Proceedings of the 1994 Winter Simulation Conference
ed. J. D. Tew, S. Manivannan, D. A. Sadowski, and A. F. Seila

A FRAMEWORK FOR GENERIC MODULAR ATMS/ATIS SUPPORT IN IVHS TRAFFIC SIMULATION

Paul T. R. Wang
Karl E. Wunderlich

The MITRE Corporation
McLean, VA

ABSTRACT

The authors present an interface to facilitate traffic simulation of Intelligent Vehicle Systems (IVHS) (Federal Highway Adminstration, 1991). Once a generic user interface is agreed upon and its standard calls established, reusable model building blocks that are portable across different traffic models could be developed. As a first step, parameter lists required for effectively simulating the Advanced Traveler Information Systems (ATIS) and/or the Advanced Traffic Management Systems (ATMS) are presented. These generic parameter lists facilitate the modular support of C-based control routines for both microscopic and mesoscopic traffic simulators. This coupling of vendor specific ATIS or ATMS control algorithms and architecture specification with existing traffic modeling tools, in this case the Integration and THOREAU traffic simulations, is a key capability in the initial evaluation and analysis of alternative IVHS architectures. This approach has been tested by The MITRE Corporation and is currently being used in the FHWA IVHS architecture study. This paper details the common data elements exchanged between a traffic simulator and user-coded IVHS algorithms to determine costs-benefits and measures-of-effectiveness (MOE) for a wide range of ATIS and ATMS implementation alternatives, such as market penetration, communication requirements, percent compliance, central vs. distributed route guidance, incident detection and recovery strategy. This set of common data elements could also provide a framework for establishing generic modular support across a wider range of traffic simulations than those selected for use in the IVHS architecture research effort.

KEYWORDS

Intelligent Vehicle Highway Systems, Object-oriented Traffic Simulation, Advanced Traffic Management Systems, Advanced Traveler Information Systems, Signal Optimization, Route Guidance, and User Interface.

1.0 THOREAU OVERVIEW

Traffic and Highway Objects for REsearch, Analysis, and Understanding (THOREAU) is the product of a 1991/92 MITRE-sponsored research project called Vehicular Traffic Analysis Capacity (McGurrin and Wang, 1991). The goal was to develop an object-oriented, trip specific, microscopic, and mesoscopic traffic simulation model. THOREAU was designed for comparative analysis of IVHS concepts such as alternative ATIS and/or ATMS algorithms, and architectural alternatives. THOREAU is coded in MODSIM II, an object-oriented simulation language developed by the CACI Products Company. Combining discrete event-driven simulation with an object-oriented approach, programming languages such as MODSIM II offer great improvements in flexibility, modularity, development speed, and compactness. MODSIM II provided an ideal platform for speedy development of THOREAU, which models vehicle-following maneuvers, Urban Traffic Control Systems (UTCS) intersection controllers, turn movements, lane shifts, and trip reroutes, along with a graphic animation capability.

As an object-oriented traffic simulation tool, THOREAU incorporates both arterial street and freeway objects such as intersection, lane, signal controller, detector, stop sign, and weaving areas. Four different classes of drivers are simulated: cautious, normal, rush, and ruthless. The characteristics of these driver classes are defined by a set of parameters that defines the population and relative aggressiveness. Three types of lane shift operations simulated are opportunistic, mandatory, and "good Samaritan" shifts. Incidents are simulated as scheduled events where obstacles are placed at predetermined spots for certain periods of time. Vehicle movements are simulated as discrete events governed either through microscopic headway and speed control or by an optional macroscopic flow-speed-density equation along link segments. The use of macroscopic flow equations while tracking individual vehicles is referred to as mesoscopic traffic simulation. The combined microscopic and mesoscopic approach provides THOREAU with the desired simulation speed for large scale traffic networks while retaining its granularity for selected links and nodes.

A generic user interface in C provides access to user-coded ATIS or ATMS algorithms for IVHS related studies. A modified Floyd algorithm (Wang and Niedringhaus, 1993) and the Webster and Cobbe algorithm (Webster and Cobbe, 1966) for signal optimization have been coded in C for use as examples of the generic user interface for IVHS related traffic simulation.

2.0 MODULAR ATIS/ATMS SUPPORT

Two aspects of IVHS technology are smart vehicle and smart roadway control, i.e., ATIS and ATMS respectively. To simulate ATIS or ATMS, a given user interface must provide data elements that define the traffic network, and its topology, including nodes, links, and connectivity, travel time, volume, capacity, and other route guidance related information. To simulate ATMS, the user interface must also provide data elements pertaining to global or local signal control and the reading or setting of signal control parameters, including the cycle time, phase splits, and interval timing. Once these data elements are identified and defined, a generic user interface can be implemented to facilitate the transfer of information between a traffic simulator and the procedures emulating specific ATIS or ATMS implementation or design. Such a generic user interface has been implemented in THOREAU for IVHS related simulation. In THOREAU, the data elements that are required by both ATIS and ATMS functions are passed as common arguments/parameters in arrays or matrices to both the ATIS and ATMS procedures. While data elements, such as link-to-link travel time, that are required for trip time computation in ATIS simulation, or the UTCS control parameters and detector readings that are required for signal timing optimization in ATMS simulation are passed only between THOREAU and the designated ATIS or ATMS procedure. Figures 2 to 4 illustrate the need of a generic modular ATMS/ATIS user interface for the cross-comparability of IVHS architecture and control algorithms impacts analysis. Such a modular user interface as procedures in C has been implemented and tested using the THOREAU object-oriented traffic simulation software.

Currently, only one user procedure, ProcATMS, has been implemented in THOREAU for the reading and setting of signal control parameters: Two ATIS user procedures, ProcATIS1 and ProcATIS2, are implemented in THOREAU for two different types of ATIS equipage. Data elements that are common to both the ATIS and ATMS user interface include the simulation clock time, network size, network connectivity, link attributes such as travel time, capacity, volume, and interlink travel time. For ATIS simulation, the ProcATIS1 or ProcATIS2 interface returns the computed alternative paths as a successor matrix that completely specifies the shortest paths for all link pairs. For ATMS simulation, major UTCS control parameters such as controller and detector Ids, link volume, link occupancy, cycle, phasing, and interval control are passed between THOREAU and the user coded C procedure via the ProcATMS interface. Both ATIS and ATMS user interfaces in THOREAU are accessed periodically with independent cycle times set by the user. Completed lists of all data blocks for the modular ATIS/ATMS user interfaces are given in the following parameter lists. Data blocks passed from THOREAU to ProcATMS or ProcATIS are labeled as IN while data blocks passed and returned from/to THOREAU are labeled as IN/OUT.

3.0 ATIS PARAMETER LIST

- time - IN Real number representing the current simulation time, in seconds.
- maxlinks - IN Integer determining the maximum number of links.
- maxnodes - IN Integer determining the maximum number of nodes.
- maxreal - IN Real constant with a value of 1010. This is used to represent an infinite link delay.
- linkids - IN 1-D array of integers that indicate the IDs of all links (size is maxlinks). This is used to associate the link ID with its index (order) in subsequent arrays. • fromnodes - IN 1-D array of integers that indicate the origin node IDs for each link (size is maxlinks).
- tonodes - IN 1-D array of integers that indicate the destination node IDs for each link (size is maxlinks).

- linkdirection - IN 1-D array of characters (S = South, N = North, W = West, and E = East) that indicate the direction of each link (size is maxlinks).
- linktime - IN 1-D array of real numbers determining the latest travel time for each link (size is maxlinks).
- linkvolume - IN 1-D array of integers determining the traffic volume in vehicles per hour of each link (size is maxlinks).
- linkcapacity - IN 1-D array of real numbers determining the capacities of each link (size is maxlinks).
- delays - IN 2-D array of real numbers specifying the latest link-to-link delay times (size is maxlinks x maxlinks).

Note: The next four arrays are used to relate the delay matrix (delays) to the successor matrix (paths), and are provided for the users convenience in construction of the network topology. The network can be built directly from the node and link files without these arrays.

- indegrees - IN 1-D array of integers that indicate the number of links (arcs) that are directed into (enter) each link (size is maxlinks).
- outdegrees - IN 1-D array of integers that indicate the number of links (arcs) that are directed out of (exit from) each link (size is maxlinks).
- inarcs - IN 2-D integer array specifying the indexes of links entering (branching into) each link (size is maxlinks x maxlinks).
- outarcs - IN 2-D integer array specifying the indexes of links exiting (branching from) each link (size is maxlinks x maxlinks).
- paths - OUT 2-D integer array specifying the successor matrix of the shortest path from link to link for each link (size is maxlinks x maxlinks).

4.0 ATMS PARAMETER LIST

- ndetectors - IN Integer determining the number of detectors.
- ddatasize - IN Integer determining the number of data fields for a detector.
- detectordata - IN 2-D integer array containing data for each detector (size is ndetectors x ddatasize).
- volume - IN 1-D array of integers that indicate the traffic volume for each detector (size is ndetectors).
- occupancy - IN 1-D array of integers that indicate the traffic occupancy for each detector (size is maxdetectors).
- ncontrollers - IN Integer specifying the number of controllers.
- cdatasize - IN Integer specifying the number of data fields for a controller.
- maxsubphases - IN Integer determining the maximum number of subphases.
- maxintervals - IN Integer determining the maximum number of intervals for each subphase.
- activesubphases - IN/OUT 2-D integer array containing the list of subphases in which a detector is active (size is ndetectors x maxsubphases).
- controllerdata - IN/OUT 2-D integer array containing data for each controller (size is ncontrollers x cdatasize).
- subphasesplits - IN/OUT 2-D integer array specifying subphase split data for each controller (size is ncontrollers x maxsubphases).
- subphasetypes - IN/OUT 2-D integer array containing the ordinal values of the subphasetypes for each controller (size is ncontrollers x maxsubphases).
- subintcounts - IN/OUT 2-D integer array containing the number of subphase intervals for each controller (size is ncontrollers x maxsubphases).
- nextsubphases - IN/OUT 2-D integer array specifying, for each controller, the ID of the next subphase for each of the controllers subphases (size is ncontrollers x maxsubphases).
- intervals - IN/OUT 3-D integer array containing each controllers subphase interval durations, in seconds (size is ncontrollers x maxsubphases x maxintervals).

Each of the numbers correspond, in order, to the following interval types: G, V, Y, R (see the array immediately below).

- controlcodes - IN/OUT 3-D character array contains each controllers subphase interval types (i.e. G, V, A, Y, R, etc.; corresponding to green, variable green, arrow, yellow, red, etc.). These types correspond to the interval durations in the array described immediately above (size is ncontrollers x maxsubphases x maxintervals).

5.0 AN EXAMPLE

As an example, a small urban traffic network URBCBD (Figure 1), was constructed using THOREAU to investigate the expected gain in vehicle-miles-total (VMT) and reduction in vehivle-hours-total (VHT) (Brand , 1993) with/without ATMS and/or ATMS tehnologies. An ATIS C-user interface implemented the modified Floyd algorithm for shourtest paths for all origin and destination (O/D) pairs; while an ATMS C-user interface implemented the Webster and Cobbe algorithm for individual signal controller. Initial analysis of the simulation results of the URBCBD model are presented. A total of 15000 trips were simulated for one hour period. The total reduction in VHT with/without the tested ATIS and/or ATIS algorithms are shown in Figure 5. Figure 6 shows the combined benefits of both intelligent route guidance and local signal optimization for various trip lengths. We have demonstrated that such a generic modular framework may be used effectively to study new IVHS related technologies and to obtain quantified MOEs in network capacity or travel time for different ATIS/ATMS implementation alternatives.

6.0 CONCLUSIONS

The separation of generic user coded procedures from an object-oriented traffic model serves two distinct purposes. First, it provides a common frame work to test different concepts or algorithms so that a meaningful comparison between two different approaches can be simulated prior to field tests. Second, it also provides a mechanism by which the traffic model can be independently validated since results can be checked against expectation or analytic prediction. In the first case, IVHS traffic engineers or designers can concentrate their effort on ATIS and ATMS aspects rather than facing the enormous task of managing their own microscopic or mesoscopic simulation with respect to vehicle following, network and paths setup, signal control, etc. In the second case, it provides a wide range of possible traffic scenarios and unpredictable conditions to test the robustness of a traffic simulation model.

ACKNOWLEDGMENT

This work was partially supported by the Federal Highway Administration under contract. The original vision of an object-oriented microscopic and mesoscopic traffic simulation model specifically designed for ATIS and ATMS applications and the development of the THOREAU model for IVHS studies were initiated and supervised by Mr. Michael F. McGurrin at the MITRE Corporation.

The URBCBD THOREAU Model

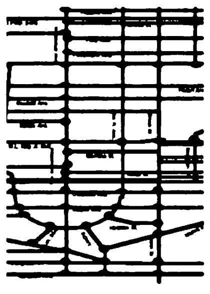


Figure 1 An Urban Traffic Network

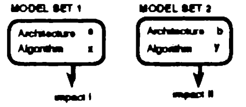


Figure 2 Impacts I and II Not Directly Comparable

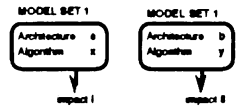


Figure 3 Impacts Comparable, Algorithms Unknowns

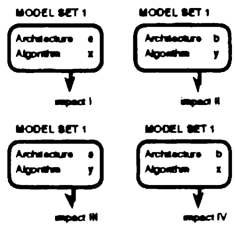
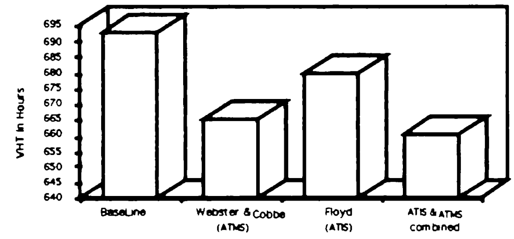


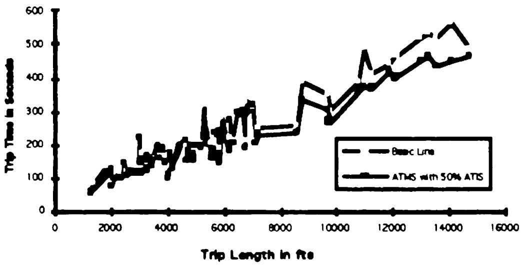
Figure 4 Cross-Comparability of Architectures and Control Algorithms

Figure 5 Vehicle Hours Total (VHT)



11419 Total Vehicle Miles by 13783 Trips

Figure 6 Trip Time Reduction



REFERENCES

Federal Highway Administration, "An Overview of the IVHS Program Through FY 1991," October, 1991, DOT, Washington, D.C.

McGurrin, M. F., and Wang, P. T. R., "An Object-Oriented Traffic Simulation with IVHS Applications,' 'VNIS'91 Conference Proceedings , Detroit, Michigan, September, 1991, pp. 551-561.

Hsin, V. J., and Wang, P.T. R., 1992. "Modeling Concepts for Intelligent Vehicle Highway Systems (IVHS) Applications", WSC'92 Conference Proceedings, Arlington, Virginia, December 1992, pp. 1201-1209.

Wang, P. T. R., and Niedringhaus, W. P., "Distributed/Parallel Traffic Simulation for IVHS Applications" WSC'93 Proceedings, Los Angeles, California, December, 1993 pp.1225-1230.

Webster, F. V., and Cobbe B. M. , "Traffic Signals," Road Research Tech. Paper No. 56, Her Majesty's Stationery Office, London, U.K., pp. 55-60, 1966.

Brand, D., "Criteria and Methods for Evaluating IVHS Plans and Operational Tests," Technical Paper 940989, Transportation Research Board, December, 1993.

AUTHOR BIOGRAPHIES

PAUL T. R. WANG graduated from National Cheng Kung Univeristy, Taiwan, China in 1966. He received the M.S. and Ph.D. degrees in Computer and Information Sciences from the Ohio State University in 1973. From 1973 to 1983 he was with the General Electric Company, serving as a consulting engineer at the Information Services Division in Rockville, Maryland. Since 1983, he has been with the MITRE Corporation. Currently, he is a Lead Engineer at MITRE's IVHS site located at Washington D.C. His interests include optimization, simulation, modeling, and queueing theory. He is a member of MAA.

KARL E. WUNDERLICH completed a B.S. degree in Industrial Engineering at Northwestern University in 1988. He received a M.S. degree in Industrial and Operations Engineering from the University of Michigan in 1990, and is currently a doctoral candidate completing a dissertation on predictive route guidance methods. From 1992 to the present he has been employed as a member of the technical staff at The MITRE Corporation's Intelligent Vehicle Highway Systems Group.