

---

**Innovations Deserving  
Exploratory Analysis Programs**

***Intelligent Transportation Systems Program***

---

## **Access Traffic Simulation Model (ACTSIM)**

Final Report for ITS-IDEA Project 78

Prianka Seneviratne, Utah State University, Logan, UT

***November 2001***

## **INNOVATIONS DESERVING EXPLORATORY ANALYSIS (IDEA) PROGRAMS MANAGED BY THE TRANSPORTATION RESEARCH BOARD (TRB)**

This investigation by Utah State University, Logan, UT was completed as part of the Intelligent Transportation Systems (ITS) IDEA program which fosters innovations in development and deployment of intelligent transportation systems. The ITS-IDEA program is one of the five IDEA programs managed by the Transportation Research Board (TRB). The other four IDEA program areas are: Transit-IDEA, which focuses on transit practice in support of the Transit Cooperative Research Program (TCRP), NCHRP-IDEA which focuses on highway systems in support of National Cooperative Highway Research Program, High Speed Rail-IDEA (HSR), which focuses on high speed rail practice, in support of the Federal Railroad Administration, and Transportation Safety Technology (TST), which focuses on motor carrier safety practice, in support of the Federal Motor Carrier Safety Administration and Federal Railroad Administration. The five IDEA program areas are integrated to promote the development and testing of nontraditional and innovative concepts, methods, and technologies for surface transportation systems.

For information on the IDEA Program contact IDEA Program, (phone: 202/334-3310, fax: 202/334-3471, <http://www4.nationalacademies.org/trb/dive-idea.nsf>)

<p>The publication of this report does not necessarily indicate approval or endorsement of the findings, technical opinions, conclusions, or recommendations, either inferred or specifically expressed therein, by the National Academy of Sciences or the sponsors of the IDEA program from the United States Government or from the American Association of State Highway and Transportation Officials or its member states.</p>
---

## **ACKNOWLEDGEMENTS**

Several groups and individuals contributed to the success of this project, to whom the present author is extremely grateful. In particular, Brent Seeley of the Transportation Systems Lab at Utah State University deserves special recognition for his innovative ideas and perseverance. He was the principal force behind system design and programming. The IDEA Project Manager, Keith Gates must be specially thanked for being always available for guidance and providing opportunities for disseminating product information. Finally, the author wishes to his appreciation to the Wasatch Front Regional Council, Salt Lake City International Airport, and Leigh Fisher Associates, that believed in the product, supported it from the outset, and are working with the research team to implement the product.

## TABLE OF CONTENTS

<a href="#"><u>ACKNOLEDGEMENTS</u></a> .....	2
<a href="#"><u>EXECUTIVE SUMMARY</u></a> .....	4
<a href="#"><u>INTRODUCTION</u></a> .....	6
<a href="#"><u>IDEA PRODUCT</u></a> .....	7
<a href="#"><u>CONCEPT AND INNOVATION</u></a> .....	8
<a href="#"><u>INVESTIGATION</u></a> .....	9
<a href="#"><u>DATA INPUT MODULE</u></a> .....	9
<a href="#"><u>SIMULATION AND ANIMATION MODULE</u></a> .....	11
<a href="#"><u>Vehicle Dynamics</u></a> .....	11
<a href="#"><u>Vehicle Steering</u></a> .....	15
<a href="#"><u>Computation</u></a> .....	15
<a href="#"><u>GRAPHICS AND POST PROCESSOR MODULE</u></a> .....	16
<a href="#"><u>POST PROCESSOR MODULE</u></a> .....	16
<a href="#"><u>RUNNING ACTSIM</u></a> .....	17
<a href="#"><u>PLANS FOR IMPLEMENTATION</u></a> .....	20
<a href="#"><u>CONCLUSIONS</u></a> .....	21

## EXECUTIVE SUMMARY

Adequacy and reliability of ground access are key parameters that determine throughput and competitiveness of intermodal transportation terminals. As demand increases, maintenance of adequate capacity and a high level of reliability becomes a challenge for planners who are saddled with financial, physical, and environmental constraints. Traditional rules-of-thumb and most analytical tools including commercially available software tools do not lend themselves to analyzing alternative solutions to the problems that planners are facing.

Simulation models can emulate traffic flow on the road networks within intermodal terminals at the macroscopic or the microscopic level. However, compared to highway and arterial simulation models, intermodal terminal simulation tools are few. The search undertaken as part of the present project revealed only two airport curbside simulation models. There are also few reported airport applications of microscopic models such as Traff-Netsim and PARAMICS. A discrete-event simulation model has been developed for rail freight terminal planning. In addition, there are reports of a model to assess the impact of different intermodal rail terminals on delays to trucks and loading space requirements as well as a tool for seaport landside simulation.

Access Traffic Simulation Model (ACTSIM), the IDEA product created by this project is a tool for evaluating alternative internal road layouts and traffic management strategies under variable demand conditions and different service area configurations. It uses rule-based dynamics (or driver behavior logic), comprised of a specially developed parking algorithm as well as advanced car-following and lane-changing logic, to realistically animate vehicle movement and parking in service areas. Object-oriented modeling (OOM) technique used in ACTSIM allows a service (parking) area to be examined as a composition of several objects and object types. It comprises of four *modules*.

ACTSIM *network editor module* is used to construct the roadway, intersections and other features. The *input module* reads descriptions of the access road and service area from data files, and stores them as appropriate variables. The *simulation and animation module* is comprised of separate sub-modules to generate vehicles, assign characteristics to the vehicles, record the positions of vehicles, display (animate) the activities, and compute performance measures. The fourth module is the *post-processor module*, which compiles the simulated data, computes basic statistics, and creates data files for further analysis.

ACTSIM was calibrated and verified using data from Dulles Airport in Washington, DC. Testing was done in collaboration with TransCore, a transportation-consulting firm based in Virginia. TransCore provided extensive data on traffic flow (demand), vehicle assignment to attractors, access road system layout, vehicle dwell times and classifications as well as video footages of the operations. The results were

extremely satisfactory. Strong indications of its accuracy and user-friendliness have made it attractive to potential users. It has been well received by the consulting community as well as airport authorities. The reason being that ACTSIM is one of the few models available today for planning and managing intermodal transportation terminal access road networks, which is based on advanced programming tools, proven traffic flow principles, and state-of-the-art simulation and animation concepts. At the time writing this Report, there have been approximately a dozen enquiries about ACTSIM. Of those, four firm commitments have been received for implementation. That is a powerful indication of the potential and need for simulation and animation tools in transportation planning.

## INTRODUCTION

Microscopic simulation techniques are rapidly becoming indispensable in transportation systems planning design and operation. Their growth has been fuelled by the rapid expansion of intelligent transportation systems (ITS) applications, advanced computing and programming tools, and indeed by the refinements to traffic flow models based on hydrodynamics including the introduction of physical models.

Simulation is now used to emulate traffic flow on urban road networks as well as traffic flow within intermodal terminals, at the macroscopic and the microscopic level. At the macroscopic level, the entire network is treated as one system. Macroscopic models are designed to evaluate levels of service or congestion under stochastic dwell times and demand conditions at selected locations or sections of the network. At the microscopic level, each vehicle is generated and tracked as an individual object moving through the system. The system is represented as a single entity by aggregating individual vehicle and its performance is described in terms of the averages. For example, system delay is calculated as the average vehicle delay.

At both levels of simulation, system parameters such as vehicle arrival rates, vehicle mix, and dwell times can be varied over time to reflect stochasticity. For example, in the case of airports, temporal variations in flight schedules and differences between enplaning and deplaning traffic can be included. In addition, at the microscopic level, driver preference for closer parking spaces and shorter walking distances can be incorporated by assigning vehicles, first to available spaces near the terminal entrances, and subsequently to spaces further away. In addition, the severity of congestion (and contributing reasons) at different locations along the network can be identified and evaluated.

Development of microscopic simulation tools is continuing on both sides of the Atlantic, with researchers searching for new means to represent driver behavior and include real time signal controls (see Ben Akiva, 1998, and Barcelo, et. al., 1999). Algers, et. al. (1999) have identified 58 such simulation models. Fifteen of those, including the popular CORSIM, either evolved or are under development in the US. Most European models (total of 11), including the commercially advanced PARAMICS software, hail from the UK. Barcelo, et. al. (1999) from Spain have advanced AIMSUN2 to a point where users can simulate a network in real time for traffic management purposes or using O-D matrices and route selection models for evaluation. There are also a dozen or so macroscopic and mesoscopic models in use today. The macroscopic models (e.g., METACOR) are used mostly for testing signal optimization and emergency vehicle response management. The mesoscopic models such as INTEGRATION are used for dynamic traffic assignment.

After evaluating 32 of the 58 models, Algers, et. al. (1999) have made several broad conclusions about the features, capabilities and limitations of the current models. A critical and useful observation in the conclusions that pertain to this project is that "search for parking, bicycles/motorbikes,....., and parked vehicles are essentially not modeled". It is also concluded that model developers have assumed that "vehicles go from one point to another weaving across lanes and creating queues on roads and do not often park nor search for a parking space."

Compared to highway and arterial simulation models, intermodal terminal simulation tools are few. The search undertaken as part of the present project revealed only two airport curbside simulation models (Reed, 1994, and Bender and Chang, 1997). They are parts of passenger terminal simulation packages, and are not dedicated roadway simulation. There are also few reported applications of microscopic models such as Traff-Netsim and PARAMICS at airports. Additionally, Sarosky and Wilcox (1994) have applied a discrete-event simulation model developed for rail freight terminal planning. Rizzoli, et. al. (1999) have reported the development of a model to assess the impact of different intermodal rail terminals on delays to trucks and loading space requirements. A similar tool for seaport landside simulation has been developed by Gambardella, et. al. (1998).

Although each of the above intermodal terminal simulation models treat arrival and service rates as random variables, and use object-oriented programming to efficiently describe other stochastic processes, the handling of the most important element, parking, is not explained. For example, there is no indication of how loading/unloading slots (parking spaces) are chosen, and the rules governing the movement of vehicles in and out of slots, and the search for available slots are unclear. These deficiencies as well as the desire to test the parking logic developed by the present research team led to the IDEA product described in this Report.

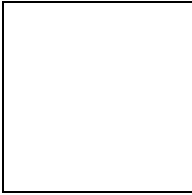
## **IDEA PRODUCT**

Access Traffic Simulation Model (ACTSIM) is the principle deliverable of this project. It was developed with two goals in mind. One is to create a tool that can be used for evaluating alternative internal road layouts and traffic management strategies under variable demand conditions and service area configurations. The second is to develop and test rule-based dynamics (or driver behavior logic) that can realistically reproduce lane changing and parking space selection in congested service areas. Given that driver behavior and needs, for example at an airport curbside, is more complex than on a highway or an urban intersection, this exercise provided an opportunity to determine how the parking logic can be



incorporated into the advanced car-following and lane-changing logic developed for AIMSUN 2 (Barcelo et. al., 1999) and MITSIM (Ben Akiva, et. al., 1998).

ACTSIM is a multi-threaded application, meaning it has two simultaneous paths of execution. The first thread is the simulation model itself, which includes pseudo random number generation, vehicle generation, car following, lane changing, path finding, and other vehicle and object interactions. The second thread takes care of all of the visual aspects of the model. This includes the program menus, toolbars, view windows, object editing, and rendering of the simulation scene in 2-D and 3-D. The two threads execute independently of one another, but share the vehicle and network data. On a multi-processor system with Windows NT, one processor can execute the first thread, while the other processor can execute the second thread taking full advantage of these systems. The main screen of ACTSIM is shown below in Figure 1.



**FIGURE 1 Animation of a typical loading/parking area**

## CONCEPT AND INNOVATION

Object-oriented modeling (OOM) technique is used in ACTSIM. OOM is a decentralized approach, which allows a service (parking) area to be examined as a composition of several objects and object types. Each object type in an OOM is defined according to its unique properties and/or behavior. In ACTSIM, for example, three types of behavior are defined for each vehicle object; *interaction with other vehicles*, *destination* (i.e., service area within the internal road network in a terminal), and *dwell time* (i.e., the time spent at each destination). The interaction with other vehicles is defined further in terms of *car following*, *finding a parking space*, *lane changing*, and *parking maneuvers*. These forms of interconnections among objects add up to an insurmountable number of combinations. Therefore, a powerful programming language that allows the gathering of ideas into a comprehensive framework had to be used.

Delphi 5.0 was chosen as the programming language of ACTSIM. Delphi was found to be ideal for meeting the conceptual design requirements, which were that it should be user friendly, be familiar to programmers worldwide, and should include the necessary graphics and file handling libraries to implement the program logic. ACTSIM was designed to run on any IBM compatible machine with a speed

of at least 500 MHz and 128MB of memory (RAM), and sufficient disk space to store the executable and input/output data files.

## INVESTIGATION

The following is a broad overview of the assumptions and features, including the sets of rules underlying ACTSIM:

The fundamental assumption is that the traffic environment is stochastic. That means demand (vehicle arrivals), service (dwell times), destination along the curb, running speed, and relative location of available parking spaces are time-dependent random quantities. Additionally, individual vehicles are tracked from the time they enter the curb area until they are processed and finally exit or circulate and return to the service area.

ACTSIM comprises of four main *modules*. One is the *network editor*. The second is the *input module*, which reads descriptions of the access road and service area from data files, and stores them as appropriate variables. The most complex and critical is the *simulation and animation module*. It is made up of several sub-modules that generate vehicles, assign characteristics to the vehicles, simulate vehicle behavior as they move through the road network, record the positions of vehicles in small time increments, and instantaneously display the activities (animate) and performance measures. The fourth module is the *post-processor module*, which compiles the simulated data, computes basic statistics, and creates data files for further analysis.

The modular design allows future expansion and improvement to the program without extensive re-coding. Moreover, it enables viewing in 3-D format and real-time model calibration.

## DATA INPUT MODULE

Input data required by ACTSIM are categorized into three fundamental types:

- *Vehicles*
- *Destinations*
- *Roadway*

*Vehicles* are further sub-divided according to follows:

- *Class*

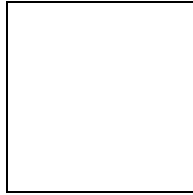
- *Type*
- *Arrival Pattern*
- *Service Pattern*

*Vehicle class* information is used to determine space requirements based on the default lengths and for distinguishing them by color in the animation of object movements (i.e., traffic flow along the curb). For instance, a bus will have its own length and a distinguishing color.

*Vehicle type* on the other hand is used to distinguish between vehicles that arrive at fixed intervals and vehicles that arrive randomly. They are defined as scheduled and unscheduled. For instance, if there are transit vehicles that arrive at fixed headways, and they can be modeled separately from others as scheduled arrivals.

ACTSIM permits users to define unscheduled *Vehicle Arrival Pattern* at each generator (entry node) in the form of demand files (i.e., arrival rates over the simulated period) or as forms and parameters of the arrival probability distributions at specified times during the simulated period. That information is used to randomly generate the headways for vehicles entering the network. Scheduled vehicle headways can be treated the same way as unscheduled vehicles or as fixed during any given period. The fixed headway can be set to change from one period to another.

*Vehicle Service Pattern* is based on whether a vehicle is dropping off or picking up a passenger, which in turn is defined as part of the input. It is assumed that the time spent at the curb ( $t_j$ ) by vehicle  $j$  is a random quantity and the Monte-Carlo technique is used in the simulation module to generate that time as an exponential random variate as follows:



Where:

$\lambda_i$  = Mean dwell time ( $i = 1$  for drop-off and 2 for pick-up)

$R_j$  = Random number (between 0 and 1) generated for vehicle  $j$

ACTSIM uses the “mother of all pseudo random number generator” developed by George Marsaglia of the Department of Statistics, Florida State University. This random number generation algorithm has been coded in assembly language, and is among the fastest pseudo random number generator algorithms. It has a period of  $3 \times 10^{47}$ , and very good randomness.

Each vehicle entering the network is randomly assigned a *Destination Zone*. Monte Carlo technique is employed for assignment according to the user-input distribution ratios (i.e., % destined to each zone). A *destination zone* is the loading/unloading (pick-up/drop-off) area, in which a vehicle will seek a parking space. Vehicles are distributed within each zone by assigning them to a series of targets represented by the main points of entry/exit of the terminal building, a bus stop, taxi cab stand, etc.

Vehicle distribution ratios can be input via an on-screen form or read from an ASCII formatted file. The physical demarcation of zones is done graphically on-screen by selecting the appropriate item from the drawing toolbox and moving the cursor over the area with the left mouse button pressed. Targets within the zone are added similarly by dragging and dropping a selected target type (e.g., door, bus-stop, taxi-cab stand). Parking space selection is centered on these targets such that the utility of the spaces decline as one moves upstream and downstream from the target.

The *Roadway* is defined in terms of the number of lanes and then, starting with the lane nearest the curb and moving outward, by the type of each lane. The type of lane is either parking and through. Certain sections of parking lanes can be designated as special parking zones. Special parking zones are areas reserved for taxis, buses, limousines, etc. Through-lanes are for traffic moving towards a designated zone, for traffic leaving the curbside area, and/or for double-parking when the curb lane is occupied. There are also median breaks (“cut-throughs”), which allow vehicles to switch between two separate sets of lanes.

## **SIMULATION AND ANIMATION MODULE**

The most important feature of the *simulation and animation module* is the rule-based dynamics. They are sets of rules that define and control the behavior and movement of each active object. A proper set of rules can ensure consistent model behavior and behavior can be changed by adding or modifying rules. The two primary sets of rules are described below. They were adapted from several sources (Kraus, 1998, Nagel, 1998, Koskonen, 1996, Ben Akiva, et. al., 1998 and Barcelo, et. al. 1999), and were modified to represent driver behavior in roadways within intermodal terminals.

### **Vehicle Dynamics**

Vehicle dynamics in the ACTSIM can be divided into three parts; *flow*, *lane changing*, and *parking dynamics*. *Flow dynamics* is what makes vehicles accelerate or decelerate in the normal traffic stream. The rules of flow dynamics govern acceleration and deceleration according to the speed of leading vehicle and spacing (gap).

The basic premise behind the *lane changing* rules is that a driver will change lanes according to the availability of a gaps and/or speeds of the adjacent lanes in relation to the present lane. If the gap is inadequate and/or relative speed is lower, then the driver will continue in the present lane, except in cases when the lane change is forced (lane end/closure or merging). In the latter case, the lane with the first available gap will be chosen. The following set of rules for *flow dynamics* and *lane changing* were adapted from HUTSIM (Koskonen, 1996):

1. Speed Up If  $\{v_{\text{current}} < v_{\text{limit}}\}$  and  $\{t - t_{\text{last}} < v_{\text{current}} * t_{\text{acceleration}}\}$

The current speed is less than speed limit and the time elapsed since last update is more than time for acceleration ( $t_{\text{acceleration}}$ ).

2. DO NOT Speed Up If  $\{D_{\text{obstacle}} < S_{\text{min}}(v_{\text{current}}, v_{\text{obstacle}}) + W_{\text{stable}}(v_{\text{current}}, v_{\text{obstacle}})\}$

The distance from obstacle ( $D_{\text{obstacle}}$ ) is less than the minimum safe distance ( $S_{\text{min}}$ ) plus the width of the stable area ( $W_{\text{stable}}$ ).

3. Slow Down If  $\{D_{\text{obstacle}} < S_{\text{min}}(v_{\text{current}}, v_{\text{obstacle}})\}$

The distance from obstacle ( $D_{\text{obstacle}}$ ) is less than the minimum safe distance ( $S_{\text{min}}$ ).

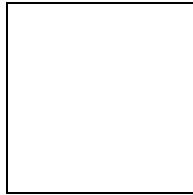
DO NOT Slow Down If  $\{v_{\text{current}} < v_{\text{obstacle}}\}$  and  $\{t - t_{\text{last}} < \max t_{\text{deceleration}}\}$

Current speed is less than the obstacle speed or maximum deceleration rate is exceeded.

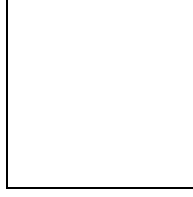
4. No Speed Change

Maintain current speed.

In ACTSIM,  $t_{\text{acceleration}}$  or the time interval between two successive steps is assumed to be constant, which makes the acceleration rate linear. Minimum safe distance ( $S_{\text{min}}$ ) and the stable area ( $W_{\text{stable}}$ ) are defined as follows:



Where:  $h$  = car-following headway (sec)  
 $S_s$  = safe separation distance (meters)  
 $d_{\text{average}}$  = average deceleration rate (m/sec<sup>2</sup>)



Values of the above parameters were obtained from traditional traffic flow models and the Highway Capacity Manual (Transportation Research Board, 1994), and are set as default values in ACTSIM.

*Lane changing dynamics* consist of two sets of rules. One for voluntary lane changing and another for forced lane changing. These two sets of rules can be stated as follows (Koskonen, 1996):

1. Decide to Change to Left If  $\{v_{\text{current}} < v_{\text{limit}}\}$  and  $\{t_{\text{lane}} > T_{\text{min}}\}$  and  $\text{Obsacle (left)} < \text{Obstacle (right)} * C_{\text{left}}$ .
2. Decide to Change to Right If  $\{t_{\text{lane}} > T_{\text{min}}\}$  and  $\text{Obstacle (right)} < C_{\text{right}} C_{\text{left}}$  and  $C_{\text{right}} =$  are lane calibration coefficients (0 to 1)

$$(N_R = 1) \text{ and } (t_{\text{front}} > T_{\text{min}}) \text{ and } \{t_{\text{back}} > T_{\text{min}}\} \text{ or} \\ (N_R > 1) \text{ and } (t_{\text{front}} > h)$$

Where:

- $N_R$  = Number of lanes available
- $t_{\text{front}}$  = Gap relative to vehicle ahead
- $t_{\text{back}}$  = Gap relative to vehicle ahead
- $h$  = Normal car-following headway
- $T_{\text{min}}$  = Critical gap for lane changing
- $C_{\text{left}}$  and  $C_{\text{right}}$  = Relative threshold coefficient allows lane changing to be controlled so that too frequent (unrealistic) switching does not occur (value between 0 and 1)

*Parking dynamics* also consist of several rules. The basic underlying premise there is that each driver starts looking for available parking spaces as soon as he or she reaches a user defined point upstream from the intended destination within the terminal area (attractor). Any space within a reasonable distance upstream and downstream of the attraction (i.e., within the user specified loading zone) will be chosen. However, the objective of a driver is to find a space closest to the attraction, and therefore, the space

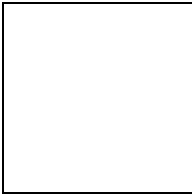
directly in front of an attractor has the highest preference. Preference for spaces upstream and downstream of the most preferred space is assumed to uniformly decrease. When a driver cannot find a space within the loading zone, he or she is assumed to join the re-circulation traffic.

When there is only one lane of curb parking, ACTSIM can be used to analyze situations where drivers park on the lane adjacent to the curb lane (unauthorized parking). In this case, it is assumed that the loading zone is an elliptical area centered at the attraction. To keep the computations simple, a lane distance weight factor (LDWF) is used to rank order the parking spaces in the elliptical zone. Accordingly, the preference for space on the adjacent lane is equal to a curb lane space at a distance ( $LDWF * a$ ) from the attractor. The value of “ $a$ ” is equal to the distance from the curb to the centerline of the adjacent lane. For example, if  $LDWF=3$ , it means that parking space nearest to the door in the adjacent lane and a parking space at a distance three times the perpendicular distance are ranked equally.

Vehicles parked along the curbside or on the opposite side of a median, adjacent to it, are counted as single parked vehicles. A vehicle will start looking for a parking space as soon as it is at a specified distance of the door. This distance is called the ‘upstream’ distance, and is specified in the vehicle arrivals input form. The distance at which a vehicle will give up parking is the ‘downstream’ distance. The above rules are illustrated as a flow diagram in Figure 2.

## Vehicle Steering

Vehicles traveling across a link or node steer by aiming at a point directly ahead of them. Vehicles changing lanes will change their destination point from one in their own lane to one in the adjacent lane. The vehicle will then change direction by rotating the front tires and ‘steering’ toward the new destination point. Vehicles can only change direction while they are moving. Tire angular acceleration is assumed constant. The vector representing the direction the vehicle is moving in is broken into two components, the direction of the tires, and the vector normal to the tires. It is assumed that the tires never slip sideways, and that this component of the forward force is lost. The force vector in the direction of the tires (minus the rolling resistance) is then broken into two components, a force in the direction the car is moving, and a force normal to the direction the car is moving. This normal force creates a moment about the point between the back tires. The vehicles moment of inertia is taken into account, its angular acceleration is calculated, and the vehicle is rotated.

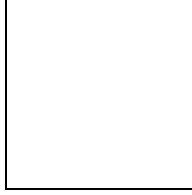


**FIGURE 2 Flow chart of parking decision process**

## Computation

To enhance the quality of the animation, objects are updated every second. Thus, vehicles entering a chosen zone, waiting in that zone, and leaving the zone can be viewed as close to real life as possible. While the simulation model is running, data are written to a file at user specified intervals. The data include the position of each vehicle in relation to the generators, attractors, sinks and other objects on the access road network as well as dwell time, speed of each vehicle. These data are in turn used to estimate total vehicle arrivals and departures during that interval (to obtain accumulation), delay (total time spent by a vehicle in a given section minus the actual dwell time minus the time to traverse that section at the maximum allowable speed). A typical set of instantaneous performance indicators that can be generated for any given section of the roadway network is illustrated below in Figure 3.





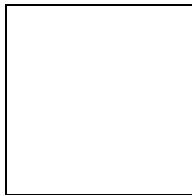
**FIGURE 3 Typical instantaneous performance indicators generated for each section**

## **GRAPHICS AND POST PROCESSOR MODULE**

As evident from the above description of the rules and procedures, ACTSIM is powerful, but at the same time detailed and complex. Consequently, it may sometimes be difficult to understand the methodologies and syntax as well as the significance of the simulation results. Therefore, a special graphical user interface (GUI) is used for to represent terminal roadways that look and behave like the real terminal roadways. It now animates graphical objects, which allows a user to monitor important system parameters in a visually compact, domain-specific manner.

The GUI permits the user to select a series of objects and direct operations of the selected object(s) with the click of a mouse-button. Then the mutual connections can be assigned and finally object attributes, if any, can be entered. Subsequently, the creation can be saved into configuration files.

A 3D view screen has been created, which only has limited capabilities. The OpenGL API was used to create the scene. Links are represented by 2D strips, nodes by 2D polygons, generator/sink pairs by 3D boxes, and vehicles by 3D boxes. The arrow keys are used to change the viewpoint. The scene is rendered in real time using the data generated by the simulation thread as shown in Figure 4.



**FIGURE 4 OpenGL 3-dimensional view**

## **POST PROCESSOR MODULE**

Results of each simulated run are written into an aggregate output file. In order to minimize memory requirements, the user can specify the simulation period, which must be recorded. It holds information on the accumulation of vehicles in the each zone area during the specified period, the utilization of curbside area (including the distances from the parking space to the door and number of vehicles parked in through lanes), and the overall delay to vehicles passing through the curbside area.

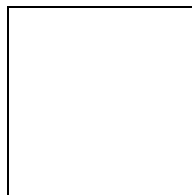
## **RUNNING ACTSIM**

Clicking on a desktop icon, which opens a 2D view/editing window, activates ACTSIM. The next step is to import a Windows Meta file created from a basic CAD file of the airport or other road network that needs to be analyzed. A special feature of AUTOCAD is used to convert the CAD file into a Meta file scaled to the right dimensions before importing.

The first step in network construction is creating the links, intersections, and the nodes and sinks, which they connect. This is done by first selecting the “Link” (first from left) button on the construction toolbar (see Figure 1) located on the top of the editing window. Holding the cursor over a button on the construction toolbar will cause the program to provide a description of the button.

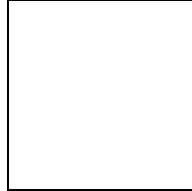
Once clicked, the cursor is moved to the starting point of the link. After clicking on the start point, one can then trace along the path you want the link to follow by clicking to place markers all the way to the end point. When the markers are all placed, clicking on the ‘Finish’ button will make the new link will appear on the screen.

Each link is automatically curved to fit all of the markers that have been placed using a B-Spline function. By default, each link drawn is a two-lane road. After it is created, the “Red-Arrow” button on the tool bar can be used to select and edit (by clicking on the right mouse button) the link to represent the true properties such as lane width, number of lanes, link speed, etc. Keyboard commands can then be used to add lanes to the left or the right sides of a link, to add or take away lanes part way through a link, and to change longitudinal markings. It may also be done by first selecting the red arrow from the toolbar and then clicking on the right mouse button with the cursor placed on the link. From the menu that appears, selection of the “Edit” option brings up the data input form shown below in Figure 5.



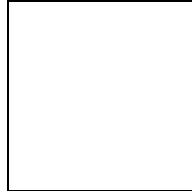
**FIGURE 5 Link properties input form**

Links can also be categorized as Parking, Traveling or Queuing. The categorization could be done by clicking on the link as before and selecting the “Properties” option, which brings up the link input data form shown in Figure 6.



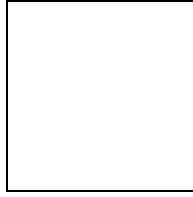
**FIGURE 6 Link data input form**

Each link can be connected to one or more links by placing an intersection (node) between them. A click on the “Node” (second button from left) button will create a circle, the center of which can be moved to the space between the links and clicked to create the connecting intersection. Node information can be edited by pointing and clicking or by selecting the node input form shown below in Figure 7 from the Main Menu (see Figure 1):



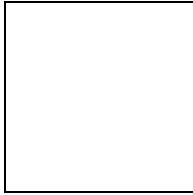
**FIGURE 7 Node properties input form**

Once the road network has been created, the vehicles need to be defined. That is done by selecting the View/Vehicles and selecting ‘Options/Add Vehicle’ to open the Vehicle Characteristics Input Form (see Figure 8), and then entering the vehicle parameters by hand. Alternatively, one can import a pre-existing vehicle definition file by selecting ‘File/Open’ and opening a ‘\*.VEH’ file as shown in Figure 5. The percentages of each vehicle type that will be created are listed in the first column. The values in this column must add up to 100. This data is used by the generators to introduce the correct mix of vehicles to the road network.



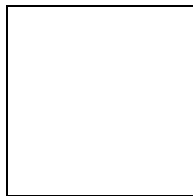
**FIGURE 8 Screen shot of vehicle characteristics input form**

The free link ends need to be capped with Generator/Sink pairs. Generators contain information on vehicle arrivals and the forms of the arrival distributions as shown in Figure 9. Sinks contain information on the links connected to it.



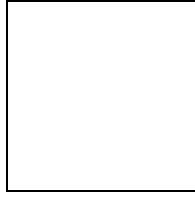
**FIGURE 9 Generator data input form**

Attractors can be assigned to any position along a link by selecting the “Attractor” button from the toolbar and clicking on the point it needs to be located. The “Properties” option of attractors can be chosen in the same manner as the links to activate the input data form for the attractors shown in Figure 10.



**FIGURE 10 Attractor data input form**

After completing the data input, simulation can be started by clicking on the “Start” button on the main window. Simulation can be run at the computer speed or at real time. Moreover, the simulation can be timed. These options can be selected from the Simulation menu on the main menu bar on top of the window shown in Figure 1 to open the simulation properties input form shown in Figure 11.



**FIGURE 11 Simulation properties input form**

## **PLANS FOR IMPLEMENTATION**

ACTSIM was calibrated and verified using data from Dulles Airport in Washington, DC. It was done in collaboration with TransCore, a transportation-consulting firm based in Virginia. TransCore provided extensive data on traffic flow (demand), vehicle assignment to attractors, access road system layout, vehicle dwell times and classifications as well as video footages of the operations. These data were used as input to compute such measures as vehicle speeds, vehicle accumulations at selected points and dwell time distributions. The modeled data were subsequently compared with observed values and it was found that ACTSIM was able to produce results with 85 to 95% accuracy. The level of accuracy of the animation, on the other hand, was lower (about 70%). It was tested using data from Salt Lake City International Airport, which took the form of comparing the number of double and triple parked vehicles estimated by ACTSIM with observed values. However, it is difficult to make a definitive conclusion because the observation period was relatively short.

Strong indications of model accuracy and user-friendliness have made it attractive to potential users. The present research team was invited in May 2000 by the Montreal Airports Authority (ADMI) to perform a series of simulations to analyze traffic management strategies at Montreal International Airport in Dorval, Canada. ADMI, which provided guidance during the development, considered ACTSIM to be the type of tool that they wish to use for landside access roadway planning. Additionally, URS Greiner Inc. has expressed interest in applying ACTSIM at JFK airport in New York for Terminal 4 curbside planning. The Wasatch Front Regional Council that overlooks transportation planning issues of the Salt Lake County in Utah, which was one of this projects primary sponsors, have invited the present research team to evaluate the Salt Lake City Olympic Committee's proposed operational plans during the 2002 Winter Olympics at Salt Lake City International Airport.

In general, ACTSIM has been well received by the consulting community as well as airport authorities. The reason being that ACTSIM is one of the few models available today for planning and managing intermodal transportation terminal access road networks, which is based on advanced programming tools, proven traffic flow principles, and state-of-the-art simulation and animation concepts.

## CONCLUSIONS

ACTSIM is based on classical traffic engineering principles, advanced vehicle control logic, and state-of-the-art object oriented programming tools. It is especially designed to handle parking as well as bottlenecks and incidents in a traffic stream. ACTSIM simulates every vehicle independently, allowing them to interact with each other, and other objects in the network. It uses a distribution of population behavior, and can handle time-variant demand or short-run traffic flow variations. This type of micro-simulation makes it possible to model curbside parking and crosswalks, features which would otherwise be impossible to model.

The OOM technique described earlier has allowed ACTSIM to provide a continuous picture of the network traffic flow and changes in performance measures. Traffic phenomena represented graphically include the grouping of vehicles behind slower vehicles, vehicle accumulation at crosswalks, conditions where double parking occurs, and where previously serviced vehicles are blocked from exiting. Simulation results can be viewed in terms of statistics; instantaneously, for a specific period or as a summary for the entire period of simulation.

Several advanced concepts emerged during the development of ACTSIM. One is the innovative parking logic (rules) that enabled the simulation and animation of a curb front where service is not only a first-come first served basis. Two types of parkers are considered in the model. One type is drivers wishing to park closest to the destination and the others who would join a queue (e.g., taxis). The logic representing the former group's search for parking space and the ultimate parking and departure from the loading zone is unique, and ACTSIM is the only model described in public domain literature that uses such logic.

The second, and perhaps a more generic advancement is the on-screen network editing capability. In comparison to the currently available public domain traffic simulation software, ACTSIM offers a unique interface for data entry and editing that is both user-friendly and graphical. Point and click capability for editing and tracking simulation data enables a user to test alternatives instantaneously or note possible errors in collected and entered data.

Up to the time writing this Report, there have been approximately a dozen enquiries about ACTSIM. Of those, four firm commitments have been received for implementation. That is a powerful indication of the potential and need for simulation and animation tools in transportation planning.

## REFERENCES

Algers, Stefan, et. al. (1999). Review of micro-simulation models. SMARTTEST project deliverable. <http://www.its.leeds.ac.uk/smartest/deliv3.html>.

Barcelo, J. et. al. (1999). Modeling advanced transport telematics applications with microscopic simulators: the case of AIMSUN 2.

Ben Akiva, Moshe, et. al. (1998). A simulation laboratory for evaluating dynamic traffic management systems. Paper presented at *INFORMS/CORS* Montreal 1998 Cluster: Advances in Microscopic Traffic Simulation.

Bender, G.G. and Chang, K.Y., (1997). Roadway and curbside traffic simulation for Las Vegas McCarran International Airport, *ASCE Conference on Airport Modeling and Simulation*, May.

Gambarella, L. M. et. al. (1998). Simulation and planning of an intermodal container terminals. Special issue on Harbor and Maritime Simulation in *Simulation Journal*. Vol. 21. No. 2 . pp. 107-116.

Gipps, P. G. (1986). A model structure for lane changing decisions. *Transportation Research*. 20B. pp 403-414.

Gipps, P. G. (1981). A behavioral car-following model for computer simulation. *Transportation Research*. 15B. pp 105-111.

Koskonen, I. (1996). HUTSIM – Simulation Tool for Traffic Signal Control Planning. Department of Civil & Environmental Engineering. Helsinki University of Technology. Laboratory of Transportation Engineering. Publication # 89.

Kraus, Stefan. (1998). Microscopic modeling of traffic flow: investigation of collision free vehicle dynamics. Doctoral Thesis. University of Cologne. DLR-Forschungsbericht 98-08.

Nagel, Kai. (1998). From particle hopping models to traffic flow theory. *Transportation Research Record 1331*. Transportation Research Board. National Research Council. Washington, D.C. pp. 1-9.

Mumayiz, S.A. (1990). An overview of airport terminal simulation models. *TRB 69th Annual Meeting*, January 1990.

Reed, K.A. (1994). Landside computer simulation using the airport landside planning system (ALPS). JKH Mobility Services, Inc.

Saroksy, Thomas and Wilcox, Terry. (1994). Simulation of a railroad intermodal terminal. Proceedings of the 1994 Winter Simulation Conference. ed. J. D. Tew, S. Manivannan, D. A. Sadowski, and A.F. Seitz. Association of Computing Machinery. December.

Transportation Research Board. (1994). Highway Capacity Manual. *Special Report 209*, National Research Council, Washington, D.C.