD
transd-spec
MICRO_GENR MACRO_GENR
~ # of vehicle/time ~ group of vehicle/time
MICRO_TRANSD MACRO_TRANSD
~ # of vehicle/time ~ group of vehicle/time

Figure 2. System entity structure (SES) representation of the transportation system

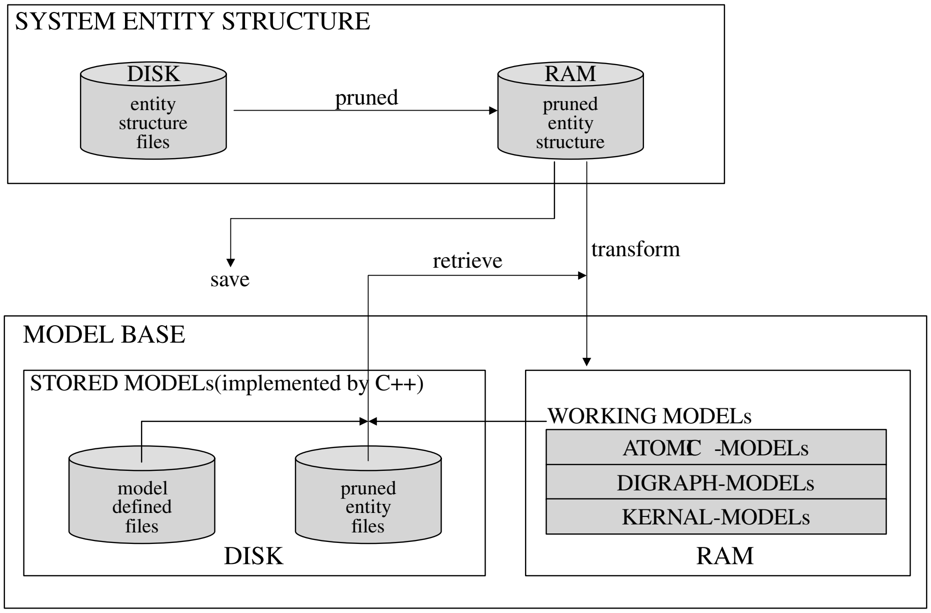


Figure 3. The system entity structure/model base environment [17]

is especially important for flexible traffic analysis since it reduces the complexity of a model while retaining its validity relation to the modeling objectives and experimental frame (i.e., conditions under which the model will be observed or experimented with). In this abstraction process for the traffic model, the following aggregation method can be suitably applied, as shown in Figure 4: (1) grouping vehicles on a given segment into cells, (2) abstracting cells into a simplified form (i.e., ROAD), and (3) grouping the ROAD of a given area into a street. In this way, the real

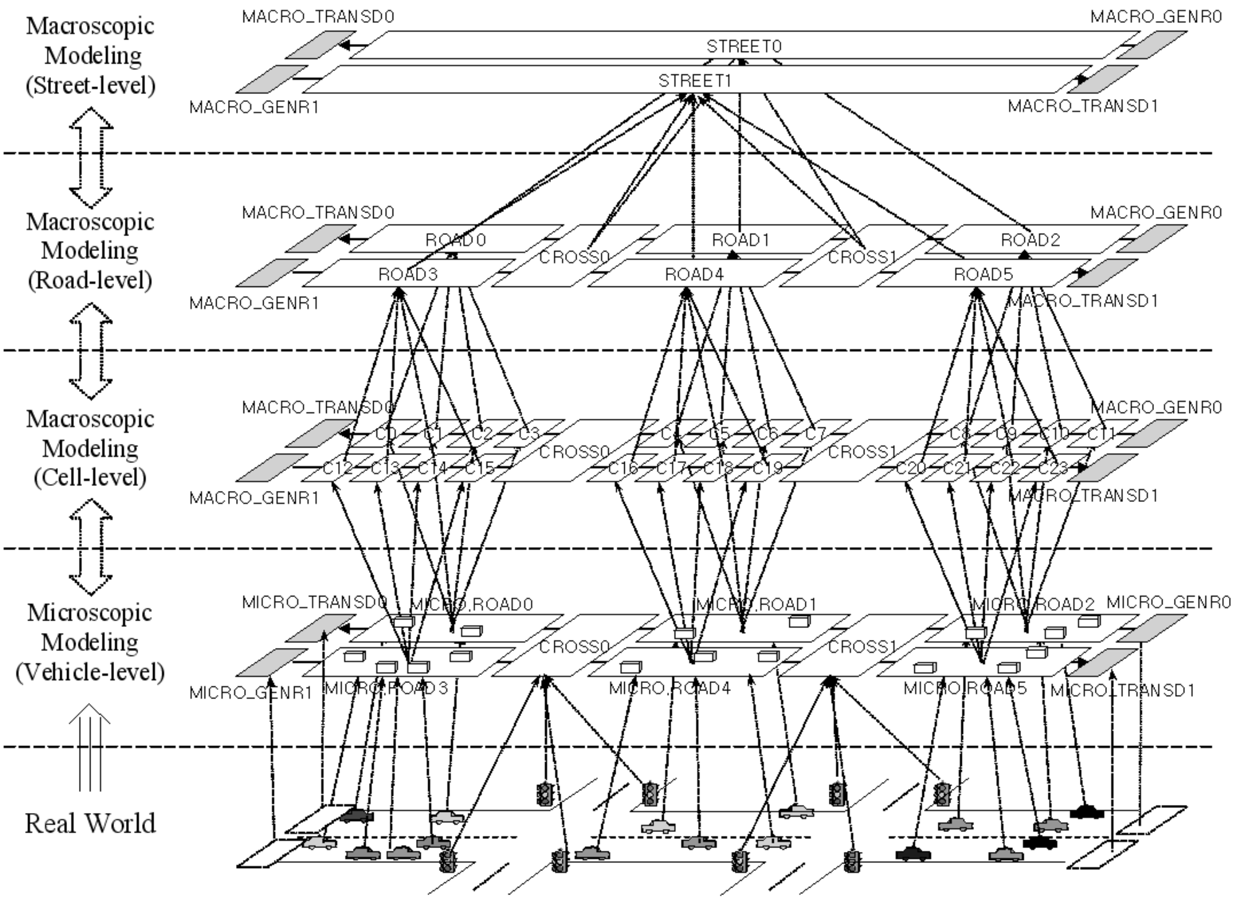


Figure 4. Abstraction relations for traffic modeling

world can be coherently abstracted from the vehicle level (microscopic level) to the street level (macroscopic level).

6. Layer 3: Traffic Modeling

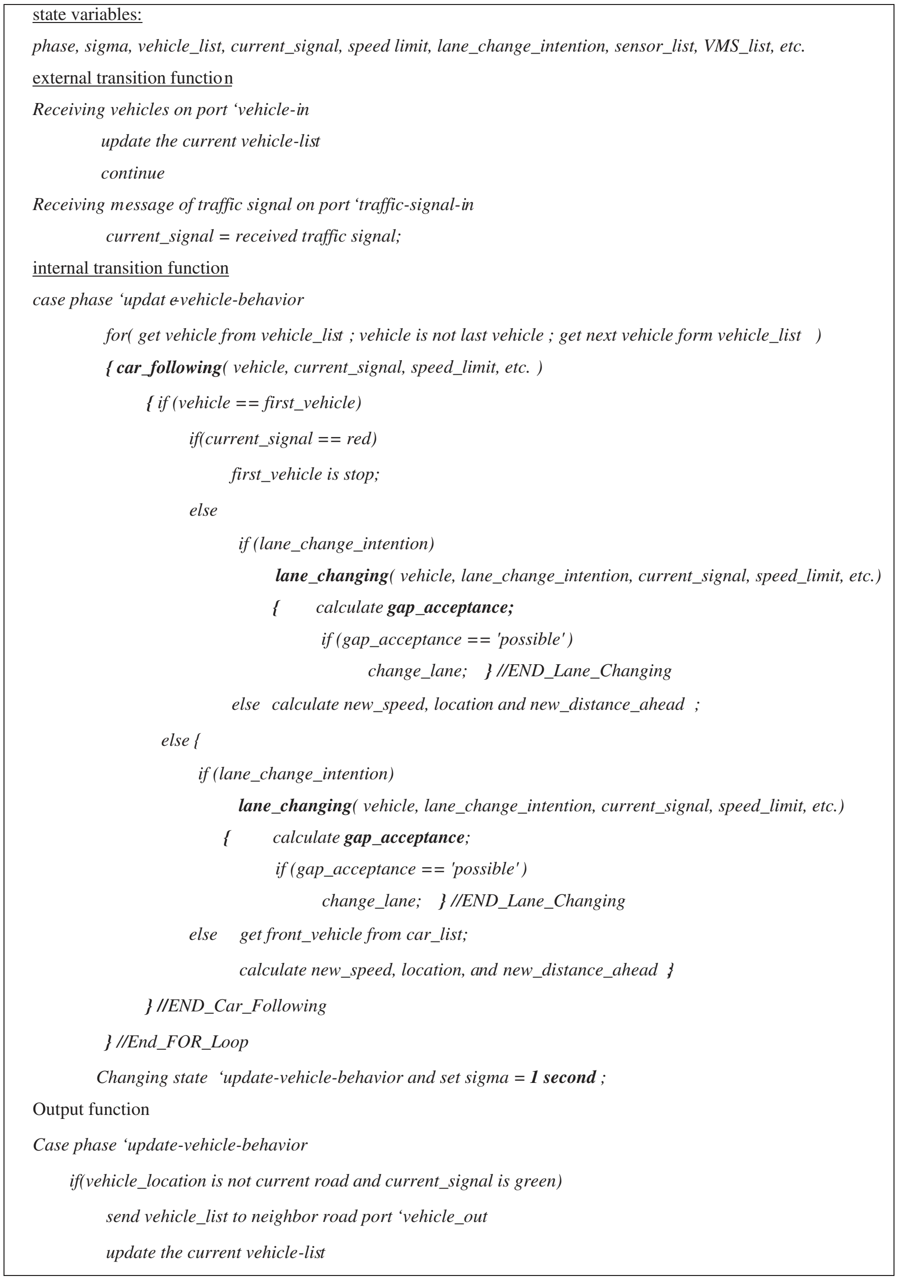
6.1 Vehicle-Level Modeling

Until recently, most of the models in general use have been analytical or stochastic. These models calculate the performance of the elements of the road network based on a generalized mathematical formula. These macroscopic or analytical models do not model individual vehicles in the network. Rather, they model how all vehicles behave on average (with knowledge of the likely distribution of behavior about this average).

Microscopic simulation traffic models predict the performance of the road network by defining rules for various aspects of the interaction among drivers, vehicles, and the road network and then simulating all the individual vehicles in the network. With the microscopic simulation traffic model, the behavior of each individual vehicle on the road network is continuously modeled throughout the simulation time period (based on the discrete time approach). Each vehicle moves through the network according to various driver behavior models such as car following, lane changing, and gap acceptance. The movement of the vehicle must also take account of control elements such as traffic signals, sign-controlled intersections, and traffic rules [25].

MICRO.ROAD (Microscopic ROAD) Model. Figure 5 shows the simple pseudo-code of a microscopic road model at the vehicle level. The microscopic road model applies the car-following algorithm and the lane-changing algorithm, which includes gap acceptance insides to individual vehicles per every unit of time and decides the condition of the model through the behaviors of the individual vehicles it has. Also, various rules that can affect the traffic signal of a neighboring CROSS—speed limit of roads and the movement of individual vehicles—are applied.

The car-following model studies how the driver of a following vehicle tries to conform with the behavior of the lead vehicle with acceleration and/or deceleration activities



```
state variables:
phase, sigma, vehicle_list, current_signal, speed limit, lane_change_intention, sensor_list, VMS_list, etc.
external transition function
Receiving vehicles on port ‘vehicle-in
        update the current vehicle-list
        continue
Receiving message of traffic signal on port ‘traffic-signal-in
        current_signal = received traffic signal;
internal transition function
case phase ‘update-vehicle-behavior
        for( get vehicle from vehicle_list ; vehicle is not last vehicle ; get next vehicle form vehicle_list )
        { car_following( vehicle, current_signal, speed_limit, etc. )
            { if (vehicle == first_vehicle)
                 if(current_signal == red)
                      first_vehicle is stop;
                 else
                      if (lane_change_intention)
                          lane_changing( vehicle, lane_change_intention, current_signal, speed_limit, etc.)
                          {      calculate gap_acceptance;
                                 if (gap_acceptance == 'possible' )
                                      change_lane;   } //END_Lane_Changing
                      else   calculate new_speed, location and new_distance_ahead ;
              else {
                  if (lane_change_intention)
                      lane_changing( vehicle, lane_change_intention, current_signal, speed_limit, etc.)
                   {      calculate gap_acceptance;
                          if (gap_acceptance == 'possible' )
                               change_lane;   } //END_Lane_Changing
                 else    get front_vehicle from car_list;
                         calculate new_speed, location, and new_distance_ahead }
            } //END_Car_Following
        } //End_FOR_Loop
       Changing state  ‘update-vehicle-behavior and set sigma = 1 second ;
Output function
Case phase ‘update-vehicle-behavior
    if(vehicle_location is not current road and current_signal is green)
        send vehicle_list to neighbor road port ‘vehicle_out
        update the current vehicle-list
```

Figure 5. Pseudo-code of MICRO_ROAD model

[4, 26]. In this model, the first vehicle on the road decides its acceleration/deceleration rate according to the speed limits of the occupied road, the speed capacity of the vehicle, and the characteristics of the driver. On the other hand, if the vehicle is not the first one on the occupied road, the acceleration/deceleration rate is decided by the car-following model, which provides this rate according to the distance and speed of the front vehicle.

The lane-changing model, the core of the multilane microscopic traffic model along with the car-following model, is relatively complex because of the intervention of the driver’s characteristics and decision making [1].

```
State variables :phase, sigma, vehicle_list, current_signal, neighbor_road_congestion
external transition function
Receiving start message on port ‘start_genr
        change phase to  ‘vehicle_generating and set sigma = 0;
Receiving stop message on port ‘stop_genr
        passivate;
internal transition function
case phase ‘vehicle_generating
        vehicle_generating_time = distribution of headway (poisson count or pearson type III)
        change phase to ‘vehicle_generating and set sigma = vehicle_generating_time
output function
case phase ‘vehicle_generating
         if(current_signal == green and neighbor_road_congestion != full)
              send generated vehicle_list to port ‘ generator_out
         else add generated vehicle to vehicle_list
```

Figure 6. Pseudo-code of the MICRO_GENR model

In general, the lane-changing model is classified into two types: mandatory and discretionary. The mandatory lane-changing model occurs when the driver has to change his or her lane to avoid a restricted lane, bottleneck area, and so on. On the other hand, the discretionary lane-changing model refers to cases in which the driver changes his or her lane to increase speed, bypass slower or heavy vehicles, connect to the next link on the path, and so on.

MICRO_GENR (Microscopic GENERATOR) Model. This is a model that generates each vehicle according to headway distribution (Poisson count distribution, formal distribution, Pearson type III distribution, etc.) as it combines with microscopic road model. Therefore, the MICRO_GENR model, which is a model that abstracts outside environments that consist of the microscopic road model, has rules that consider congestion status and the traffic signal of neighboring roads. Figure 6 represents a simple pseudo-code of the MICRO_GENR model.

MICRO_TRANSD (Microscopic TRANSDUCER) Model. This is a model that abstracts outside environments of road nets, which consist of a microscopic road model such as the MICRO_GENR model, and manages individual vehicles from the road network. It also has an influence on the neighboring microscopic road model according to the traffic signal and congestion condition of the outside road network. Figure 7 represents a simple pseudo-code of the MICRO_TRANSD model.

6.2 Cell-Level Modeling

Microscopic behaviors for individual vehicles at the vehicle level are abstracted to cell-level models, with occupancy of the space based on discrete time. The cell space modeling technique provides an efficient way to build abstraction-related models when more detailed analysis is required. The road can be spatially decomposed into small pieces of road so that a detailed level of analysis can be achieved. A cell space model structure is shown in Figure 4. It considers models of spatially distributed systems with the property of uniformity (or homogeneity). A classical prototype of such models is cellular automaton, which represents both space and time in a discrete form. The uniformity in space, which will be apparent shortly, means that the system will possess spatial invariance (with respect to translation) as well as time invariance. This uniformity simplifies the description since only one cellular component and its interaction with its neighbors need to be specified. A brief description on each cell space model is as follows.

C_n **(CELL) Model.** This is the unidirectional cell space model of a road. It receives a group of vehicles from the rear cell and sends a group of vehicles to the front cell at every periodic event time, which depends on the distance of the road and the average speed. The DEVS representation of C_n is shown in Figure 8.

6.3 Road-Level Modeling

The traffic network model basically consists of nodes and links, as shown in Figure 2. A node represents either an intersection (CROSS), an external sink (TRANSD), or a source (GENR); a link (ROAD) denotes a unidirectional pathway for vehicles between nodes. A brief description on each model in MBASE follows.

```
State variables :phase, sigma, vehicle_list, current_signal, neighbor_road_congestion
External transition function
Receiving start message on port 'start_transd
          change phase to 'received_vehicle and set sigma = 0;
Receiving vehicle_list on port 'vehicle_in
          add received vehicle_list to current vehicle_list
internal transition function
case phase 'received_vehicle
        change phase 'not_received_vehicle and set sigma = red_time
case phase 'not_received_vehicle
        change phase 'received_vehicle and set sigma = green_time
output function
case phase 'received_vehicle
        send green_signal and congestion to port 'neighbor_road_out
case phase 'not_received_vehicle
        send red_signal and congestion to port 'neighbor_road_out
```

Figure 7. Pseudo-code of the MICRO_TRANSD model

```
state variables: phase, sigma, group of cars, periodic time, etc.
external transition function
Receiving group of passed cars on port 'car-in
          update the current group of cars
          continue
internal transition function
case phase
          passing: change phase to 'stay
                   update sigma is periodic time
output function
case phase
          passing: send group of passing cars to port 'front-cell-out
```

Figure 8. Pseudo-code of the C_n model

CROSS Model. This is a signal light model. It sends a light signal to its neighboring ROADs depending on the signal-switching control strategy, which can be characterized by cycle, split, and offset. DEVS formalism, represented in the simplified pseudo-code form of CROSS, is summarized in Figure 9.

ROAD (Macroscopic ROAD) Model. This is an unidirectional road model between nodes (i.e., CROSS, GENR, and TRANSD). It receives a group of vehicles from rear nodes. When it receives a green signal from the front node, it starts to send a group of vehicles to its neighboring node. The number of vehicles to be sent to next roads can be computed on the basis of the duration of the green signal, the available number of vehicles, the probabilistic route choice, and so forth of the ROAD. Traffic accidents and/or other blocking events can be specified within this model. The DEVS representation for ROAD is summarized in Figure 10.

MACRO_GENR (Macroscopic GENERATOR) Model. This is an external source model. It injects a group

```
state variables: phase, sigma, signal light control strategy
external transition function
Receiving start message on port ‘start-signal
          change phase to ‘signaling during ‘signal-time (based on state variables)
internal transition function
case phase
          ‘signaling :  rest the sigma to ‘signaling -time
output function
case phase
          ‘signaling : send signal  message to port ‘signal-x (depends on the signal control strategy).
```

Figure 9. Pseudo-code of the CROSS model

```
state variables: phase, sigma, # of lanes, # of cars, route choice, incidence level, etc.
external transition function
Receiving signal message on port ‘signal-in
          change phase to ‘passing during ‘signal-time
Receiving # of passed cars on port ‘car-in
          update the current # of cars
          continue
internal transition function
case phase
          passing: change phase to ‘stay
output function
compute # of passing cars (based on state variables)
case phase
        passing: send # of passing cars to port ‘u-out, ‘l-out, ‘s-out, ‘r-out
```

Figure 10. Pseudo-code of the ROAD model

of vehicles to its front ROAD, depending on the control strategy, so that it behaves like a mixed model of CROSS and ROADs that is external to the given traffic network. The DEVS representation of the MACRO_GENR model is summarized in Figure 11.

MACRO_TRANSD (Macroscopic TRANSDUCER) Model. This is an external sink model. It first generates a light signal to its rear ROAD and then receives a group of vehicles from the ROAD so that it behaves like a mixed model of CROSS and ROADs that is external to the given traffic network. The DEVS representation of the MACRO_TRANSD model is summarized in Figure 12.

The simulation process is initiated by specifying the given traffic situations to test a candidate traffic control strategy (i.e., representing a particular control strategy, current number of cars, routing probability, etc.). Subsequently, the traffic control system can be refined to improve its performance. This framework represents a laboratory for testing and refinement of control system designs. Such a laboratory has been identified as a necessary tool for future development in dynamic traffic control systems [27].

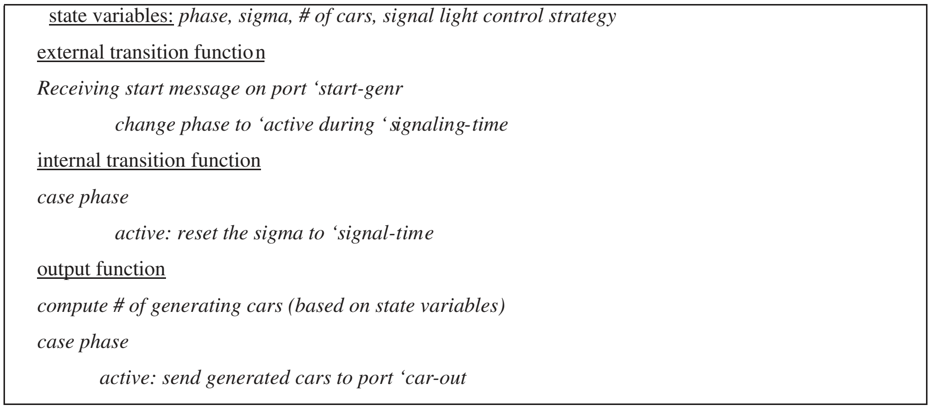
6.4 Street-Level Modeling

By increasing the level of abstraction, street-level traffic modeling can be achieved. This macroscopic model may be properly employed for studying large-scale traffic analysis problems.

STREET Model. This is a macroscopic model that abstracts several ROADs and CROSSes. The DEVS representation of the STREET model is summarized in Figure 13.

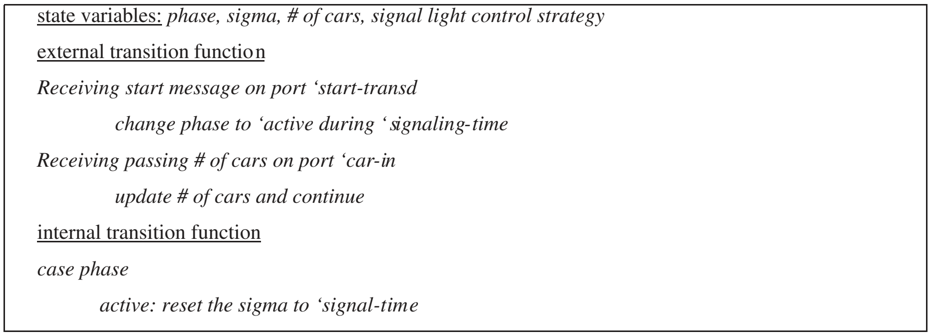
6.5 Simulation Example

In this section, simulation is performed by making simulation models according to the abstraction method for the



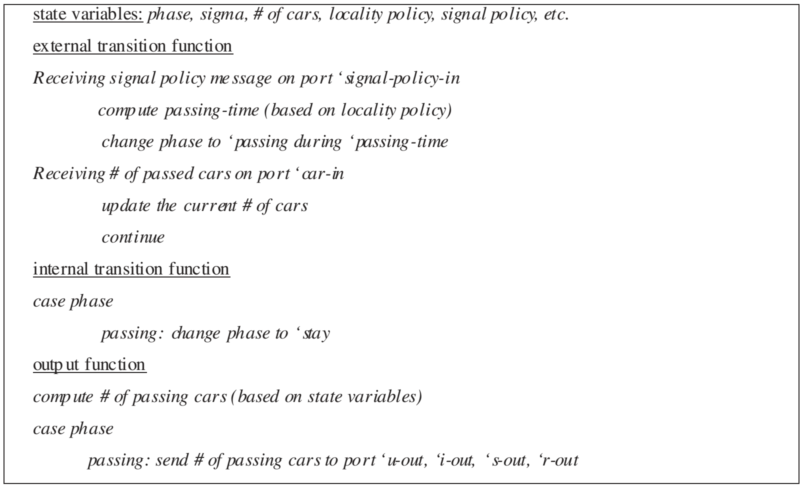
```
state variables: phase, sigma, # of cars, signal light control strategy
external transition function
Receiving start message on port ‘start-genr
        change phase to ‘active during ‘signaling-time
internal transition function
case phase
        active: reset the sigma to ‘signal-time
output function
compute # of generating cars (based on state variables)
case phase
      active: send generated cars to port ‘car-out
```

Figure 11. Pseudo-code of the MACRO_GENR model



```
state variables: phase, sigma, # of cars, signal light control strategy
external transition function
Receiving start message on port ‘start-transd
        change phase to ‘active during ‘signaling-time
Receiving passing # of cars on port ‘car-in
        update # of cars and continue
internal transition function
case phase
      active: reset the sigma to ‘signal-time
```

Figure 12. Pseudo-code of the MACRO_TRANSD model



```
state variables: phase, sigma, # of cars, locality policy, signal policy, etc.
external transition function
Receiving signal policy message on port ‘signal-policy-in
        compute passing-time (based on locality policy)
        change phase to ‘passing during ‘passing-time
Receiving # of passed cars on port ‘car-in
        update the current # of cars
        continue
internal transition function
case phase
        passing: change phase to ‘stay
output function
compute # of passing cars (based on state variables)
case phase
      passing: send # of passing cars to port ‘u-out, ‘i-out, ‘s-out, ‘r-out
```

Figure 13. Pseudo-code of the STREET model

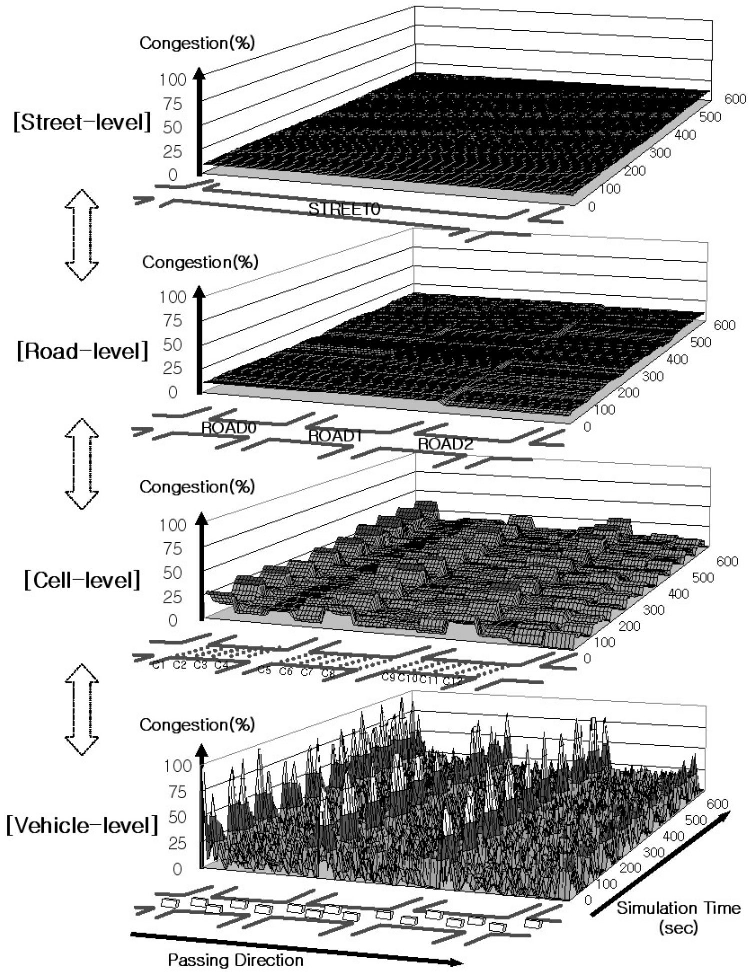


Figure 14. Abstraction-related simulation results: (a) normal case (continued on next page)

vehicle level (microscopic model), the cell level (macroscopic model), the road level (macroscopic model), and the street level (macroscopic model), which Figure 4 shows. The initial conditions of the test are as follows: simulation time is 600 sec, cell length is 200 m, road length is 800 m, and street length is 2400 m. The signal policy is as follows: cycle length is 100 sec, green split is 25 sec, and offset is 75 sec. In addition, the generation distribution about the vehicle is based on the assumption that it is generated on the basis of Poisson distribution, in which λ is equal to 23 sec. Figure 14a shows the results of simulation in a normal traffic flow from the vehicle level to the street level. Figure 14b shows the far-reaching effect of congestion based on the time when an accident has occurred from a collision between vehicle 12 and vehicle 34 at the vehicle level, given that the simulation time is equal to 120 sec.

In Figure 14, the vehicle level is expressed through the time-space diagram. The time-space diagram can describe behaviors of individual vehicles effectively as a picture according to the flow of discrete time. The time-space diagram expresses positions of individual vehicles with space as simulation time ticks by. These microscopic behaviors of individual vehicles at the vehicle level are abstracted to the cell level, which consists of macroscopic models that have occupancy of the space based on discrete time. Also, macroscopic models at the cell level are abstracted to the road level and street level easily by the aggregation morphism method. Like this, hierarchical expression according to abstract way about road networks offers merits and demerits of each level. The vehicle level has a strength in that it can know detailed information about when, where, how, and which vehicle is in the accident by depicting traffic flow

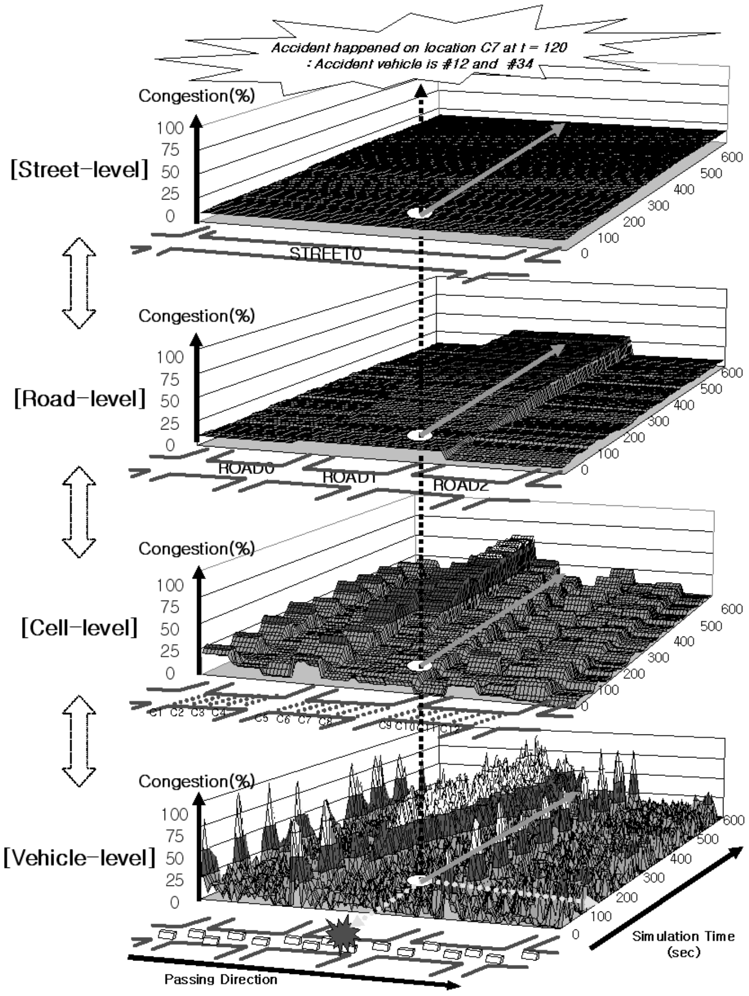


Figure 14. Abstraction-related simulation results: (b) abnormal case (continued from previous page)

status around individual vehicles, and it can also obtain detailed information about the form in which the status will evolve. However, it increases system overhead because it gives unnecessary information to users who need summarized and conceptual congestion information about a large road network. The street level, which is just below the abstraction level, provides useful information in making a transportation policy or a transportation plan by offering the administrator notional and comprehensive information about road nets. The administrator controls broad road nets, but this level's weakness is that it cannot express detailed information about individual vehicles. Thus, the abstraction method of the road network, according to the abstraction relation in traffic modeling and simulation, will be widely used because it is possible to express and analyze various requirements of the traffic analyst hierarchically and structurally, from detailed descriptions of individual vehicles to conceptual, general, and summarized descriptions.

7. Layer 4: ITS Simulation Systems

A simulation application system for ITS, such as ATIS and ATMS, can be accomplished through the abstraction method and traffic modeling based on the hierarchical modular modeling and simulation environment proposed earlier. First, a specification phase about the requirements for analysis and constraint conditions is needed. Detailed information is needed to make a library about structural/dynamic constraints of roads, the traffic signal control policy, linkage, and components of the ITS. Second, the database that includes the PES, the temporal database, the creation of structural/dynamic models, and the construction of a traffic information database are required. The PES consists of microscopic and macroscopic models obtained through SES and pruning. SES and pruning express road networks according to the abstraction methodology as a structural expression for traffic simulation. The temporal database contains simultaneous accidents on roads. Structural/dynamic models construct a database on all kinds of traffic information. Third, a simulation model structure of the area needs to be created by using various databases and the component library of the ITS made in the second step. When this is combined with dynamic models based on the abstraction methodology, a simulation performance phase is needed through building simulation models. Finally, various analyses (measure of effect, etc.) and expressions (2-D/3-D graphic animation, etc.) about simulation results obtained in the third step should be made. A graphic user interface is required to analyze and express simulation results for the convenience of users. Therefore, access to diverse application systems such as ATMS, ATIS, ETCS (Electronic Toll Collection System), BIS (Bus Information System), CVO (Commercial Vehicle Operations), and RGS (Route Guidance Service) offered in ITS by a method such as this will be possible.

7.1 Case Study 1: ATIS: The Next Generation

We have implemented the S/W package, called “ATIS: The Next Generation,” for ATIS by using hierarchical modular modeling and simulation, dynamic path searching, and other advanced techniques such as multimedia database and virtual reality. Our system differs from other ATIS systems in that it (1) supports traffic congestion simulation based on the macroscopic traffic models, (2) provides dynamic optimal path searching by using the forecasted traffic congestion dynamics, and (3) allows 3-D, graphics-based virtual driving along with the optimal path. Figure 15 partially shows screen copies of major features such as real-time monitoring with CC-TV, virtual investigation, virtual driving, congestion dynamics, main building retrieval, and so on. More detailed descriptions of our system are available in Chi et al. [28].

7.2 Case Study 2: I^3D^2 Transportation Simulation System

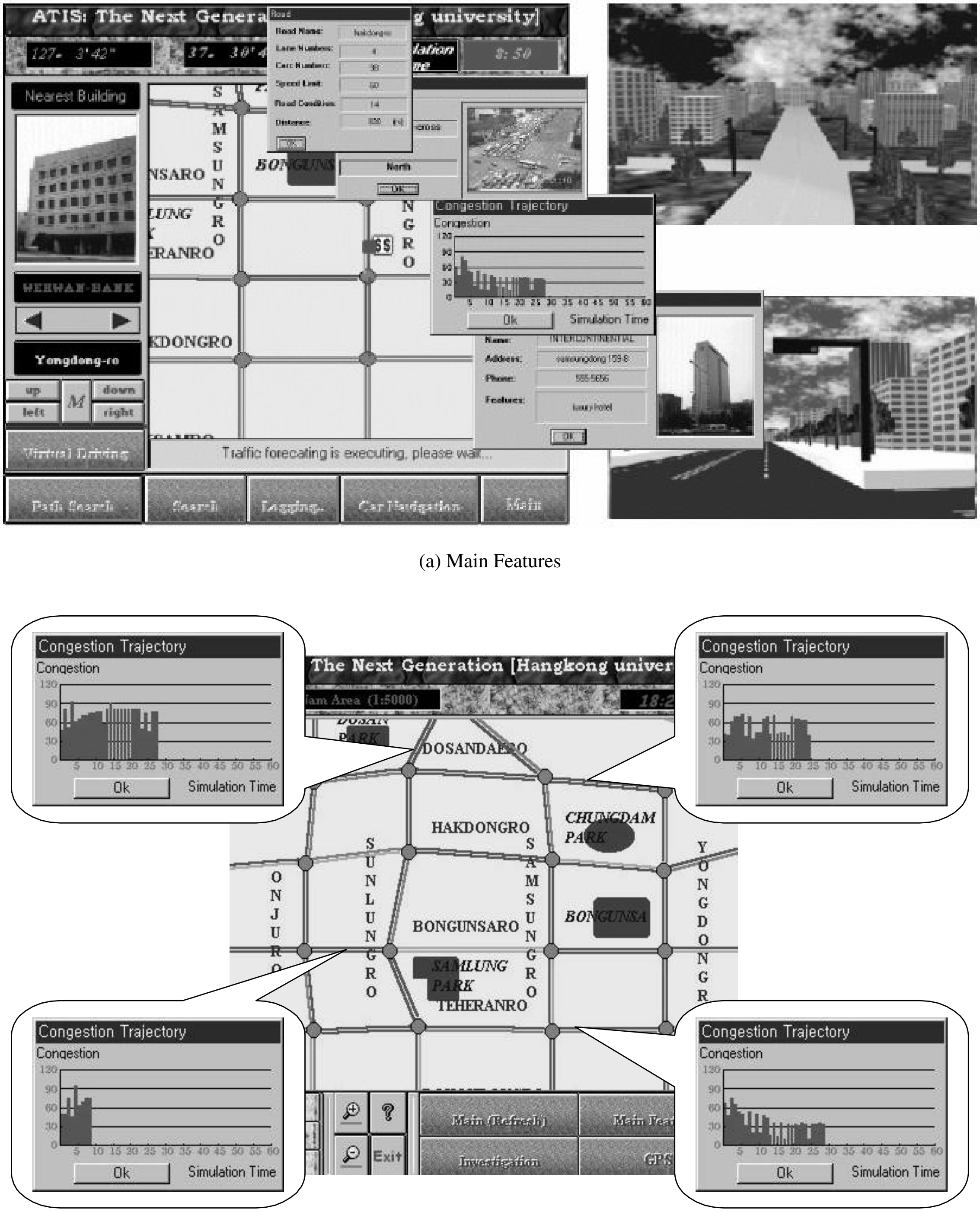
Based on the hierarchical modeling and simulation methodology for ITS, we have also developed the I^3D^2 Transportation Simulation System, in which I^3D^2 stands for intelligent, interactive, integrated, DEVS based, and distributed [29]. *Intelligent* means that it provides a knowledge-based modeling environment for consideration of the human factor (i.e., the driver’s behavior upon receiving the ATIS information). *Interactive* means that it supports the runtime interaction to modify the simulation condition and/or generate the simulation reports during the simulation. *Integrated* means that it allows an integrated modeling environment by combining the traffic dynamics with other transportation components such as driver’s behavior, signal control policy, sensory readings, and the transportation database. *DEVS based* means that it was developed on the basis of advanced hierarchical modular object-oriented techniques. Finally, *distributed* means that it supports a HLA-compliant distributed simulation environment. The I^3D^2 system is currently being developed and tested under the stand-alone environment. Several screen shots of the user interface of the I^3D^2 system are presented in Figure 16, in which the online analysis of each road, cross, signal, detector, and individual vehicle during the simulation is partially shown. Traffic models in the proposed environment were verified in that they express the real world effectively by using real data from the Korea Highway Corporation. The data consist of vehicles that drive a 32-km block from East Seoul T.G. to Hobup I.C. of the Joongboo highway in Korea [30]. Relative performance of the I^3D^2 system was also verified by comparing its performance with the TSIS [31] of the Federal Highway Administration in the United States [32].

8. Conclusions

The hierarchical and modular design methodology of the traffic simulation system for ITS has been discussed. To do this, we proposed the four-layered approach for traffic simulation based on the object-oriented programming environment: (1) the system entity structure/model base layer, (2) the model abstraction layer, (3) the traffic modeling layer, and (4) the ITS simulation systems layer. The vehicle-level, cell-level, road-level, and street-level modeling approaches have been developed by the DEVS formalism. ITS simulation methodology supports the intelligent, interactive, and integrated transportation simulation environment. Therefore, it provides a convenient means for evaluating the alternative signal control strategies at the operation level of the ATMS, as well as generating the simulation-based forecasting information for ATIS. Based on this methodology, ATIS: The Next Generation (for ATIS) and the I^3D^2 Transportation Simulation System (for ATMS) have been successfully implemented and tested. The abstraction method of the road network according to the abstraction relation in traffic modeling and simulation will be widely used because it is possible to express and analyze various requirements of the traffic analyst hierarchically and structurally, from detailed descriptions of individual vehicles in a road network to conceptual, general, and summarized descriptions.

9. Acknowledgment

The authors would like to thank Dr. B. J. Kim, K. C. Jung, and M. W. Lee, Hangkong University, and Dr. C. H. Park and Dr. K. S. Chun, Seoul National University. This research was partially supported by the Ministry of Science & Technology under contract 2000-N-Nl-03-A-02, “Development of Simulation System Architecture and Application Systems.”



Road Name: hakdongro
Lane Numbers: 4
Cars Numbers: 38
Speed Limit: 60
Road Condition: 14
Distance: 820 [m]

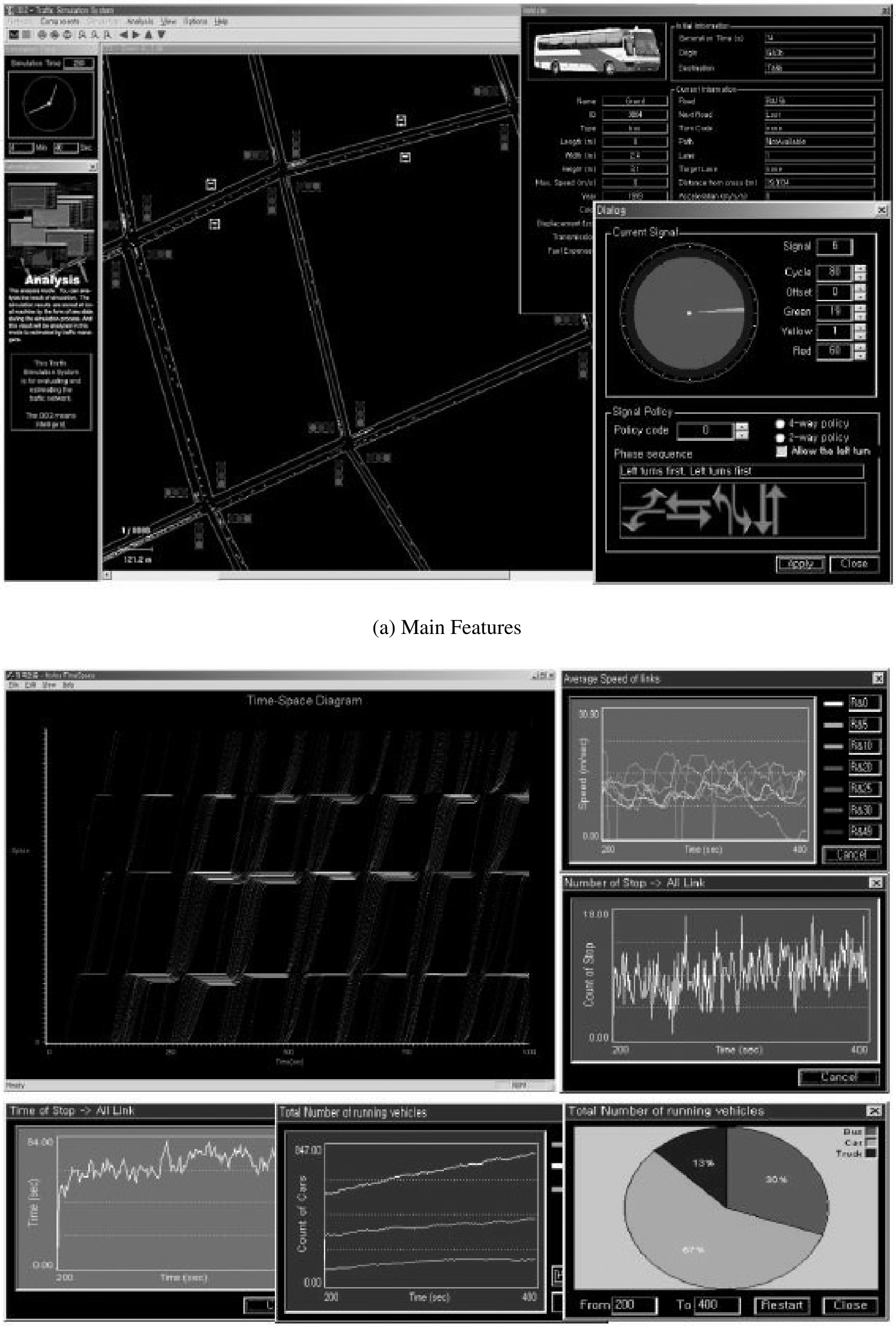
Name: INTERCONTINENTAL
Address: samsungdong 159-8
Phone: 555-5656
Features: luxury hotel

Traffic forecating is executing, please wait...

(a) Main Features

(b) Macroscopic Simulation Example

Figure 15. ATIS: The Next Generation



Current Signal
Signal 5
Cycle 80
Offset 0
Green 19
Yellow 1
Red 60

Signal Policy
Policy code 0
● 4-way policy
● 2-way policy
☑ Allow the left turn
Phase sequence
Left turns first, Left turns first

From 200 To 400

(a) Main Features

(b) Microscopic (vehicle-level) Simulation Example

Figure 16. I^{3}D^2 Transportation Simulation System

10. References

[1] Yang, Q. 1997. A simulation laboratory for evaluation of dynamic traffic management systems. Ph.D. diss., Department of Civil and Environmental Engineering, MIT, Cambridge, MA.

[2] Tritter, D. B., and J. Zietlow. 1993. Designing a traffic management communication system to accommodate intelligent vehicle highway systems. In *Proceedings of the IEEE-IEE Vehicle Navigation and Information Systems Conference*, October, pp. 182-5.

[3] Nelson, J. R., F. Spitzer, S. Stewart. 1993. Experiences gained in implementing an economical, universal motorist information system. In *Proceedings of the IEEE-IEE Vehicle Navigation and Information Systems Conference*, October, pp. 67-71.

[4] Simonsson, S. O. 1993. Car-following as a tool in road traffic simulation. In *Proceedings of the IEEE-IEE Vehicle Navigation and Information Systems Conference*, October, pp. 150-6.

[5] Mahmassani, H. S., T. Y. Hu, S. Peeta, A. Ziliaskopoulos. 1994. Development and testing of dynamic traffic assignment and simulation procedures for ATIS/ATMS applications. Report DTFH61-90-R-00074-FG. McLean, VA: U.S. Department of Transportation, Federal Highway Administration.

[6] Development of a deployable real-time dynamic traffic assignment system. 1996. Technical Report Task B-C, Intelligent Transportation Systems Program and Center for Transportation Studies, MIT, Cambridge, MA.

[7] Silbley, S. W. 1985. NETSIM for microcomputers simulates microscopic traffic flow on urban streets. *Public Roads* 49:54-9.

[8] Barcelo, J., and J. L. Ferrer. 1995. A simulation study for an area of Dublin using the AIMSUM2 traffic simulator. Technical report, Department of Statistics and Operation Research, Universitat Politenica de Catalunya, Spain.

[9] Niedringhaus, W. P., and P. Wang. 1993. IVHS traffic modeling using parallel computing. In *Proceedings of the IEEE-IEE Vehicle Navigation and Information Systems Conference*, October, pp. 157-62.

[10] Federal Highway Administration. 1994. FRESIM user guide. Technical report, version 4.5, Federal Highway Administration, U.S. Department of Transportation, McLean, Virgina.

[11] Gilmore, J. F., S. P. Roth, H. C. Forbes, K. A. Payne. 1995. ATMS universal traffic operation simulation. Paper presented at the annual meeting of the 5th IVHS America, Washington, DC.

[12] Abdulhai, B., J. B. Sheu, and W. Recker. 1999. Simulation of ITS on the Irvine FOT area using "Paramics 1.5" scalable microscopic traffic simulation: Phase I. Model calibration and validation. Research Report No. UCB-ITS-PRR-99-12, California PATH, University of California, Berkeley.

[13] Paramics Traffic Simulation Ltd. 1997. *Paramics v1.5 manuals*. Edinburgh, Scotland: Paramics Traffic Simulation Ltd.

[14] Deshpande, A. R. 1996. *SHIFT reference manual*. Berkeley: California PATH.

[15] Eskafi, F., and A. Göllü. 1997. Simulation framework requirements and methodologies in automated highway planning. *Transactions of the Society for Computer Simulation International* 14 (4): 169-80.

[16] Sargent, R. 1993. Hierarchical modeling for discrete event simulation. In *Proceedings of the Winter Simulation Conference*, p. 569.

[17] Zeigler, B. P. 1990. *Object-oriented simulation with hierarchical, modular models: Intelligent agents and endomorphic systems*. San Diego: Academic Press.

[18] Zeigler, B. P. 1984. *Multifacetted modeling and discrete event simulation*. London: Academic Press.

[19] Zeigler, B. P. 1989. DEVS representation of dynamical systems: Event-based intelligent control. *Proceedings of IEEE* 77:72-80.

[20] Luh, C. J., and B. P. Zeigler. 1993. Abstracting event-based control models for high autonomy systems. *IEEE Transactions on Systems, Man, and Cybernetics* 23:42-54.

[21] Zeigler, B. P., T. G. Kim, and H. Praehofer. 2000. *Theory of modeling and simulation*. 2d ed. New York: Academic Press.

[22] Cho, H. J. 1999. Discrete event system homomorphisms: Design and implementation of quantization-based distributed simulation environment. Ph.D. diss., Department of ECE, University of Arizona, Tucson.

[23] Zeigler, B. P., and Y. Moon. 1997. Abstraction methodology based on parameter morphisms. In *Enabling Technology for Simulation Science, SPIE AeoroSense 97*.

[24] Ijiri, T. 1971. Fundamental queries in an aggregation theory. *Journal of the American Statistics Association* 66:766-82.

[25] David, S. 2001. The application of microscopic simulation traffic models to traffic management [Online]. Available: www.paramics-online.com/tech_support/reports.htm

[26] Han, L. D. 1992. Traffic flow characteristics of signalized arterials under disturbance situations. Ph.D. diss., University of California, Berkeley.

[27] Ben-Akita, M. E., H. N. Koutsopoulos. 1994. Integrated simulation framework for evaluating dynamic traffic management systems. In *Proceedings of the 1st Conference on Application of Transport Telematics and IVHS*, December, Paris, France.

[28] Chi, S. D., Y. K. Kim. 1996. A role of discrete event modeling for advanced traveler information systems. In *4th AIS Conference of the Society for Computer Simulation International*, March, pp. 37-43.

[29] Chi, S. D., and J. K. Lee. 2001. DEVS-based modeling and simulation for intelligent transportation systems. In *Discrete event modeling and simulation: A tapestry of systems and AI-based theories and methodologies*, edited by Hessam S. Sarjoughian and Francois E. Cellier, 215-27, Springer-Verlag.

[30] Lee, J. K., S. D. Chi, and B. P. Zeigler. 2003. Predictive quantization based filtering for distributed traffic simulation. Paper presented at the 2003 Summer Computer Simulation Conference, July, Montreal, Canada.

[31] Federal Highway Administration. 1998. *Traffic software integrated system version 4.2 user's guide*. McLean, VA: Federal Highway Administration, U.S. Department of Transportation.

[32] Yoon, D. Y., J.R. Jung, J. K. Lee, S. D. Chi, and B. J. Kim. 2001. Performance analysis of I^3D^2 Transportation Simulation System and TSIS. Paper presented at the 2001 Autumn Conference of Korea Society of Transportation, November.

Jong-Keun Lee *is a senior researcher of Telematics Research Dept. at Cartronics R&D Center of Hyundai MOBIS, Seoul, Korea, from Sept. 2003. He received a Ph.D degree in computer engineering from Hangkong University, Seoul Korea, in 2002. From 2002 to 2003, he was a post-doctoral researcher at Arizona Center for Integratied Modeling & Simulation of University of Arizona, Arizona, USA. His main research interests are Driver Information System, Telematics, discrete-event modeling and simulation, intelligent system design.*

Ye-Hwan Lim *is a researcher of ROTIS Ltd. Seoul, Korea. He received BS and MS degrees in computer engineering from Hangkong University, Seoul, Korea, in 1999 and 2001.*

Sung-Do Chi *is a professor at the Intelligent Systems Research Laboratory in the Department of Computer Engineering, Hangkong University, Seoul, Korea. He received a Ph.D degree in electrical and computer engineering in 1991 from the University of Arizona, Tucson, USA. His research interests include traffic modeling, modeling based reasoning, intelligent system design methodology and network security modeling and simulation.*