High Fidelity, Wide Area Traffic Simulation Model

Qi Yang
Caliper Corporation
1430 Hardy Court, McLean, VA 22101
tel: (703) 556-3401 fax: (703) 556-3402
email: qiyang@caliper.com

Howard Slavin
Caliper Corporation
1172 Beacon Street, Newton, MA 02461
tel (617) 527-4700 fax. (617) 527-5113
email: howard@caliper.com

ABSTRACT. This paper presents TransModeler, a comprehensive traffic simulation software for modeling vehiclar flow, traffic surveillance, and control systems. Using a geographic information system (GIS) based traffic network database, TransModeler can utilize existing network and travel demand data to simplify data preparation tasks required by traffic simulation applications. Its traffic flow simulation module allows microscopic and macroscopic traffic models to be applied to different parts of a network, providing modelers with the flexibility in handling large network and modeling operational details of selected sub-networks. The modular design of the software and its application programming interface allows the users to customize and extend its fucntions.

INTRODUCTION AND OVERVIEW

Despite the existence of a wide variety of traffic simulation models, an easy to use and integrated simulation tool that provides all the functionality needed by the traffic engineering, planning, and research community has been lacking. Planners, for example, often want to add traffic operations details to their studies done using traffic assignment based tools. However, existing simulation models are either too difficult and expensive to use, or cannot support networks the size of planning networks. Traffic operations analysts, on the other hand, are often mislead in their microscopic simulator-assisted impact studies because their tools fail to provide a system wide solution and depend too much on arbitrary input assumptions such as turning movements, which are subject to high variance. Another notable shortcoming of existing simulation models is that their software architectures are inflexible and preclude adding features required by technologies such as Advanced Traveler Information Systems (ATIS) and Advanced Traffic Management Systems (ATMS). Existing traffic simulators are also inflexible with regard to their input requirements and have unduly tedious data preparation and analysis requirements.

Caliper has been developing a new traffic simulation software package named TransModeler to address all of these deficiencies. This paper presents a brief overview of the functionality, software architecture, and principal applications for TransModeler.

TransModeler is an advanced traffic simulation software package with three design goals: comprehensive capabilities, ease of use, and extensibility. One of its key component modules, the vehicle mover, is a traffic flow simulator with hybrid fidelity. By hybrid we mean that the traffic simulator provides users the ability to simulate various parts of the network in the most detailed microscopic fashion while simulating other parts in less detailed but nevertheless adequate detail in terms of feeding and extracting traffic flow to and from the study area and deriving system wide measures of effectiveness. TransModeler is designed to support all the functionality commonly wanted by traffic engineers and planners and also to simplify significantly the tasks required for conducting advanced traffic simulation studies. The software also features TransModules™ – a set of Application Programming Interfaces (APIs) – to allow advanced users to extend the software beyond the capabilities provided in the package.

TransModeler is GIS-aware and compatible with most GIS and CAD packages. TransModeler can import road networks from ArcView, ArcInfo, AutoCAD (DXF), MapInfo, Maptitude, GIS+, and TransCAD. TransModeler can also import road networks and transit networks from transportation planning software including TranPlan, MINUTP, emme/2, Trips, and TransCAD. TransModeler can be fully integrated in a modeling chain with TransCAD–based travel demand forecasting models, thus achieving a complete linkage from trip generation to traffic operations on road networks. Network travel times can be returned to TransCAD for use in traffic assignment and mode choice models, and output data from TransModeler simulations can be viewed and analyzed in TransCAD.

Not only is TransModeler GIS-aware, but TransModeler directly employs new GIS technology to greatly simplify data preparation for traffic simulation. An innovative method has been devised that extends geographic information systems (GIS) technology to facilitate the management, analysis, and simulation of traffic data from road networks. Information such as traffic intersection characteristics, lane configurations and connections, and traffic signal settings can all be managed in a GIS environment and can be represented with a high degree of geographical accuracy.

The method by which this has been achieved has been to extend the conventional GIS data model of points, lines, and areas to include lanes and intersections. This is superior to previous schematic representations of these transportation features, which are neither geographically accurate nor can they be created from existing GIS data files. The geography for the lanes is generated dynamically and lane connectors are created from the starting and end points based on geography. Shape points and geometric curves are used to represent the road network accurately. Geographic editing tools can be used to change the traffic model network rapidly and easily.

Another innovation is that a geographic polygon overlay process is used to compute display regions for underpasses and overpasses. Polygon overlay is a process that identifies the areal intersections of polygons; when combined with elevation information, it can be determined which portions of the roadway are on top of other roads and thus the visibility region for the traffic simulation can be automatically computed and utilized.

Using GIS-based approach also allows more convenient data management and retrieval, automation of many data preparation tasks, and improved visualization of model results.

TransModeler is a completely new product developed from the ground up with a modular extensible architecture implemented with object-oriented programming and component object model (COM) techniques. The system is written in C++ and runs under Microsoft Windows 2000 and XP.

MODELING STRUCTURE AND FUNCTIONALITY

TransModeler is designed to support many different traffic simulation tasks faced by traffic engineers and planners, including construction impact studies, evaluation of geometric designs, identification of congestion hotspots, and evaluation of advanced, intelligent traffic management systems. The software is carefully divided into coherent modules. It consists of a traffic network database server, a set of traffic simulation modules, user interface components, and data filters. The generic structure of the software is depicted in Figure 1.

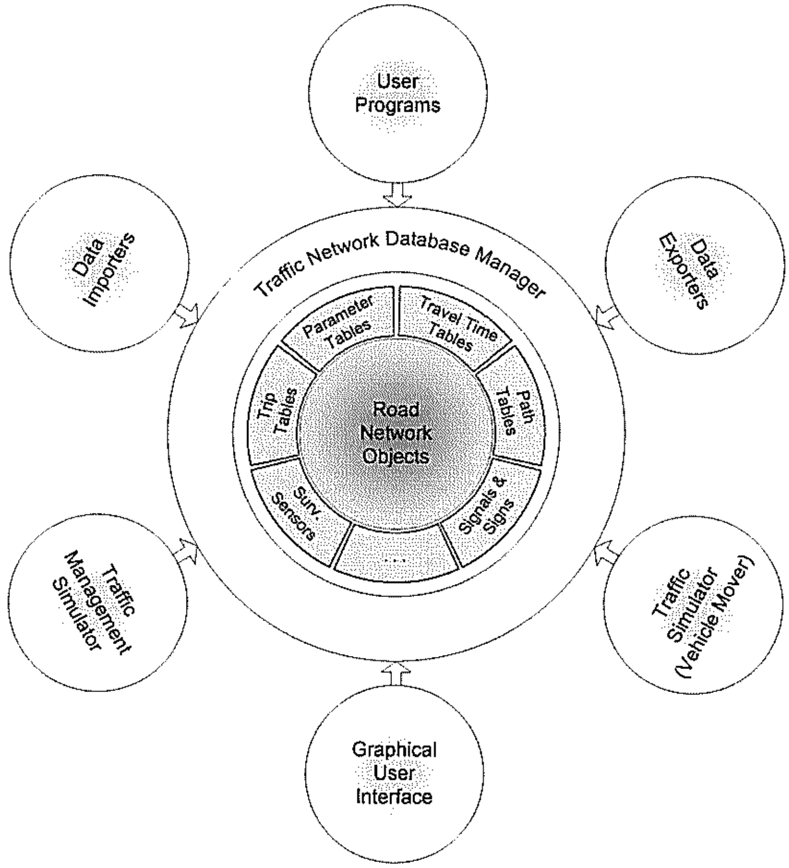

Figure 1 Overview of TransModeler's Software Structure

At the core of TransModeler is the Traffic Network Database. This traffic network database contains the records of road network objects; an inventory of surveillance sensors, traffic signals and signs; trip tables that represent time-variant travel demands between various origin-destination (OD) pairs; path tables that explicitly list the routes habitually taken, or models and rules that determine such routes; parameter tables that represent traffic characteristics and travel behavior; and travel time tables that describe the perceived historical and real-time travel time in the network.

The Traffic Network Database Manager (TNDM) is the engine that drives TransModeler software. It is a server that provides TransModeler modules, as well as foreign modules, services for manipulating the network and traffic objects. The TNDM is responsible for initializing, caching, indexing, searching, locking, and unlocking network and traffic objects and synchronizing their states.

TransModeler modules, built on the TNDM, provide the default functions of the system including traffic management, vehicle movement, graphic user interface, data import and export, and interface to user customized programs.

The Traffic Management Simulator (TMS) mimics operations of the traffic management system under evaluation and determines the states of traffic signals and signs. The Vehicle Mover (VM) simulates vehicles' movements, including the operation of transit vehicles, in the network and feeds the TMS with "real-time" traffic data. The Graphical User Interface (GUI) is the front end for managing the database objects and interactively visualizing the simulation process.

Several Data Importers are used as input filters to communicate with GIS software and transfer existing data sources into the TransModeler database formats. Data Exporters and Post Processors are used as output filters to provide customized output and to analyze the simulation results. Charting, tabulating, and reporting utilities are provided to facilitate analysis of model output.

TransModeler provides a rich Application Programming Interface (TransModules^{TM}) to assist users in customizing the modules, extending their functionality, and adding new modules into the systems. Users can write programs in C++ or VB and use the objects and methods provided in TransModules to manipulate network objects, access surveillance sensor data, manage the state of traffic signals and signs, and override the default travel behavior and traffic models used in the simulation.

These modules communicate with each other directly using Components Object Model (COM) or indirectly through the traffic network database. Some of these modules are further divided into smaller components that may be customized or replaced by the user's own modules. For example, VM includes components such as Vehicle Router and Trip Maker. The vehicle router calculates pre-trip and en-route path selection based on drivers' behavioral parameters and availability of "real-time" traffic information. The default vehicle router may be replaced at the run-time by a user implemented module. The Trip Maker is reponsible for generating vehicles using time-variant OD matrices or link traffic counts as input. TransModeler uses TransModules^{TM} and COM technology to make "plug-and-play" adaptability and extensibility a key feature of the software. By adopting an open architecture, TransModeler is particularly useful for researchers who want to use TransModeler as a testbed to design and test new dynamic traffic assignment methods, travel behavior models, traffic control algorithms, and data collection methods before they emerge into the real or simulated world.

TransModeler is not just another traffic simulator. Users are not restricted or required to use the simulation methods provided in the package. Instead, TransModeler can be utilized as an environment for hosting and using other traffic simulators and models if desired.

In the following section, we briefly describe the characteristics and capabilities of TransModeler's main components.

TRAFFIC NETWORK DATABASE AND MANAGER

The network used in TransModeler is usually imported from GIS data files. Network data files used by several commonly used simulation tools such as CORSIM and Integration can also be imported into TransModeler using its input filters.

Based on network geometry, topology data, and travel demand inputs, TransModeler builds its traffic network database. Since in many cases existing data sources may not provide enough details needed for simulation at a microscopic level, TransModeler's Network Importer can make educated guesses in providing the default values for the missing data items such as lane connections, segment attributes, usable paths, and so on. The imported or "generated" data items can be further modified interactively in TransModeler using its network editing tools.

TransModeler's network database is a collection of records for network objects and a description of their relationships. Most basic attributes and relationships of the network objects are stored in the database files, with the ability to join with external tables to access and manage customized data items. Network objects are organized hierarchically into records of nodes, links, segments, and lanes. Lane connections, lane-use privileges (ETC, HOV, etc.), regulation of turning movements at intersections, surveillance sensors, traffic control devices, toll plazas, and transit stops can all be adequately represented.

Intersections (nodes) in TransModeler are defined as either microscopic or macroscopic. Road segments between two nodes are defined as microscopic segments or macroscopic segments, depending on the type of nodes that they connected to or from. In microscopic segments, vehicle movement is simulated in detail using the car-following and lane-changing models; while in macroscopic segments, vehicle movements are simulated using speed-density functions based on macroscopic traffic models.

A variety of surveillance systems can be represented in TransModeler. The surveillance sensors are classified into three categories – point data sensors (e.g. loop detectors), point-to-point sensors (e.g. vehicle to roadside communication devices), and area-wide data sensors (e.g. CCTV) – based on the ways that data are collected and reported. Sensors are represented by their technical capabilities such as operational status, type of data collected, and measurement error.

TransModeler supports the simulation of a wide range of traffic control and route guidance devices, including intersection traffic signals, yield and stop signs, ramp meters, lane use signs, variable speed limit signs, portal signals at tunnel entrances, and variable message signs. Traffic control devices have visibility parameters that determine where vehicles may start responding to them and parameters that define the probabilities that drivers comply with the information provided by these devices.

TransModeler allows any number of incidents to be specified in the simulated network. An incident may completely block one or more lanes and/or produce a rubbernecking effect where vehicles slow down to a particular speed. Incidents are also characterized by their duration (clearance time), which may depend on the detection delay and response plans of the TMS.

TransModeler uses time-dependent origin to destination (OD) trip tables to represent the travel demand. OD trip tables can be specified individually for each vehicle type or, alternatively, the simulator can randomly assign a type to the vehicles based on a global fleet-mix. Vehicle type is a combination of vehicle class (e.g. high or low performance cars, buses, trucks, trailer trucks, and so on), lane-use privilege (e.g. HOV or ETC lane), access to information (e.g. uninformed, informed, and the way they use information), and driver behavior (e.g. aggressiveness and compliance). In order to support transit and special vehicles, a trip table enlisting individual vehicle departures and their paths can also be used.

Pedestrians and transit passengers are part of trip tables in TransModeler. The attributes of these data records are handled by the transit component in the Vehicle Mover module. For example, the waiting time and in-vehicle time can be recorded for each of the simulated individual trips by transit.

TransModeler maintains two sets of time-variant travel time information for every link in the network. Historical travel times, usually estimated from sensor data or coming from the output of planning models such as user equilibrium assignment, are used to assign vehicles to their habitual routes. Real-time travel times are updated periodically by TMS. For sophisticated ATIS/ATMS systems, for example, predicted travel times can be used. The updated travel times are used to calculate the pre-trip and en-route path choice probabilities for trips that are designated to access the real-time traffic information.

The traffic network database is a common link shared by all TransModeler modules. These modules read and/or write databases through the TNDM. Network objects are loaded into memory on demand and unloaded when they are out of operation scope or reach predetermined caching thresholds. The TNDM caches and streamlines the lifespan of various network objects in memory to make best use of the available computer resources and to allow simulation of large scale networks.

Most network objects are serialized with unique IDs. Input and output filters and external programs use these IDs to exchange data with TransModeler or directly update the TransModeler network database using the TransModules API. The data exchange between TransModeler and external programs can use either the binary format supported by TransModeler or the XML format defined by TransModules™ data type definitions (DTD).

TRAFFIC MANAGEMENT SIMULATOR

The traffic management simulator (TMS) mimics the traffic control and information systems under evaluation. The TMS simulates the operations of a wide range of traffic control and advisory devices, including intersection signal controls, ramp metering, and mainline lane control systems.

The simulation of traffic control and management in TransModeler is implemented in two tiers, each of which needs a different level of effort from the modeler:

- Signals and signs in TransModeler are controlled by 3 types of basic controllers, namely: pre-timed, traffic adaptive, or metering controllers. These controllers can be cast to represent a vast majority of traffic signal operations.

- Interfaces to specific signal control strategies can be implemented using TransModules API as derived controllers.

Interfaces to several commonly used Urban Traffic Control Systems and a few ramp metering algorithms and mainline lane control algorithms are included in the standard package.

Non-signalized intersections use stop and yield signs to establish the priority rules that govern the right of way for different movements.

Pretimed controllers are specified by an offset and a timing table, which consists of a set of phases and control intervals. A control interval represents a period of time during which states of all signals remain constant. The data items describing a control interval are its duration and a vector of signal states, which specify the right-of-way for various turning movements.

Adaptive controllers use real-time data from surveillance sensors and pre-specified control logic. Depending on the particular signal system, its control logic may be a special case of the basic adaptive controller, or coded as a customized controller module to interface with TMS. The modular design and TransModules API facilitate the easy addition of new types of controllers into the system.

The basic adaptive controller contains three sets of data records: (i) signals; (ii) phases; and, (iii) detectors.

The signal records prescribe the maximum red times, and the phase to be called next in the event that continuous red time for that signal has reached its maximum value.

The phase records represent the timing data and control sequence as in pretimed controllers. However, for adaptive controllers, a phase can be either extendible, callable, or both. A phase is extendible if its green interval can be extended when detector data satisfies certain criteria and no conflicting movement has reached its maximum red time. A phase is callable if, after completion of the current phase, signal operations can be shortcut to this phase without completing the subsequent phases in the cycle.

The detector records specify the logic for extending the current phase and calling a new phase. A controller may contain any number of detector records, each corresponding to a single detector. These records contain flags that specify the conditions for extending or calling a particular phase.

Ramp and mainline metering can be represented by either pretimed or adaptive controllers. TMS uses "desirable network states," such as occupancy at given locations, to compute the signal timing table. The desirable network state can either be predetermined or set dynamically by external control modules. A metering rate can also be based on changes in the inflows and outflows at given locations defined by two sets of upstream and downstream detectors.

Several freeway incident detection algorithms and a rule-based incident management scheme that influences the state of lane control devices are also implemented in TMS as optional components.

A TRAFFIC SIMULATOR WITH HYBRID FIDELITY

The Vehicle Mover (VM) module in TransModeler uses a time-based simulation logic to move vehicles in the network, either from their origins to destinations based on the path determined by Vehicle Router, or from link to link based on intersection turning movement tables if the input demand is not given in OD format. The length of time step and the logic for updating vehicles' state and position depends on whether the vehicles are in microscopic or macroscopic segments.

- In microscopic segments, vehicles calculate their acceleration rates and make decisions on lane changes according to car-following, lane-changing, event and signal response logic. The time step in this case can be as small as 1/10 second. In microscopic segments, vehicles' position, speed, acceleration, lane changes, and a number of other state variables are tracked in detail.

- In macroscopic segments, vehicle positions and destinations are still tracked but speed-density functions associated with the segment are used to calculate their average speed as a group. The step size for speed and position updating is usually 1 second. Capacity constraints and queue spillbacks at intersections are considered by introducing the traffic stream concept. A traffic stream is a group of vehicles that move to the same downstream link. Traffic streams that share the same segment impede each other, but may move independently (at a lower speed) as permitted by the downstream traffic condition and state of traffic signals.

TransModeler's VM module utilizes findings from the latest research on driver behavior and traffic flows and provides the tools for testing and evaluating models derived from such studies.

Each trip in TransModeler is tracked individually during the simulation. Trip departure times, origins, destinations, paths, and travel times can be individually logged. Dynamic states such as vehicle position, speed, acceleration, and lane changing can also be monitored and manipulated by programs that communicate with the simulator at run time.

In order to support ATIS applications and simulation of transit operations, the VM module in TransModeler is designed as a "path-based" simulator, as opposite to the "intersection turning proportion-based" approaches found in many other simulators and also supported in TransModeler. Each vehicle trip in TransModeler is associated with a record in the traffic network database. A vehicle moves along a path connecting its origin and destination according to appropriate traffic models. The initial path is chosen from the set of habitual routes for the OD pair. However, en-route decision-making can occur when unexpected events such as an incident are observed or when real-time (or predicted) traffic information is accessed. Both the pre-trip and en-route decisions are made according to a set of complex route choices models, implemented in a component module named Vehicle Router.

Traffic performance functions are used to calculate the speed of the traffic cells and streams in macroscopic segments. TransModeler's network database maintains an inventory of traffic performance functions and their parameters such as free flow speed, saturation flow rates, and speed-density relationships. Each macroscopic segment is associated with one of these functions, whose parameters can be estimated and updated using field (or simulated) detector data.

Vehicle movements in microscopic segments are modeled at a greater level of detail. Each vehicle and driver pair is assigned values for a number of behavior parameters based on given distributions provided in the input data file. This enables TransModeler's microscopic traffic simulation to capture a rich set of driver behaviors, including driver's desired speed, compliance with traffic regulations, acceptable gap threshold for lane changes and merging, willingness to yield to others, and vehicle's acceleration and speed profiles. Lane transitions at intersections and lane changes are classified into two categories - mandatory or discretionary. Mandatory lane changes and transitions apply to cases in which a vehicle has to change lanes and make the connection in order to stay on its path, bypass a lane blockage downstream, avoid using a restricted lane, or respond to traffic signals and signs. Discretionary lane changes refers to cases in which drivers change lanes in order to improve their driving experience. The decision to seek a discretionary lane change depends on the vehicle's speed, the difference in traffic conditions between the current and adjacent lanes, driver's desired speed, etc.

Once a simulated driver has decided to change lanes, the lead and lag gaps in the target lane are examined to determine whether the desired lane change can be executed. If both the lead and lag gaps are acceptable, the desired lane change is executed in steps. Importantly, a vehicle may occupy both lanes until the lane change is completed. The minimum acceptable gaps and the length of lane changing interval take into account the speed of the subject vehicle, speed of the lead and lag vehicles, remaining distance, and whether the lane change is mandatory or discretionary.

Probabilistic courtesy yielding and forced merging are explicitly considered to ensure the traffic flows are simulated realistically even in very congested scenarios.

The VM module is also responsible for updating the states of the surveillance sensors. Point sensor data, such as traffic counts, occupancies, and speeds, are reported at a fixed frequency. Sensor ID and vehicle information such as vehicle ID, type, speed, etc., are reported each time a probe vehicle passes a point-to-point sensor. Snapshots of data can also be collected periodically for vehicles within the detection zone of area data sensors.

USER INTERFACE

TransModeler offers an intuitive and user friendly graphical user interface (GUI). It can also be run in a windowless mode and be called in a script (using application executables) or other programs supporting COM (using dynamic link libraries). The GUI allows users to specify interactively the inputs to the simulation, change the simulation parameters, edit the network objects, and choose the desired output. The GUI also provides animation of the simulation process:

- The road network can be color coded and/or labeled by facility type, density, speed, flow, or other attribute and state of road segments. Dynamic information (e.g. density and speed) is updated at a user specified frequency.
- Surveillance sensor measurements (e.g. counts, speeds) are displayed and refreshed at fixed time intervals.
- The state of traffic signal and signs are displayed by dedicated symbols.
- Vehicle movements are animated and information such as vehicle type, car-following and lane-changing status are selectively displayed when the zoom scale is sufficient.

TransModeler uses a geographically accurate map to display output. Many types of images and aerial and satellite photography (TIFF, JPEG, BMP, SPOTView, USGS Digital Orthophoto, etc.) can be registered as map layers in TransModeler and used as the background for map display.

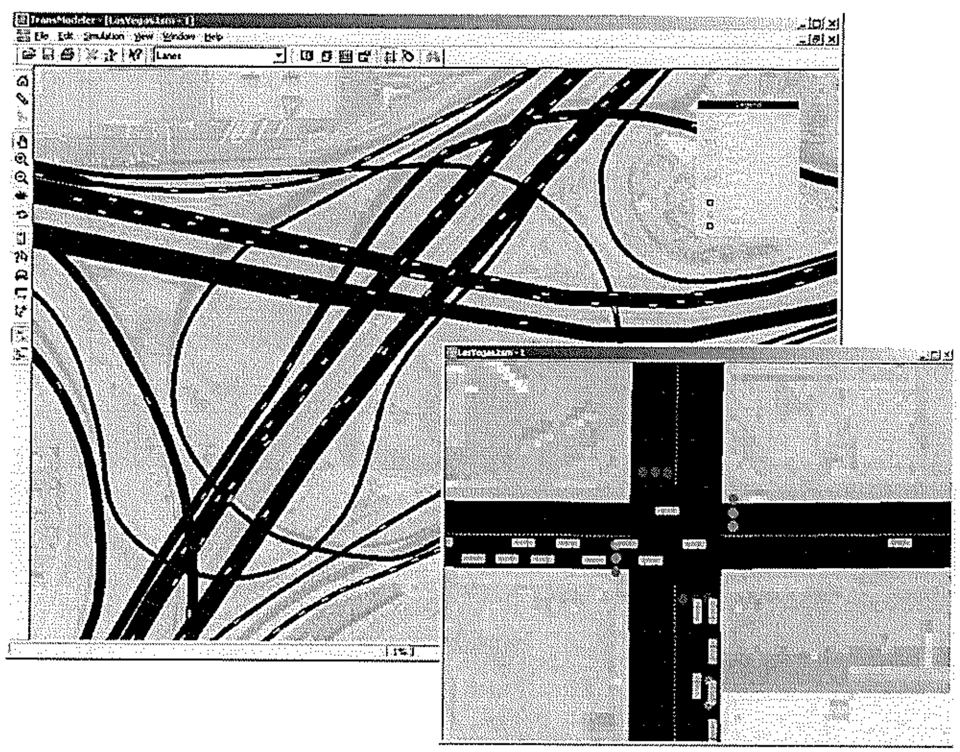

Figure 2 Animation of Vehicle Movement and Signal Controls

The editing tools in TransModeler allow modification, addition, and deletion of network objects (e.g. nodes, links, segments, lane connectors, surveillance sensors, traffic signals and signs, and so on) interactively. Each type of network object can be selected and customized as a layer in a map view and can also be shown and edited in a tabular data view. TransModeler's GUI implements a single document, multiple view interface. Multiple map views can be opened to see the different parts of the network, color code different traffic variables, and/or zoom (i.e. scales) to different levels of detail. The current object being browsed in a tabular data view can also be synchronized and highlighted in the active map view.

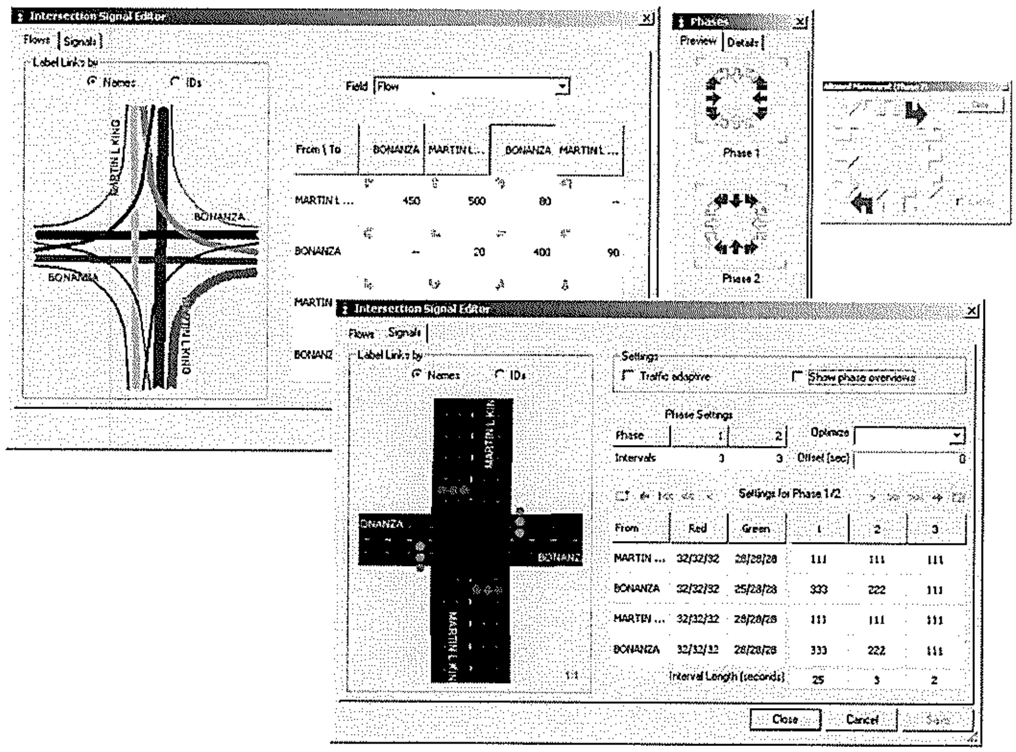

Figure 3 Intersection Traffic Signal Editor

Bar or line charts for a selected variable of several network objects or several variables of a single network object can be displayed in a chart view with the x-axis being the simulation time.

All types of views, including maps, tables, and charts, are treated as observers of the traffic network database and updated dynamically during the simulation according to user-chosen frequencies. External modules programmed using the TransModules API may also be hooked to the traffic network database manager to send or receive the updates to the state of road network objects.

TRANSMODELER SOFTWARE

TransModeler is programmed in C++ and runs under Microsoft Windows 2000. TransModeler is developed and supported by Caliper Corporation, a worldwide leader in transportation and GIS software. A wide array of development, customization, application, and training services are available from Caliper for simulation with TransModeler. For more information, contact Caliper Corporation, 1172 Beacon Street, Newton, MA 02461-9926, Tel: (617) 527-4700, Fax: (617) 527-5113, Email: info@caliper.com.

CONCLUSION

TransModeler is an advanced traffic simulation software package that supports a broad range of traffic analysis and modeling protocols. Its traffic simulation module allows the use of both microscopic and macroscopic traffic models for different parts of a network. The use of hybrid fidelity in modeling traffic flows provides modelers the flexibility to handle very large networks in a realistic fashion and to study ATMS strategies for regions.

TransModeler is also GIS-aware in that the geometry of the road segments are represented by their geographical coordinates. Utilities and importers have been developed to convert network data from available GIS data sources, such as street databases and transportation planning networks, into a traffic network database suitable for traffic simulation studies. TransModeler's traffic network database is designed to efficiently handle not only microscopic traffic simulation, but also other traffic analysis and data inventory applications. The advanced GIS capabilities of TransModeler greatly reduce the cost and complexity of data preparation for traffic simulation as well as improving its accuracy.

The user graphical interface of TransModeler is intuitive and easy to use. Initial cases studies indicate that both the microscopic and macroscopic simulation of traffic flow perform well in a variety of traffic scenarios. Many of the models and parameters used by TransModeler are drawn from previous research; however, application of the model in any real world project still warrants model calibration and validation.