

# Smartest

Simulation Modelling Applied to Road Transport European Scheme Tests

<http://www.its.leeds.ac.uk/smartest>

## Micro-simulation workshop

Eric Bernauer, Laurent Bréheret, Jean-François Gabard

J. Barceló, J. Casas, J.L. Ferrer, D. García

Gianni Canepari, Carlo Di Taranto,

Ken Fox and Ronghui Liu

SMARTEST Project Deliverable D9

Submission Date: July 1999

Circulation Status: P - Public

The "SMARTEST" Project

Contract N°: RO-97-SC.1059

Project part funded by the European Commission under the Transport RTD Programme  
of the 4th Framework Programme

# Smartest

Simulation Modelling Applied to Road Transport European Scheme Tests

## Micro-simulation workshop

Eric Bernauer, Laurent Bréheret, Jean-François Gabard

J. Barceló, J. Casas, J.L. Ferrer, D. García

Gianni Canepari, Carlo Di Taranto,

Ken Fox and Ronghui Liu

### DOCUMENT CONTROL INFORMATION

Title	:	Best Practice Manual
Author(s)	:	Eric Bernauer, Laurent Bréheret, Jean-François Gabard, Jaime Barceló, Jordi Casas, Jaime Ferrer, David García, Gianni Canepari, Carlo Di Taranto, Ken Fox and Ronghui Liu
Reference Number	:	SMARTEST/D9
Version	:	1.0
Date	:	July 1999
Distribution	:	ITS, SODIT, UPC, Mizar, Transek, Softeco, CTS, CERT, DGVII(5), HIPERTRANS
Availability	:	Public
File	:	D:\SMARTEST\D9
Authorised by:	:	Ken Fox
Signature	:	

## **TABLE OF CONTENTS**

<b>INTRODUCTION</b>	<b>1</b>
<b>INTRODUCTION TO THE SMARTEST PROJECT</b>	<b>2</b>
<b>MODELLING ADVANCED TRANSPORT TELEMATIC APPLICATIONS WITH MICROSCOPIC SIMULATORS: THE CASE OF AIMSUN2</b>	<b>11</b>
<b>ENHANCEMENT OF THE DRACULA SIMULATION MODEL</b>	<b>25</b>
<b>ENHANCEMENT OF THE NEMIS MICRO-SIMULATION MODEL FEATURES</b>	<b>35</b>
<b>THE SITRA-B+ MICROSCOPIC TRAFFIC SIMULATION MODEL - EXAMPLES OF USE AND FUTURE DEVELOPMENTS</b>	<b>49</b>
<b>TRANSFERABILITY OF URBAN MICRO-SIMULATION METHODS</b>	<b>59</b>



## INTRODUCTION

This document is the ninth deliverable of the SMARTTEST project. The SMARTTEST project directly addresses task 7.3/17 in the second call for proposals in the Transport RTD, Road Transport Traffic, Transport and Information Management area. The project is directed toward modelling and simulation of dynamic traffic management problems caused by incidents, heavy traffic, accidents, road works and events. It covers incident management, intersection control, motorway flow control, dynamic route guidance and regional traffic information. The project's objectives are to:

1. review existing micro-simulation models, so that gaps can be identified
2. investigate how the SMARTTEST models can best be enhanced to fill the identified gaps, thus advancing the State-of-the-Art
3. incorporate the findings of the study into a best practice manual for the use of micro-simulation in modelling road transport and to disseminate these findings throughout Europe.

This document responds to the third objective of the SMARTTEST project: disseminate these findings throughout Europe, and it is an output of Workpackage 6, Dissemination. It consists of the proceedings of a SMARTTEST micro-simulation workshop that was held in Toulouse on 30<sup>th</sup> July 1999. Papers were presented on the enhancements made to the four micro-simulation models (AIMSUN2, DRACULA, NEMIS and SITRA-B+) of the SMARTTEST partners. The workshop programme is shown below.

### Workshop Programme

9.00-9.15	Welcome - Introduction to the SMARTTEST Project
9.15-10.00	AIMSUN 2 simulation model
10.00-10.45	DRACULA simulation model
10.45-11.00	Coffee break
11.00-11.45	NEMIS simulation model
11.45-12.30	SITRA - B+ simulation model
12.30-12.45	Discussion
12.45	Lunch
14.00	Demonstration - Workshop conclusions

## Introduction to the SMARTEST Project

Ken Fox

Institute for Transport Studies, University of Leeds, Leeds, LS2 9JT, UK

e-mail: [K.A.Fox@leeds.ac.uk](mailto:K.A.Fox@leeds.ac.uk)

### *Introduction*

Traffic congestion is a major scourge of modern life. In the UK alone it has been estimated that congestion costs the economy £15bn every year. It is now commonly accepted that trying to solve the congestion problem by building new roads will not work and is unacceptable. Road building is not only expensive and damaging to the environment but it also only provides a temporary respite to the problem. Soon after new roads have been built they induce extra traffic that negates all the benefits produced by their construction.

An alternative approach to alleviate congestion is to develop Intelligent Transportation Systems (ITS) that help make more efficient use of the existing road space. Responsive traffic control systems have been developed which measure on-street traffic flows and adapt signal timings accordingly to minimise delays. They can also detect and give priority to Public Transport vehicles. Ramp metering systems ensure traffic on motorways flows smoothly. Roadside Variable Message Signs (VMS) are used for incident management, speed control and parking guidance systems. Automatic Intelligent Cruise Control (AICC) systems allow fast moving platoons of motorway vehicles to be created and thus increase motorway capacities. Dynamic Route Guidance (DRG) systems let equipped vehicles choose the fastest route to their destinations and respond to incidents and congestion in real time. Road charging and zone access systems allow extra revenue to be collected from drivers during periods of peak congestion.

Anyone considering using of one of these new traffic management control or information systems needs to be able to predict the consequences of its introduction. They need to know both whether it works and how well it works, as it is often necessary to quantify the benefits of the new system so that they can perform a cost benefit analysis. Any disbenefits need to be identified to determine whether they are acceptable. For example a route guidance system might reduce queues in one part of the network but direct traffic through another part where it is not welcome, such as past a school or hospital or through a residential area. A wide range of performance indicators should be examined so that safety and environmental impacts can be assessed as well as efficiency. It is also useful to be able to optimise the operation of the new system to get the best out of it in the chosen location. Training in using the new system is required to ensure that it is operated correctly and efficiently.

The evaluation of these new systems to quantify their benefits can be difficult. One way would be to put the new system out on street in a before and after trial, but these trials can be difficult to assess. Many new systems are expected to have modest benefits, e.g. a new urban traffic control system may reduce travel times by less than 10%. Travel times however vary a lot from day to day anyway, so it is can be hard to determine whether any measured changes are due to the new system or simply due to chance. It is often difficult to determine whether any measured changes are due to the introduction of the new system or are due to inherent variability in the network conditions. A more promising approach is to use a traffic model to assess the system. Then the traffic engineer has complete control over the network conditions and before and after cases can be compared with greater confidence in the results.

*The Role of Micro-Simulation Models*

Traditional traffic models often treat traffic as homogenous platoons that obey simple speed / flow relationships. Such models find it difficult to assess the effectiveness of ITS which often requires interactions between individual vehicles and the new systems to be modelled.

Micro-simulation models are becoming increasingly popular for the evaluation and development of ITS (see e.g. Fox et.al., 1995, Fox et.al., 1998, Druitt, 1998). These are computer models where the movements of individual vehicles travelling around road networks are determined by using simple car following, lane changing and gap acceptance rules.

Improvements in computer performance now mean that it is possible to model peak periods on quite large road networks (hundreds of junctions and typically tens of thousands of vehicles per hour entering the network) at the micro level with a typical office PC.

An essential property of an Intelligent Transportation System is that it responds to changes in network conditions. Many implemented systems interact with individual vehicles. Responsive signal control, public transport priority and ramp metering systems react to vehicles approaching junctions. Dynamic Route Guidance systems supply specific information to individually equipped vehicles. Intelligent Cruise Control systems adjust the speeds of equipped vehicles. Therefore to assess the potential benefits of using an Intelligent Transportation System it makes sense to use an assessment tool that is capable of modelling interactions at the level of individual vehicles.

Micro-simulation models, which can reproduce individual driver behaviour, should therefore be an essential part of any such assessment tool. Moreover, as individual vehicles are being modelled it is often possible to use the micro-simulator as a proxy for the real world and connect it to real systems directly. This negates the need to produce a model of the system being assessed. For example, suppose one wanted to evaluate the benefits of introducing a responsive Urban Traffic Control (UTC) system, such as SCOOT, SCATS, UTOPIA or PROLYN. It is straightforward to link up one of these UTC systems to a micro-simulation package. The micro-simulator provides the UTC system with vehicle flows as simulated vehicles are counted by simulated detectors and the UTC system provides the micro-simulator with signal settings that it has determined will minimise costs.

Micro-simulation can be used to develop new systems and optimise their effectiveness. They can easily estimate the impacts of a new scheme by producing outputs on a wide range of measures of effectiveness. Many of these impacts, such as the amount of pollution emissions, are often difficult to measure in the field. Micro-simulation tools are also capable of providing realistic training for system operators and users prior to operation in the real world.

*The SMARTEST project*

In its Fourth Framework Programme the European Commission realised the importance of micro-simulation and put out a call for project proposals to develop microscopic modelling and simulation tools. The SMARTEST (Simulation Modelling Applied to Road Transport European Scheme Tests) project was funded to answer this call. The project started in March 1997 and was completed in July 1999. It was a multi-national collaborative effort. Partners came from the UK (University of Leeds, project co-ordinators), France (CERT and SODIT), Spain (UPC), Sweden (CTS and Transek) and Italy (Mizar and Softeco). Four micro-simulation tools were enhanced by the partners: AIMSUN2 (Barcelo et.al., 1995) - UPC, SITRA-B+ (Barbier et.al., 1990) - CERT/SODIT, DRACULA (Liu et.al., 1995) - ITS and NEMIS (Beccaria et.al., 1990) - Mizar as part of the project.

The SMARTEST project was directed towards modelling and simulation of dynamic traffic management problems caused by incidents, heavy traffic, accidents, road works, and events. It covered: incident management, intersection control, motorway flow control, dynamic route guidance and regional traffic information.

The project's objectives were to:

- review existing micro-simulation models, so that gaps can be identified.
- investigate how the existing models can best be enhanced to fill the identified gaps, thus advancing the State-of-the-Art. Prime objectives of these enhancements were to ensure that they would be transferable across Europe and that they were based on sound statistical analysis.
- incorporate the findings of the study into a best practice manual for the use of micro-simulation in modelling road transport and to disseminate these findings widely throughout Europe.

The main outputs of the project were: a State-of-the-Art review of micro-simulation models, an enhanced set of micro-simulation tools for helping network managers solve their short term traffic management problems and a best practice manual detailing guidelines and procedures for their selection and use.

#### *Review of Micro-Simulation Tools*

A review of existing models and simulation tools was performed to find problem areas that need to be modelled when developing solutions to short-term traffic management problems. A bibliographic search was carried out to find micro-simulation model developers. The search revealed the existence of fifty-seven micro-simulation models. A written questionnaire was sent out to each of the developers of these models. Thirty-two replies were received, allowing the simulation models to be analysed in a systematic fashion. Virtually all the major model developers replied to the questionnaire. The user requirements for micro-simulation models of traffic were also investigated. Data was again collected from a questionnaire, this time one was sent out to known users of road traffic micro-simulation models. In this way gaps that existed between current micro-simulation model capabilities and users' requirements were identified.

Nearly all the models use a time stepping approach where the vehicles are moved around the road network using a fixed time step, typically at one-second intervals. Only three models use an event-based approach where the states of objects in the network are changed at discrete times in response to events on an event list. Simple car following, lane changing and gap acceptance laws are used to govern vehicle movements along road links.

The number of vehicles using the network is defined by specifying origin-destination (O-D) data. The routes vehicles take have to be determined from the input O-D data. This can be done using an assignment model. Some of micro-simulation models have an assignment model or a simple dynamic route choice model built-in. Others are closely integrated with a separate assignment model allowing common use of inputs and outputs. Some of the models do not do any assignment themselves; they assume that this will be done using an external model.

Most of the models have the capability of displaying an animation of the vehicles moving round the network as the simulation progresses. Very few have a graphical network builder, which can reduce the amount of time required to input the network details considerably (see Figure 1). Most of the models provide outputs that allow efficiency indicators to be measured. These usually include travel times, travel time variability, queue lengths and vehicle speeds. About half the models now include fuel consumption and pollution emission outputs allowing environmental objectives to be assessed. Very few models produce outputs to measure safety or comfort indicators.

Most of the models are flexible in the way that key parameters can be user-defined. Integration with other models and with other databases is not so easy. Typical execution speeds are of between 1 and 5 five times faster than real time.



**Figure 1: AIMSUN2 has a user-friendly network builder**

Most of the micro-simulation models studied have been developed to quantify the benefits of Intelligent Transportation Systems, primarily Advanced Traffic Management Systems and Advanced Traveller Information Systems. The scale of application ranges from a small number of vehicles and intersections to a large number, about 200 nodes and many thousands of vehicles. Huge networks (300+ nodes and 1 million+ vehicles) can be considered by models that run on parallel architectures.

The models are usually used to estimate traffic efficiency in terms of speed and travel time, sometimes also considering congestion and queue length. They mainly concentrate on simulating traffic signal control, route guidance and traffic condition estimation. Motorcycles, bicycles, pedestrians, public transport, weather conditions and on-street parking receive little attention.

A total of fifty-one responses were received from the User Requirements Questionnaire. These came from fourteen different countries, mainly from the US, UK, France and Sweden. Half of the sample represented research organisations, another quarter road authorities, 14% were private consultants and 9% manufacturers.

The main gaps identified were a need for improved models for incident management, adaptive signal control, public transport priority, ramp metering, variable message signs, dynamic route guidance, public transport stops, vehicle detectors, roundabouts, parking and traffic calming measures. Better user interfaces and more work on validation of the existing models is also needed.

*Enhancements to the SMARTTEST Simulators*

Generic models and procedures were then specified and developed to fill the most important gaps. The new models and procedures were developed and validated using data collected from sites in Barcelona, Toulouse, Stockholm, Leeds, Turin and Florence. The SMARTTEST micro-simulation

models have been enhanced to include the new models. Comparisons were made between the new model outputs and the data collected. Table 1 shows the enhancements that were made to the four SMARTTEST micro-simulation models during the project.

Feature	AIMSUN2		DRACULA		NEMIS		SITRA-B+	
Public Transport Services			✓	☺	✓	☺	✓	☺
Roundabout			✓	☺	✓			☺
Traffic Calming				☺	✓			
Parking Management							✓	☺
Adaptive Traffic Signals	✓	☺	✓	☺	✓	☺	✓	☺
Public Transport Priority			✓	☺	✓	☺	✓	☺
Vehicle Detectors	✓		✓	☺	✓	☺	✓	
Variable Message Signs	✓	☺			✓	☺		☺
Dynamic Route Guidance		☺			✓	☺	✓	☺
Incident Management	✓	☺					✓	☺
Ramp Metering	✓	☺						
Network Builder	✓			☺				
Results Analysis		☺			✓	☺		
Better Validation		☺	✓	☺		☺		☺
✓	already existed							
☺	implemented or improved within the SMARTTEST project							

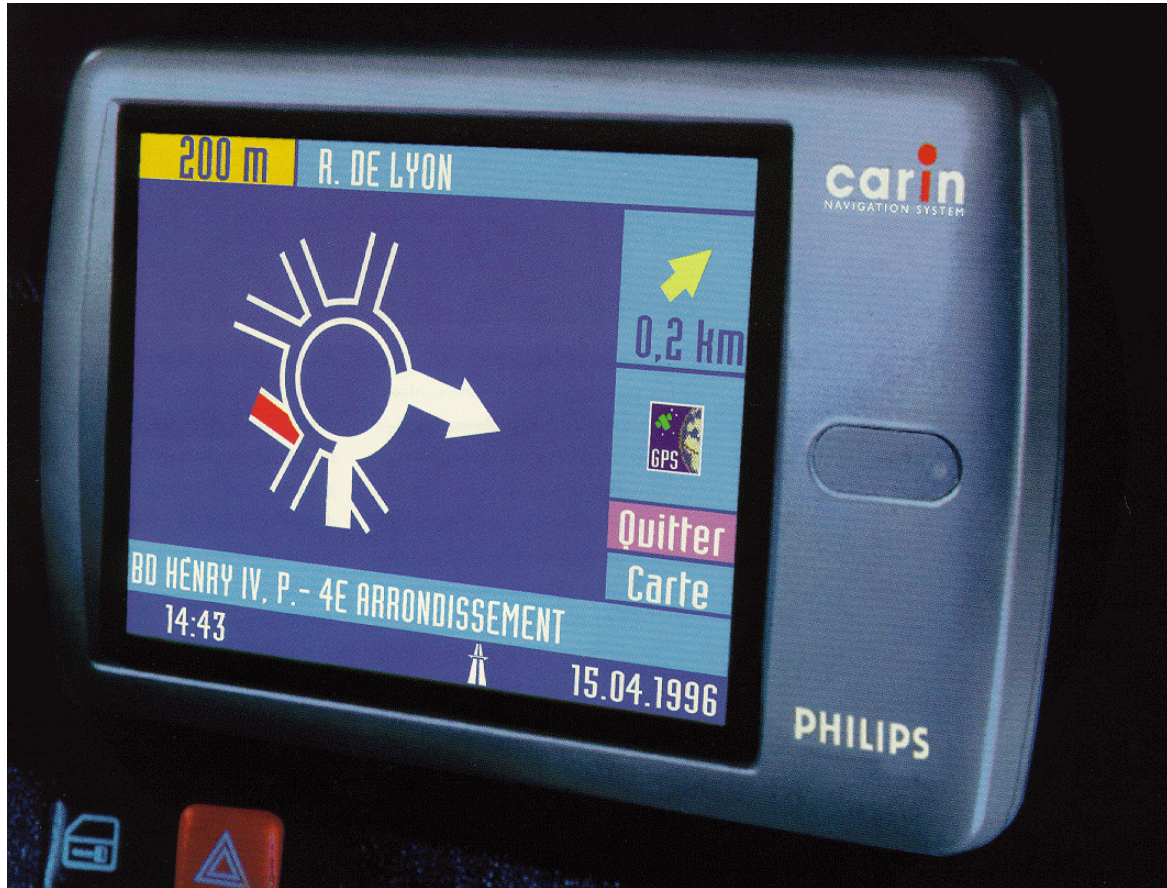
**Table 1: Improvements and new implementations in the SMARTTEST models.**

### AIMSUN2

Improvements to the incident generation model included deterministic and random incident generation. Deterministic incidents are defined either through the user's interface or by means of an incidents log file. Random incidents can be generated according to certain random distributions that are variable according to certain section characteristics.

The adaptive traffic signals improvements consisted of a new and more flexible definition of the traffic control plans and the development of a new interfacing protocol between AIMSUN2 and any external traffic control or management application. This link was implemented by the use of Dynamic Link Libraries (DLL) through which any user can implement or communicate any control or management strategy. Through this interfacing protocol it is possible not only to control any traffic signal but also any ramp metering or Variable Message Sign.

Regarding VMS and Dynamic Route Guidance Systems (see Figure 2), a better behavioural model that emulates the influence that routing information may have on the drivers was implemented. To achieve a better characterisation of the drivers, several former global parameters were transformed into local or individual parameters (i.e. compliance level and speed acceptance parameters).



**Figure 2: Improvements were made to allow better modelling of route guidance systems**

A new Result Analysis Tool was developed. Its main functions are to define and conduct simulation experiments, to perform results analysis and make data representation and to provide statistical tools for model calibration and validation.

### *NEMIS*

The main improvement to the NEMIS model was the standardisation of the interface between the micro-simulation model and external ITS applications. This activity involved producing an interface based on a TCP/IP communication protocol that was adopted to connect the computer where the model runs to the external strategy modules.

Parameters such as driver compliance to VMS and DRG indications were calibrated against the information made available by surveys conducted in the test-site by other projects. Data collection was also carried out to improve the calibration of the car following rule.

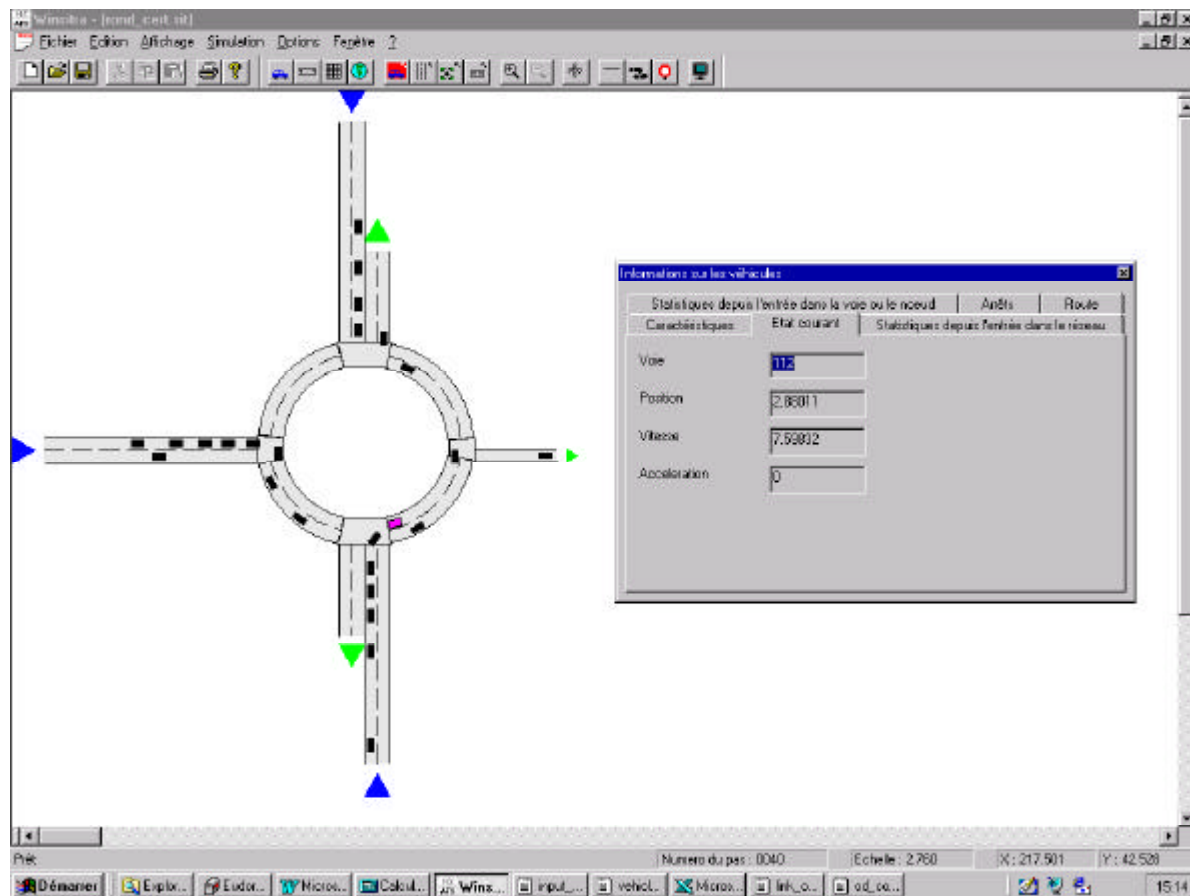
*DRACULA*

Improvements in the PT services model included a new bus stop model and the development of guided bus (see Figure 3: DRACULA can model kerb guided buses) operations. New roundabout and traffic calming models were developed, which were calibrated and validated using data collected in Leeds.



**Figure 3: DRACULA can model kerb guided buses**

The adaptive traffic signals improvements developed a link from DRACULA to a BALANCE UTC system. The improvements in the detector model in DRACULA concentrated on providing the BALANCE system with the on-street information it required. As well as the usual loop detector data this also included both public transport and emergency vehicle location information. PT Priority looked at the priority measures to both buses and trams that are provided by the BALANCE system. A test network in London was used to calibrate and validate the new models.

*SITRA-B+*

**Figure 4: The new roundabout model in SITRA-B+**

The improvements to the public transport services modelling consist of a better definition of routes, schedules and stops. The ability to model bus stop lay-bys, including new behaviour rules for pulling into or out at the bus stop, has been added.

A complete roundabout model was implemented (Figure 4). This new development addresses both driver and vehicle behaviour models. Simple rules were defined in order to deal with lane changing decisions both approaching and driving on the roundabout, and new behaviour parameters were introduced for the gap acceptance model when joining the roundabout. Video data from a test site in Toulouse was used for the validation of this roundabout model.

The Parking Management model improvements mainly deal with on-street parking. Parking places are not considered as destination or origin nodes, but as intermediary destination nodes with a given stopping probability. A series of parking spaces at precise locations is associated with each street parking node. Mean and standard deviations of parking duration are parameters that can be set by the user for each street parking set.

Improvements were also made to the VMS model and the Incident Management model.

#### *Best Practice Manual*

A best practice manual directed towards modelling and simulation of dynamic traffic management problems was produced. It covered incident management, intersection control, motorway flow control, dynamic route guidance and regional traffic information. It included:

- a methodology for defining scheme objectives and relating them to performance indicators,
- a set of guidelines for selecting a suitable micro-simulation model,

- a comparison of micro-simulation with macro-simulation,
- calibration and validation procedures, with examples from the SMARTTEST test sites,
- procedures to help ensure that scheme evaluation tackles issues such as robustness of conclusions, introducing variability and statistical significance,
- recommendations concerning when and how micro simulation models should be used.

A full description of the project and all the SMARTTEST project reports can be found on the project web site at: <http://www.its.leeds.ac.uk/smertest/>

#### *References*

BARBIER, M., CHARBONNIER, C., FARGES, J.L., GABARD, J.F., HENRY, J.J. and POMMIER, L. "Algorithmes et Simulation de la Gestion de Feux et du Trafic Routier - Rapport Final", 1/7714, Office National D' Etudes et de Recherches Aeronautiques (ONERA); Centre D' Etudes et de Recherches de Toulouse (CERT), 1990.

BARCELO, J., FERRER, J.L., GRAU, R., FLORIAN, M., CHABINI, L. and LE-SZUX, E. "A Route Based Variant of the AIMSUN2 Microsimulation Model", Proceedings of the Second World Congress on Intelligent Transport Systems '95 Yokohama, November 9-11, 1995.

BECCARIA, G., BIORA, F., DI-TARANTO, C. and WRATHALL, C. "The NEMIS Urban Microsimulator: A Tool For The Evaluation Of Dynamic Network Control". Proceedings of the PROMETHEUS Workshop on Traffic Related Simulation, Held Stuttgart, December 1992.

DRUITT, S. "An introduction to microsimulation", *Traff. Engng Control*, 39(9), September 1998.

FOX, K., MONTGOMERY, F.O. AND MAY, A.D. "Integrated strategies for urban arterials: DRIVE II project PRIMAVERA. 1: Overview", *Traff. Engng Control*, 36(5), May 1995.

FOX, K., CLARK, S., BODDY, R., MONTGOMERY, F.O. AND BELL, M.C. "Some benefits of a SCOOT UTC System: An independent assessment by micro-simulation", *Traff. Engng Control*, 39(9), 484-489, September 1998.

LIU, R., VAN VLIET, D. AND WATLING, D.P. "DRACULA - Microscopic Day-to-Day Dynamic Modelling of Traffic Assignment and Simulation", Paper presented at the Fourth International Conference on Applications of Advanced Technologies in Transportation Engineering, Capri, Italy, 27-30 June 1995.

## Modelling Advanced Transport Telematic Applications with Microscopic Simulators: the Case of AIMSUN2

J. Barceló<sup>1</sup>, J. Casas<sup>2</sup>, J.L. Ferrer<sup>1</sup> and D. García<sup>2</sup>

<sup>1</sup> Laboratori de Simulació i Investigació Operativa, Departament d'Estadística i Investigació Operativa, Universitat Politècnica de Catalunya, Pau Gargallo 5, 08028 Barcelona, Spain

e-mail: lios@eio.upc.es

<sup>2</sup> TSS-Transport Simulation Systems, Tarragona 110-114, 08015 Barcelona, Spain

e-mail: info@tss-bcn.com

### *Abstract*

The simulation of Advanced Transport Telematic Applications requires specific modelling features, which have not been usually taken into account in the design of microscopic traffic simulation models. This paper discusses the general requirements of some of these applications, and describes how they have been implemented in the microscopic traffic simulator AIMSUN2.

### *1. Introduction*

Microscopic traffic simulators are simulation tools that emulate realistically the flow of vehicles on a road network. The main modelling components of a microscopic traffic simulation model are: an accurate representation of the road network geometry, a detailed modelling of individual vehicles behaviour, and an explicit reproduction of traffic control plans. The primary attention has been paid usually to the proper modelling and calibration of all these model components, namely the car-following, gap acceptance, lane change, and other internal models which along with other modelling parameters accounting for attributes of the physical system entities, allow the microscopic simulation model reproduce flow, speed, occupancies, travel time, average queue lengths, etc. with enough accuracy to consider the model valid.

The advent of the Advanced Transport Telematic Applications made possible by combining the developments in informatics and telecommunications applied to transportation problems, has created new objectives and requirements for micro-simulation models. Quoting from Deliverable D3 of the SMARTTEST Project [1]: "The objective of micro-simulation models is essentially, from the model designers point of view, to quantify the benefits of Intelligent Transportation Systems (ITS), primarily Advanced Traveller Information Systems (ATIS) and Advanced Traffic Management Systems (ATMS). Micro-simulation is used for evaluation prior to or in parallel with on-street operation. This covers many objectives such as the study of dynamic traffic control, incident management schemes, real-time route guidance strategies, adaptive intersection signal controls, ramp and mainline metering, etc. Furthermore some models try to assess the impact and sensitivity of alternative design parameters".

The current trend in the development of Advanced Transport Telematic Applications, either real-time adaptive, or based on other specific approaches, is far from being standardised. It is therefore an exercise of dubious utility to try to integrate them in a fixed way in a microscopic traffic simulator. The relative gain achieved by including any of these, as an in-built function of the microsimulator is limited to simulating, on an easier way, those road networks on which the selected application is operating. However there would be no means of simulating other systems with that microsimulator. This is true whenever we address the problem of simulating adaptive traffic control systems as, for example, SCOOT, SCATS, vehicle actuated, control systems giving priority to public transport, etc., Advanced Traffic Management Systems (using VMS, traffic

calming strategies, ramp metering policies, etc), Vehicle Guidance Systems, Public Transport Vehicle Scheduling and Control Systems or applications aimed at estimating the environmental impacts of pollutant emissions, and energy consumes. The main question then is: How can these Advanced Transport Telematic Applications be properly evaluated and tested by simulation?

From a conceptual point of view the operation of these modern systems can be described as follows: for certain applications the road network is suitably equipped with traffic detectors of various technologies (loop detectors, image processing detectors, etc.), with a specific layout depending on the requirements of the control approach. They supply the necessary real-time traffic data (flows, speeds, occupancies, etc) with the required degree of aggregation. These real-time traffic measurements feed the logic of the traffic control or management system which, after suitable processing, makes ad hoc control decisions: e.g. extend the green phase, change to the red phase, apply some traffic calming strategies, etc.. Other applications as, for example vehicle guidance, public transport monitoring systems, or the evaluation of environmental impacts, require the access to vehicle data (position, speed, acceleration, etc.), to emulate the up-link messages in vehicle guidance applications, the vehicle tracking for the public transport monitoring or fleet management systems, or simply to provide the required data for certain fuel consumption or pollutant emissions models. To evaluate and test any of these systems a microsimulator must be capable of incorporating in the model the corresponding traffic devices as objects: i.e. detectors, traffic lights, VMS, etc. It must also emulate their functions: provide the specific traffic measurements at the required time intervals, increase the phase timing in a given amount of time, implement a traffic calming strategy (slow down the speed on a road section, recommend an alternative route, etc). How can such evaluations be done by simulation without explicit in-built modelling of the specific Advanced Telematic Application?

## 2. *The GETRAM/AIMSUN2 Micro-simulator*

### 2.1 GETRAM

GETRAM (Generic Environment for Traffic Analysis and Modelling), [2-3], is a simulation environment comprising a traffic network graphical editor (TEDI), a microscopic traffic simulator (AIMSUN2), a network data base, a module for storing results and an Application Programming Interface to aid interfacing to assignment models and other simulation models, as for example the macroscopic traffic assignment EMME/2 system, [4]. The functional structure of the systems is depicted in figure 1.

### 2.2 AIMSUN2 microscopic simulator

AIMSUN2 (Advanced Interactive Microscopic Simulator for Urban and Non-Urban Networks), [5], is a microscopic traffic simulator that can deal with different traffic networks: urban networks, freeways, highways, ring roads, arterial and any combination of them. AIMSUN2 simulates traffic flows either based on input traffic flows and turning proportions, or on O-D matrices and route selection models. In the former, vehicles are distributed stochastically around the network, whereas in the latter vehicles are assigned to specific routes from the start of their journey to their destination. Different types of traffic control can be modelled in AIMSUN2: traffic signals, junctions without traffic signals (give way or stop signs) and ramp metering. Vehicle behaviour models (car following, lane change, gap acceptance, etc.) are function of several parameters that allow modelling of different types of vehicles: cars, buses, trucks, etc. They can be classified into groups, and reserved lanes for given groups can also be taken into account.



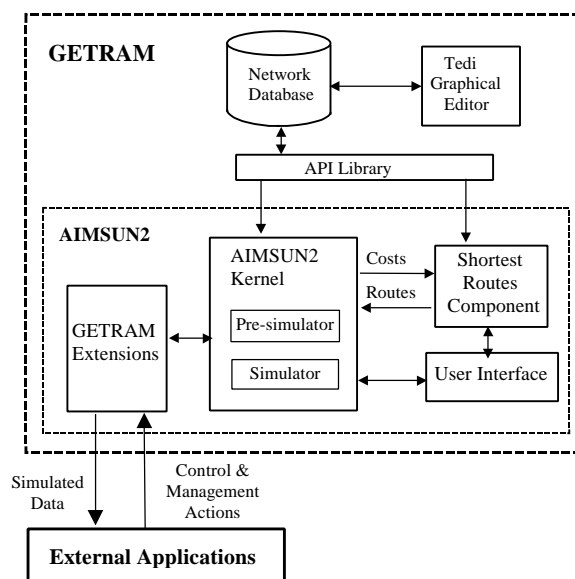


Figure 1

The logic of the simulation process in AIMSUN2 is illustrated in the diagram of figure 2. It can be considered as a hybrid simulation process combining an event scheduling approach with an activity scanning. At each time interval (simulation step), the simulation cycle updates the unconditional events scheduling list, that is events like traffic light changes which not depend on the termination of other activities. The “Update Control” box in the flow chart represents this step. After this updating process starts a set of nested loops updating the states of the entities (road sections and junctions) and vehicles in the model. Once the last entity has been updated the simulators performs the remaining operations: input new vehicles, collect new data, etc.

Depending on the type of simulation new vehicles are input into the network according to flow generation procedures (headway distributions for example) at input sections, or using time sliced O-D matrices and explicit route selection, as shown in the diagram. The simulation process includes in this case an initial computation of routes going from every section to every destination according to link cost criteria specified by the user. A shortest route component calculates periodically the new shortest routes according to the new travel times provided by the simulator, and a route selection model assigns the vehicles to these routes during the current time interval. Vehicles hold the assigned route from origins to destinations at least the had been identified as “guided” at generation time, then they can dynamically change the route en route as required for simulating vehicle guidance and vehicle information systems.

The car following model implemented in AIMSUN2 is based on the Gipps model [6], and can be considered as an ad hoc evolution of this empirical model in which the model parameters are not global but determined by the influence of local parameters depending on the “type of driver” (limit speed acceptance of the vehicle), the geometry of the section (speed limit on the section, speed limits on turnings, etc.), the influence of vehicles on adjacent lanes, etc. The lane change model in AIMSUN2 can also be considered as a further evolution of the seminal Gipps lane change model [7]. Lane change is modelled as a decision process analysing the necessity of the lane change (as in the case of turning manoeuvres determined by the route), the desirability of the lane change (as for example to reach the desired speed when the leader vehicle is slower), and the feasibility conditions for the lane change that are also local, depending on the location of the vehicle on the road network.

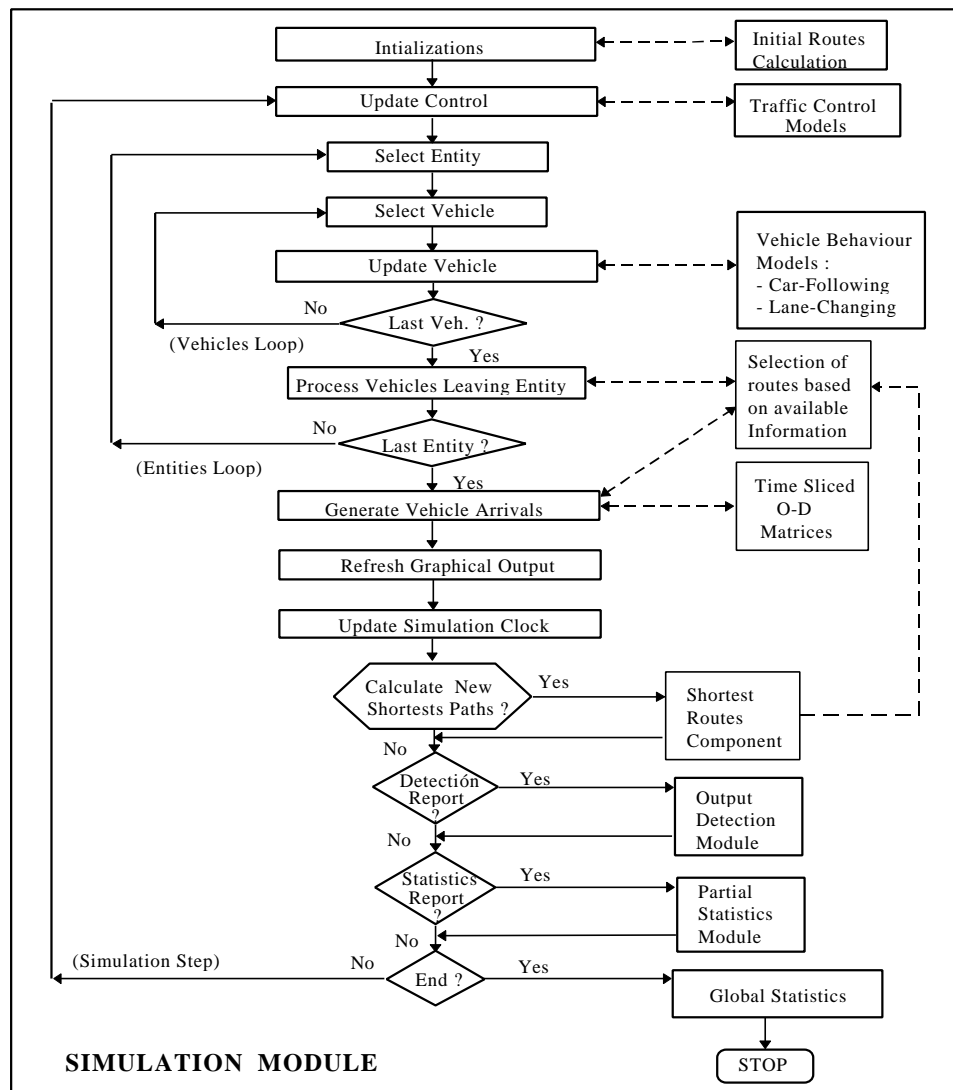


Figure 2: Logic of the Simulation Process in AIMSUN2

AIMSUN2 can also simulate any kind of measurable traffic detector: counts, occupancy and speed. AIMSUN2 has a user-friendly interface through which the user can define the simulation experiment. It also provides a picture of the network and an animated representation of the vehicles in it. The user has an overview of what is happening in the network that aids performance analysis. Through the interface, the user may access any information in the model and define traffic incidents before or during the simulation run.

### 2.3 TEDI network editor

TEDI is a graphical editor for traffic networks. It has been designed with the aim of making the process of network data entry and model building user-friendly. Its main function is the construction of traffic models with which to feed traffic simulators like AIMSUN2. To facilitate this task the editor accepts as a background a graphical description of the network area, so sections and nodes can be built subsequently into the foreground. The editor supports both urban and interurban roads, which means that the level of detail covers elements such as side lanes, entrance and exit ramps, intersections, traffic lights and ramp metering. TEDI has an interface to the EMME/2 DATA BANK, providing the means to complement a macroscopic analysis effortlessly with a microscopic one using the same traffic data (i.e. O/D matrices). The user can define a hierarchical tree of views so that a traffic model can be restricted to one of these views. The editor is designed for the level of detail required by regional, intermediate and local areas, with increased modelling detail. A library of high-level, object-based application programming

functions, named TDFunctions, assists the development of interactive external applications, and, in general to access any data. The TDFunctions enable objects in the network to be read and manipulated or can restrict the view to a sub-area. Results storage and control plans are also accessed with these functions.

### 3. GETRAM/AIMSUN2 Extensions

To cope with the requirements of simulating Advanced Transport Telematic Applications specific extensions to GETRAM/AIMSUN2 have been developed. These extensions fall into three categories:

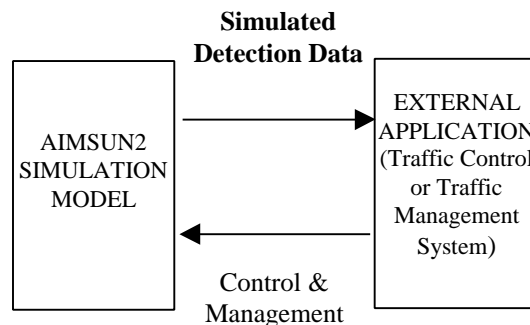
3.1 Adaptive Traffic Control, Traffic Management Systems and Incident Management Systems

3.2 Vehicle Guidance, Fuel Consumption and Emissions

3.3 Public Vehicle Scheduling and Control Systems

The approach taken in GETRAM/AIMSUN2 consists of considering the Advanced Telematics Application to be tested as an EXTERNAL APPLICATION that can communicate with GETRAM/AIMSUN2. An ad hoc version of AIMSUN2 including a set of DLL has been developed for this purpose. This library gives AIMSUN2 the ability to communicate with almost any of the above-mentioned external applications.

Using the TEDI & AIMSUN2 functions the detector, VMS and traffic lights can be modelled and their attributes defined. The process of information exchange between AIMSUN2 and the external application is shown in Figure 3. The AIMSUN2 model of the road network emulates the detection process providing the external application with the required "Simulation Detection Data". The EXTERNAL APPLICATION (user provided) decides which control and/or management actions have to be applied on the road network and sends the corresponding information to the simulation model which then emulates their operation through the corresponding model components such as traffic lights, VMS, etc. Another set of DLL function enables the user to access the information on each vehicle state (position, speed, acceleration,



etc.) at each simulation cycle.

Figure 3

#### 3.1 Simulating Management and Control actions

Three main types of actions, as a result of the actuation of EXTERNAL APPLICATION on the simulation, are taken into account:

- 1) Actuate control of traffic lights and ramp metering
- 2) Actuate control of ramp metering
- 3) Supply information to the driver using Variable Message Signs (VMS)

### 3.1.1 Traffic Control Actions

AIMSUN2 takes into account two types of traffic control: traffic lights and ramp metering. The first is considered for urban type intersection nodes, while the second one is for controlling freeway entrance ramps.

For the intersection control, a phase-based approach is applied in which the cycle of the junction is divided into phases where each one has a particular set of signal groups with right of way (a signal group is considered as one traffic light). During the simulation of a scenario, AIMSUN2 executes a fixed control plan taking into account the phase modelling for each junction. An EXTERNAL APPLICATION can modify this execution by means of different actions. The available actions are to:

1. Change the duration of each phase: The EXTERNAL APPLICATION can increase or decrease the duration, but the control plan structure is not modified.
2. Disable the fixed control plan structure: The EXTERNAL APPLICATION disables the structure of the control plan and completely controls the phase changing.
3. Change the current phase: The EXTERNAL APPLICATION can change the current phase to another. If the fixed control plan (timings) of the junction have not been disabled, AIMSUN2 programmes the next changing of phase taking into account the duration of the phases. Otherwise AIMSUN2 holds the new phase until the EXTERNAL APPLICATION changes it to another.

Examples of functions of the DLL library relative to control junctions are:

- Read Number of junctions: Reads the number of junctions present on the road network.
- Read the Identifier of a junction: Reads the identifier of junction elem-th present on the road network.
- Read the Name of a junction: Reads the name of junction identified by idJunction.
- Read the number of Signal Groups of a junction: Reads the number of signal groups defined in junction identified by idJunction.
- Read the Number of Phases of a junction: Reads the total number of phases of a junction in the current control plan.
- Read Time Duration of a phase of a junction: Reads the maximum, minimum and current duration of a in junction during the current control plan.
- Read the Current Phase of a junction: Reads the current phase of a junction, even when the junction has disabled the fixed control plan.
- Disables the fixed control plan of a junction: Disables the fixed control plan of a junction, so the phase changing is completely controlled by the EXTERNAL APPLICATION (i.e. an adaptive traffic control system).
- Change of Phase: Changes to idPhase in junction identified by idJunction.
- Change the Current Duration of Phase: Changes the duration of idPhase in junction identified by idJunction in the current control plan.
- Change the State of a Signal Group: Changes the state of signal group idSigGr in junction identified by idJunction.

### 3.1.2 Ramp Metering

AIMSUN2 also incorporates ramp-metering control. This type of control is used to limit the input flow to certain roads or freeways in order to maintain certain smooth traffic conditions. The objective is to ensure that entrance demand never surpasses the capacity of the main road. AIMSUN2 considers three types of ramp metering depending on the implementation and the parameters that characterise it:

1. Green time metering, with parameters green time and cycle time. It is modelled as a traffic light.
2. Flow metering, with parameters platoon length and flow (veh/h). The meter is automatically regulated in order to permit the entrance of a certain maximum number of vehicles per hour.
3. Delay metering, with parameters mean delay time and its standard deviation. It is used to model the stopped vehicles due to some control facility, such as a toll or a customs checkpoint.

The EXTERNAL APPLICATION can modify this modelling by different actions. It can:

1. Change the parameters of a metering, the EXTERNAL APPLICATION can dynamically modify the parameters that define a ramp metering.
2. Disable the control structure: EXTERNAL APPLICATION disable the structure of the ramp metering and completely controls the state changing.
3. Change the state of a metering: The EXTERNAL APPLICATION can change the current state to another. If the metering has not disabled the control, AIMSUN2 programmes the next changing of state taking into account the parameters, which define the control. Otherwise AIMSUN2 holds the new state until the EXTERNAL APPLICATION changes it to another.

Examples of functions of the DLL relative to ramp metering are:

- Read the Section Identifier of a metering: Reads the section identifier that contains the metering element present on the road network.
- Read the Type of metering: Reads the type of metering present in an section. (The type of a metering can be 0: None; 1: GREEN METERING; 2: FLOW METERING; 3: DELAY METERING.)
- Read the Control Parameters of a Green Metering: Reads the parameters of a green metering that are defined in the current control
- Change the Control Parameters of a Green Metering: Changes the parameters of a green metering that are defined in the current control.
- Read the Control Parameters of a Flow Metering: Reads the parameters of a flow metering defined in the current control.
- Change the Control Parameters of a Flow Metering: Changes the parameters of a flow metering that are defined in the current control.
- Read the Control Parameters of a Delay Metering: Reads the parameters of a delay metering that are defined in the current control.
- Change the Control Parameters of a Delay Metering: Changes the parameters of a delay metering that are defined in the current control.
- Disable the fixed control plan of a metering: Disables the fixed control plan of a metering, so the state changing is completely controlled by the EXTERNAL APPLICATION.

### 3.1.3 Variable Message Signs

Providing information to drivers is a possible action of a Traffic Management System on a road network equipped with Variable Message Sign infrastructure. Messages may inform drivers about the presence of incidents, congestion ahead or suggest alternative routes. AIMSUN2 takes into account the modelling of Variable Message Sign (VMS) as defined in TEDI by means of a dialogue including the VMS name or identification code, its position in the section the activated message, if any, the list of feasible messages for this VMS, and the list of all Actions available for this network associated to the messages.

Messages in a VMS from its message list may be activated in two different ways: directly through the user interface or by an external application through the communication interface. In both cases, it will cause the message to be displayed as Activated Message and the Actions associated with it to be implemented. Each message has a list of Actions associated with it which appear in the list box named 'Mess Actions' in the VMS Information Window. The list box named 'Actions' contains all actions available for this network. An Action represents the expected impact a message has on driver's behaviour. Examples of Actions are: modifications of the speed limit, modification of the input flow, modifications of the turning proportions

When simulating with the Route Based option actions can also imply a re-routing, that is the possibility of altering the vehicle's path. This effect is accomplished by defining the next turn and/or defining a new destination. The re-routing effect is defined by the following for each modality independently:

- Compliance level ( $\delta$ ): This parameter gives the compliance level of the action. If  $\delta=1$  then it causes the re-routing to be followed by all vehicles (i.e. it is obligatory). If  $\delta=0$  then the re-routing action will be followed depending on the driver behavioural parameter (a local parameter of it), i.e. it is an information only. When  $0 < \delta < 1$ ,  $\delta$  gives the level of acceptance, e.g. it is advice
- Modify the next turning: Change, which is the next turning that the vehicle must follow. This action is defined taking into account the destination
- Modify destination: When a vehicle enters into a section affected by an action, the simulator changes its destination

Functions relative to VMS of the DLL library are:

- Read the Identifier of a VMS: Reads the section identifier that contains the VMS elem-th present on the road network.
- Read the message of a VMS: Reads the text of a message elem-th in a VMS
- Read the Current Active Message: Reads the current active message in a VMS
- Active a Message in a VMS: Actives in PanelName the Message. AIMSUN2 executes the actions associated in Message

### 3.1.4 Detector Measurements

Detection output data is produced by AIMSUN2 periodically, provided that there are detectors defined in the network and the Detection Function of the simulator is activated. Currently there are two main types of Detection implemented: Common Detection Model and EXTERNAL APPLICATION type Detection Model.

In the Common Detection Model, the data produced depends on the measuring capabilities of the detectors. There is a data line for each detector, which contains the detector identifier and the list of measures gathered. They may be Count (number of vehicles per interval), Occupancy (percentage of time the vehicle is on the detector), Speed (mean speed for vehicles crossing the

detector) and Presence (if a vehicle has been on the detector, it is set to 1). These data are stored in ASCII files.

In the EXTERNAL APPLICATION type Detection Model, the measures are given at every simulation step or aggregated each detection interval. The gathered measures are: Counts (Number of vehicles), Speeds (Mean speed for vehicles crossing the detector), Occupancies (percentage of time the detector is activated), and Presence (whether a vehicle is over the detector or not). The EXTERNAL APPLICATION can undertake the following actions with detectors: retrieve the number of detectors in the network, retrieve the name of each detector, retrieve the detection interval, retrieve the detector measures gathered in each simulation step, retrieve the aggregated detector measures.

### 3.2 Simulating incidents and incident management

The diagram in figure 4 schematises the methodological procedure proposed for the simulation of incident detection and management based on the EXTERNAL APPLICATIONS. The procedure is based on a microscopic simulation model of a site that emulates traffic conditions on the site, and generates traffic data: flows, occupancies, speeds, (travel times when required), at the sampling rate requested by the external applications (for example 30 seconds is a standard request for most automatic incident detection algorithms, [8]), with the format proper of the technology used at the site. These traffic data feed the Incident Warning, Incident Detection and Traffic Management Modules implementing the corresponding External Applications.

The Incident Warning applications estimate an incident probability [9] that is sent as a warning to the Traffic Management System that may take it into account. The Simulation model dialogues with the Management System, as an external application, in the way described above. Once the Incident Detection Module detects an incident, it generates an incident alarm, which is sent to the Traffic Management System. The management decisions are communicated to the simulation model through the proper dialogue as described above. Simulation of Vehicle Guidance

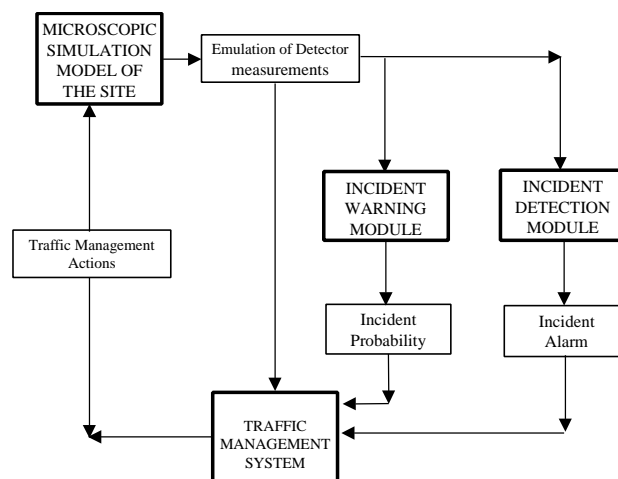


Figure 4

### 3.3 Simulation of Vehicle Guidance

Vehicle Guidance can also be modelled as an External Application that can be properly simulated with AIMSUN2 by means of an exchange of information using the suitable DLL functions. Essentially the simulation of a Vehicle Guidance system, [10], consists of an exchange of information between the equipped car and the Traffic Information Centre. The on board equipment collects information on the vehicle's position, travel time, speed, experienced delay, number of stops, etc. which is sent to the Traffic Information Centre, [11-13], by means of the

telecommunication technology on which the system is based, as for example, beacons or GSM, or both.

The vehicle data collection can be modelled as follows: each equipped vehicle sends the information to the system when passing through certain points on the network. In the network description the definition of these Data Collection Points (DCP) have been included working like detectors. Each time the position of a guided vehicle is updated it is checked whether or not it has passed through a DCP. If so, a message to the information centre is sent and the vehicle information is updated. Alternative data collection procedures can be implemented for the simulation, as for example assuming that there are a set of fixed DCP which correspond to the end of each section, or that DCP are variable and their position depends on the behaviour of each guided vehicle, for instance, if it takes more than certain time for a vehicle to cross a section or if it has to stop during a section journey. Examples of floating car data collection sampling procedures, as illustrated in figure 5, are: section based (a vehicle sends a message whenever it reaches the end of any section in the network); time based (a vehicle sends a message every certain time interval whose length can be selected by the user as a simulation input parameter); dual mode sampling (time and section based, if a vehicle takes more than one time interval to travel one section, a message is sent for each time interval); speed and section basis: a vehicle sends a message at the end of each section. Besides, if during the trip along a section the vehicle speed falls below a minimum value (which is set by the user as a simulation input parameter), a message is sent every time interval until the end of section is reached.

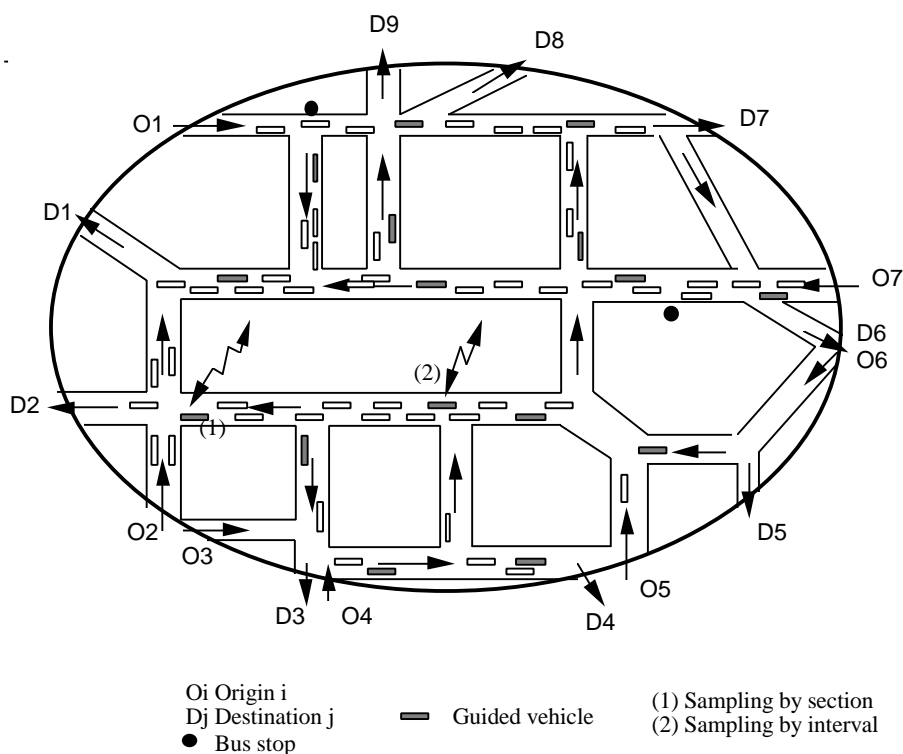


Figure 5: Microscopic Simulation of Vehicle Guidance Systems

Guided vehicles, which are a percentage of the total, are identified at generation time and the data collection process is initialised. For each guided vehicle the corresponding DLL functions collect the information on the vehicle data. Additionally, each time a guided vehicle is updated, it is checked whether or not it is time to transmit data. If so, the following information is updated by means of the ad hoc DLL: Number of messages sent, Number of data blocks transmitted, Sections



and distance travelled since last message, Time for the next transmission. In a similar way every time a guided vehicle reaches the end of a section, the following information is updated: Number of messages sent, Number of data blocks transmitted, Distance travelled since last message, Time since last transmission.

The simulation can thus provide the following information [14]: traffic volumes (guided and unguided); mean vehicles speed (km/h); number of stops per vehicle/kilometre; communications overhead in total number of messages between vehicles and information centre, per time unit, and total number of blocks transmitted per time unit; mean time between messages per vehicle; average distance travelled between messages per vehicle and average number of messages per section and vehicle. The first three items estimate the level of network congestion in the simulation experiment. The next two estimate the communications requirements for the system, and the last three are measures about the quality of the overall information received by the centre. To illustrate these simulation results let us summarily describe a typical simulation experiment of a guidance system. The parameters defining the experiment are the following: MNV, mean number of vehicles in the network; MNG, mean number of guided vehicles in the network; NTR, number of trips / hour. Trips from origins to destinations in the modelled network; SPD, mean journey speed in kilometers/hour; NST, number of stops per vehicle per kilometre travelled. The results on communications requirements produced as output by the AIMSUN2 simulation are represented by the following variables: MpS, number of messages (transmissions) per second sent from guided vehicles to the information centre; BpS, number of data blocks transmitted per second. For instance, a trip message is composed by one data block for the header and one data block for each section travelled. Taking into account the type of information of each section of the messages, the typical length of a data block could be considered as 16 bytes. AIMSUN2 output provides also an estimate of the quality of the overall data transmitted from the equipped vehicles to the information centre by means of the following variables: TbT, mean time between transmissions per vehicle, which is a measure of how often the centre has information from a given vehicle; DbT, mean distance travelled between transmissions per vehicle, this measure together with the TbT, gives an idea about the frequency of information updating for each vehicle; CpS, mean number of messages transmitted per section per vehicle. This provides measures of the information quality for each section.

Thus, for example, for a simulation experiment done in the SOCRATES project [12], [14], the traffic conditions were characterised by the following values:

MNV	MNG	NTR	SPD	NST
10000 veh	500 veh	54000 trips	30 km/h	6.2 stops

Therefore, the mean number of guided vehicles that the system was controlling was 500 vehicles from a total of 10000 (that is the 5%). The mean speed on the network was 30 km/h and a vehicle had to stop on average about 6 times every kilometre travelled.

The communications requirements obtained in that simulation experiment for the different sampling procedures are summarised in the following table:

	section	time30	time60	dual30	dual45	dual60	speed30	speed45	speed60
MpS	10.3	16.3	8.0	17.5	13.7	12.0	19.7	17.6	16.7
BpS	20.6	42.2	25.3	42.6	31.2	26.1	49.2	42.9	40.2

The quality of the data received by the traffic information centre can be represented by the following measures:

	section	time30	time60	dual30	dual45	dual60	speed30	speed45	speed60
TbT	44.5''	30.0''	60.0''	21.3''	27.0''	30.8''	18.9''	21.1''	22.2''
DbT	299 m	185 m	366 m	171 m	217 m	248 m	152 m	170 m	180 m
CpS	1	1.57	0.77	1.76	1.38	1.21	1.98	1.77	1.68

From this experiment, it appears that the best sampling procedures are those based on dual mode sampling on time basis. For not to fall in communication overheads, a time interval of 60 sec. or more is preferable.

The Traffic Information Centre collects the individual information from the equipped vehicles and, after a suitable processing produces the guidance information which is transmitted to the guided vehicles. The broadcasting of this information and the guided vehicle reactions can be simulated using the DLL in a similar way as the simulation of the VMS, based on the capability of the simulator to dynamically re-route the guided vehicles en-route.

#### 4. The parallelization of AIMSUN2

To conclude this description of improved modelling features in the microscopic simulator AIMSUN2 we will make some comments on what could be expected from the parallelisation of the simulator. A complete description can be found elsewhere, [15-17]. Some reasons to parallelize a microscopic traffic simulator could be the following: the situation of the current practices in traffic management can be summarised saying that, out of a limited traffic control practice, traffic management is currently based on manual procedures, relying on the experience of human operators, and off-line computing practices. There is a main reason for that, numerical algorithms, based either on optimisation or simulation approaches, to deal with time-varying traffic flows in real size traffic networks have very high computing requirements to be processed sequentially on the currently available computing platforms, the resort to parallel computing is a way for achieving the required performance for real-time applications.

The encouraging results obtained with the parallel version of AIMSUN2, reported in [17], based on a standard library of threads available in the current version of SUN Solaris 2.4, and a shared memory computing platform, where average speed ups of up to 3.5 times have been achieved, show that a quasi-real time operation of the systems using microscopic simulation for management purposes is feasible. In the computational experiments only the basic version of AIMSUN2 has been parallelized, certainly the inclusion of routing information and its use in the microsimulation will impose additional computational burden that must be investigated. On the other hand the parallelization studied depends, obviously, on the structure of AIMSUN2 and therefore the results cannot be extrapolated to other microscopic simulator with different internal structures. However, we believe that our results show that the parallelization of the microscopic simulators, on the currently available computer platforms, opens the door to simulation analysis of medium to large networks, and not only small networks as microscopic approaches had been restricted so far, and to the use of simulation as decision support tool in the context of Traffic Management applications

#### 5. Conclusions

The paper has described the requirements to simulate Advanced Transport Applications and the conceptual approaches to implement them in microscopic traffic simulators. The description has been completed showing how these approaches have been implemented in the case of the AIMSUN2 microscopic traffic simulator. These implementations have been mainly done through

the participation in projects of the ATT Programme of the European Union. References to these implementations can be found in the already referenced reports of SMARTTEST (simulation of VMS, ramp metering, etc.) [1], SOCRATES (Vehicle Guidance) [11-12], CLAIRE SAVE (Adaptive Traffic Control and Environmental impacts) [18], CAPITALS (Traffic Management) [19], IN-RESPONSE (Incident Management) [20], and PETRI, [16].

### 6. References

1. SMARTTEST Project Deliverable D3, August 1997, European Commission, 4th Framework Programme, Transport RTD Programme, Contract N°: RO-97-SC.1059.
2. R. Grau and J. Barceló. The design of GETRAM: A Generic Environment for Traffic Analysis and Modeling. Research Report DR 93/02. Departamento de Estadística e Investigación Operativa. Universidad Politécnica de Cataluña. (1993).
3. J. Barceló, J.L. Ferrer and R. Grau. AIMSUN2 and the GETRAM Simulation Environment. Technical Report. Departamento de Estadística e Investigación Operativa. Universidad Politécnica de Cataluña (1994).
4. INRO Consultants, EMME/2 User's Manual. Software Release 8.0 (1996).
5. J. Barceló and J.L. Ferrer. AIMSUN2: Advanced Interactive Microscopic Simulator for Urban Networks. User's Manual. Departamento de Estadística e Investigación Operativa. Facultad de Informática. Universidad Politécnica de Cataluña (1997).
6. P.G. Gipps, A Behavioural Car-following Model for Computer Simulation, *Transpn. Res.* 15B, pp 105-111 (1981).
7. P.G. Gipps, A Model for the Structure of Lane-Changing Decisions, *Transpn. Res.* 20B, pp.403-414 (1986).
8. Y.J. Stephanedes, A.P. Chassiakos and P.G. Michalopoulos, Comparative Performance Evaluation of Incident Detection Algorithms, *Transportation Research Record* 1360 (1996).
9. I.R. Wilmink and L.H. Immers, Deriving Incident Management Measures using Incident Probability Models and Simulation, TNO Research Report 95/NV/172, The Netherlands (1995).
10. D. Jeffery. Route guidance and In-vehicle Information Systems. In: *Information Technology. Applications in Transport*, P. Bonsall and M. Bell (eds), VNU Science Press, Utrecht, pp. 319-351 (1987).
11. SOCRATES, 1990, DRIVE I Project V1007, Commission of the European Communities, Report on WP1.2.3, 'Floating Car Data', Prepared by Hoffmann Leiter, Universitat Politècnica de Catalunya, and BAST, responsables G.Hoffmann and J.Barceló.
12. SOCRATES KERNEL 1992, DRIVE II Project V2013, Commission of the European Communities, Report SCKN/UPC 03.43.92, Equipped Vehicle Fleet Requirements for Monitoring Network Conditions in a Dynamic Route Guidance System Prepared by Universitat Politècnica de Catalunya, responsible J.Barceló.
13. I.Catling, "SOCRATES", in: *Advanced Technology for Road Transport*, ed. By Ian Catling, Artech House, London (1994).
14. J. Barceló, J.L Ferrer, and R. Martín, Simulation Assisted design and Assessment of Vehicle Guidance Systems, Accepted for publication in *International Transactions on Operations Research*, (1998).
15. J.Barceló, J.L. Ferrer, D. García, M. Florian and E. Le Saux, The Parallelization of AIMSUN2 Microscopic Simulator for ITS Applications, *Proceedings of the 3rd. World Congress on Intelligent Transport Systems*, Orlando (1996).

16. J.Barceló, J.Casas, E.Codina, A.Fernández, J.L.Ferrer, D. García and R.Grau, PETRI: A Parallel Environment for a Real-Time Traffic Management and Information System, Proceedings of the 3rd. World Congress on Intelligent Transport Systems, Orlando, (1996).
17. J. Barceló, J.L. Ferrer, D. García, M. Florian and E. Le Saux, “Parallelization of Microscopic Traffic Simulation for ATT Systems Analysis”, in: Equilibrium and Advanced Transport Modelling, ed. By P. Marcotte and S. Nguyen, Kluwer (1998).
18. CAPITALS, (1998), “Workpackage 5.2 Evaluation Report”, EU DGXIII, Project TR 1007.
19. CLAIRE SAVE, 1996, K. Fox, H. Kirby, G. Scemama and A. Walker, “Avoiding High Fuel Consumption in Congested Cities”, Project Final Report, Institute for Transport Studies, University of Leeds.
20. IN-RESPONSE, (1996, 1997) Deliverables 4.3, 5.1 and 5.2, EU DGXIII, Project TR1030.

## Enhancement of the DRACULA simulation model

Ronghui Liu

Institute for Transport Studies, University of Leeds, Leeds, LS2 9JT, UK

e-mail: [rliu@its.leeds.ac.uk](mailto:rliu@its.leeds.ac.uk)

### *Abstract*

The following five models have been improved within DRACULA as part of the SMARTTEST project:

- Roundabouts
- PT Services
- Adaptive Traffic Signals
- PT Priority
- Detectors

and one new model has been added:

- Traffic Calming

Improved validation of the model has also taken place. Improvements in the PT services model include a new bus stop model and the development of guided bus and tram operations. New roundabout and traffic calming models have also been developed, which have been calibrated and validated using data collected in Leeds.

The adaptive traffic signals improvements concentrated on linking DRACULA to a BALANCE UTC system that is due to be installed in Leeds and Sheffield. The installed BALANCE system will use the TCP/IP communications protocol to link up its various components. With this in mind a DRACULA interface that also uses TCP/IP has been developed. The improvements in the detector model in DRACULA concentrated on providing the BALANCE system with the on-street information it required. As well as the usual loop detector data this also included both public transport and emergency vehicle location information

### *1. Roundabouts*

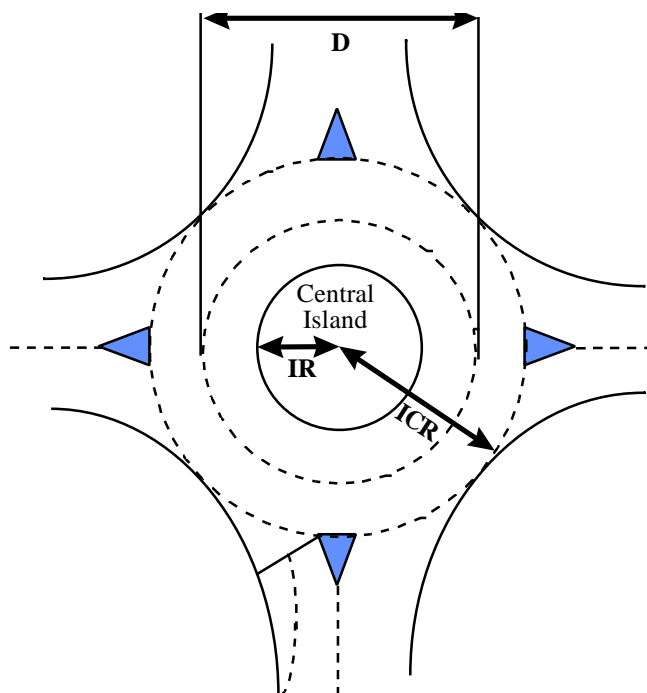
DRACULA makes the following assumptions about each roundabout:

- the roundabout is circular,
- the roundabout is modelled as a continuous link, with a given number of lanes, with entry and exit points at positions along it,
- vehicles attempt to travel at a desired circulation speed when on the roundabout,
- the usual car following rule is used on the roundabout,
- a new lane changing rule is used on the roundabout.

The following inputs are required to define the roundabout geometry:

- The roundabout diameter  $D$  (defined as the diameter of the circle round the middle of the roundabout, this is equivalent to  $(IR+ICR)$  in Figure 5: The roundabout geometry (UK), where  $IR$  is the radius of the central island and  $ICR$  is the outer radius of the roundabout).

- The desired circulation speed of vehicles travelling around the roundabout (A default value of 7.5 m/s is suggested). When each vehicle is generated it is given a scale factor to use when calculating its desired speed on a link. This scale factor is based on a random selection from a distribution with a given mean and variance. Each vehicle type has a given mean and variance to use.
- Number of lanes on the roundabout.
- Safe gap time for entering roundabout (a default value of 3.0s is suggested). This is the minimum size of gap which must exist before a vehicle will move on to the roundabout. For each vehicle type this can be modified by a scale factor. As with the desired speed this scale factor is based on a random selection from a distribution with a given mean and variance according to the vehicle type.
- Which lanes on each input arm should be used for each exit from the roundabout.



**Figure 5: The roundabout geometry (UK)**

The roundabout model uses three regimes. Firstly on approaching the roundabout vehicles have to get into an appropriate lane. When vehicles arrive at the roundabout they have to determine whether there is a suitable gap to allow them to enter the roundabout. Finally, when travelling on the roundabout vehicles have to choose an appropriate lane to allow them to leave the roundabout at the desired exit. The rules used for each of these regimes are given below in pseudo code.

Approaching roundabout

- Choose lane appropriate to exit
- Reduce speed as if approaching give way junction

Gap search - to move on to roundabout

loop through vehicles on roundabout approaching from right (UK)

if vehicle NOT exiting at this exit then

calculate time to arrival at entrance

```

    if time to arrival > safe gap time then
        move on to roundabout in exit lane
    end if
end if
until vehicle found OR distance scanned > 50m

```

Movement on roundabout

- Use usual car following rule
  - Use new roundabout lane changing rule
- if next exit is desired exit and not in outer lane then

    move to outer lane

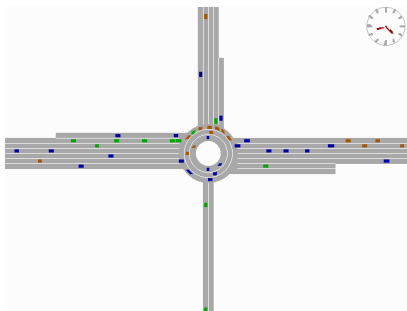
end if

if lane ahead blocked by queue from non-desired exit then

    move into non-blocked lane

end if

The main additional output is a graphical representation of the roundabout as the program runs. (See Figure 6)



**Figure 6: Roundabout output**

## 2. *Public Transport Services*

The main improvements to the Public Transport service models within DRACULA have been:

- a new public transport service model
- a new public transport stop model
- a new reserved lane model
- guided bus operation

The public transport services in the model are described by their:

- service number;
- vehicle type (bus or guided bus);
- service frequency (veh/hr);
- service route which includes:

- Origin zone number
- Destination zone number
- Number of nodes (excluding the zones) en route
- List of nodes en route
- Number of public transport stops en route
- List of public transport-stop identification numbers for all the stops en-route.

Only the departure time of the service (via a fixed hourly service frequency) is modelled. The bus schedule (in terms of route timing points) is not represented in the current version.

The public transport vehicles enter the network at regular service frequency. They follow the pre-defined fixed route through the network as other traffic except when using reserved public transport lanes where provided, stopping at public transport stops for the service.

## 2.2 Public transport stops

A public transport stop might be used by all or just some of the public transport services (or routes) that go past it. Therefore a method of defining which services use which stops has been devised.

Two types of public service stops are modelled: ordinary bus stops or bus laybys. An ordinary bus-stop is a single sign on the road side and buses stop alongside the sign on the road to pick up or put down passengers, thereby blocking upstream traffic in that lane. A bus-layby, however, provides a space for the bus to pull into at the bus-stop and thus allows following traffic to pass. A bus-layby in the model is represented as a special type of bus-stop.

Each bus stop is defined by the following data:

- A unique public transport stop identifying number
- Upstream node of the link on which the stop lies
- Downstream node of the link on which the stop lies
- Position of the stop, measured from the entry of the link
- Side of the road where the stop lies, 0=curb side, 1=median side
- Type of the stop, 0=ordinary, 1=layby
- Layby length (metre), 0 for ordinary bus stop
- Average pedestrian flow (ped/hr) to the bus stop

There are two elements required to model public transport vehicle motion in the vicinity of public transport stops. Firstly, when approaching the stop, public transport vehicles need to move into the lane where they can access the stop. The public transport vehicles begin to attempt this manoeuvre in the link before the link with the stop. Secondly, if there are passengers waiting at the stop then the public transport vehicle has to stop at the stop for sufficient time to pick up all the passengers. Pseudo code for both these elements is given below.

PT vehicle motion approaching stop

if next link has a stop on it then

try to move into lane which leads to lane with stop

if fails then

move into link



```

        look for gaps to move into lane with stop
    end if
end if

```

PT stop dwell time

```

if a bus is NOT already waiting at a bus stop then
    if number of passengers > 0 then
        stop time = a N + b
    end if
end if

```

where

N = number of passengers waiting (passengers arrive at the stop at a fixed rate as given in the input section. The stop time is extended if more passengers arrive during the stop.

a = time taken for 1 passenger to board (4s default).

b = time for doors to open and close (5s default).Reserved public transport lane

### 2.3 Reserved lanes

The lane reservation in the model is specified by:

- the link identification;
- location on the link (near or offside lane);
- type(s) of vehicles reserved for the lane;
- “set backs” at the beginning and end of the link;
- start and end time of the reservation, measured from the start of the simulation.

The following pseudo code describes the movement of public transport vehicles as they approach and move into a reserved lane.

```

if a reserved public transport lane is in the next link then
    try to change to the lane in the current link which leads to the reserved lane
    if failed to change lane then
        stay in lane until the next link
    end if
end if

```

Once in the link with the reserved lane the following logic applies

```

if there is a reserved public transport lane in the current link then
    try and move into the reserved lane
end if

```

```

if the reserved lane permits the public transport vehicle's next turn then
    stay in the reserved lane

```

else

    move off the reserved lane into a lane that allows the turn, when near the junction

end if

#### 2.4 Guided bus operation

Guided buses are represented in the model as a distinct type of vehicle. The distinction is made both in terms of vehicle characteristics and the traffic regulations governing their movement on the streets.

The guideway is represented as separate links, dedicated to the guided-bus vehicle type. The guideway can only be entered at the start of the link and exited at the end, unlike reserved lanes (see above).

The operational distinction between a guideway and a reserved lane which this implementation incorporates is that a bus may join the guideway only at dedicated points on the route whilst a bus may “drift” into and out of a reserved lane anywhere along its extent. A lane can be specified as reserved for one particular type of vehicle or a combination of vehicle types.

#### 2.5 Outputs

Additional outputs include the public transport service route specified summary statistics, which include the total travel time, distance, average speed, fuel consumption, pollutant emissions for the service route. As the user’s request, each public transport service vehicle’s link-by-link travel times are also output.

### 3. *Adaptive Traffic Signals*

It is becoming increasingly common to link micro-simulation models to real urban traffic control (UTC) systems and to then let the two systems interact. The UTC systems can obtain data from the simulated network, such as vehicle detections, and use this information to perform control actions in the simulated network. This approach has considerable merits. It negates the need to produce a model to replicate the effects of the UTC system. It also allows accurate simulations to be performed without the modeller having to know precise details of the how the UTC system works. This can avoid commercially sensitive information having to be revealed to the modeller.

Within the SMARTEST project DRACULA has been linked to the BALANCE UTC system. BALANCE (Friedrich et.al., 1995, Toomey et.al., 1998) is a distributed UTC system which has been developed at the Technical University of Munich. It underwent field trials in Munich within the EC funded DGXIII LLAMD project, and has since been tested in three other European cities, namely London, Glasgow and Belfast. It is due to be used in Leeds and Sheffield in the near future.

Within a BALANCE system, decisions about signal settings at individual junctions are made by Micro-BALANCE outstations at each junction. Strategic control decisions at the network level, which can override or weight decisions at the junction level, are made by a centralised Macro BALANCE computer. BALANCE uses standard TCP/IP communications protocols to communicate with signal controllers on-street and between its system components.

An interface between DRACULA and BALANCE was therefore developed which used the same TCP/IP communication protocols as used by the BALANCE system on-street. The interface was tested by simulating the operation of a single junction, namely the A30/Stanwell Road junction in South West London.

By using a multi-tasking operating system, such as Windows 95/98/NT, it is possible to run all the micro-BALANCE tasks on a single computer, if required, rather than use a separate computer for each one, as would be the case in the real world. A flexible approach was developed in the project to allow the micro and macro BALANCE tasks to be spread across a computer network as

available. Similarly it should be possible to treat DRACULA as just another task and run it on any of the PCs in the computer network. In practice this caused problems if DRACULA and all the BALANCE tasks were chosen to run on a single computer. It proved difficult to display the animated graphical outputs of DRACULA simultaneously with the outputs from BALANCE. A further problem was encountered when considering how to link the standard communications routines used by BALANCE into DRACULA. The DRACULA software is currently DOS based and is incompatible with the Windows based communications routines. It therefore proved impossible to link the communications routines with DRACULA directly. This problem was overcome by writing a small Windows program, called SPRUCE, to handle the communications, which was run every simulation step within DRACULA.

The design for linking DRACULA to BALANCE was based around a further INTERFACE task, which sat between DRACULA and the BALANCE tasks. The INTERFACE task acted as a server. It handled communications between all the tasks, ensured all the tasks were synchronised, and translated all the data going between the tasks into the required formats. The INTERFACE task also performed some of the duties normally carried out by on-street signal controllers, such as checking that minimum green periods had elapsed before changing signal phases.

DRACULA required the signal settings to be input every second. This was provided as a list of the current signal aspects (Red, Amber or Green) for all the possible turns at the signalised junction.

The INTERFACE program received detector data from SPRUCE / DRACULA and signal force bits from BALANCE. The force bits simply specify which stage is to be run next.

BALANCE received detector data bits from the INTERFACE program. The detector data was in the format of SCOOT nibbles (see Section 5).

The initial proposal required that DRACULA should communicate with BALANCE via an INTERFACE program. This interface is based on a function that would need to be used by BALANCE to send out signal settings and receive detector data. This function was written using Microsoft Visual C++ and used TCP/IP communications protocols. It was supplied via a DLL called ChatDll.dll and contained the exportable function 'XDataOnTCPIP' which takes four arguments as shown below

```
int XDataOnTCPIP(char* RemoteIP, int RemotePort, char* Msg2Send, char* Msg2Take);
```

This function just uses TCP/IP to send and/or receive message strings between two computers on a network. The function was to be called every second, by DRACULA, BALANCE and the INTERFACE program to transfer data between them.

To use the function it required:

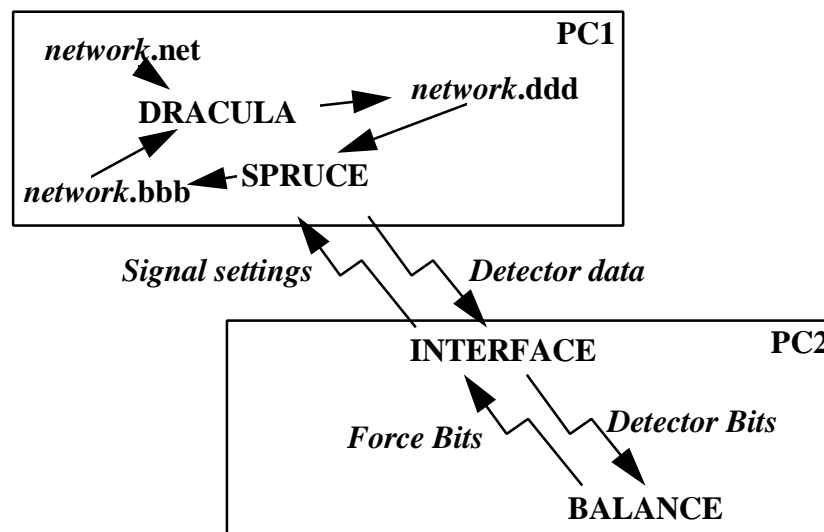
- i) BALANCE to be adapted to use the interface function
- ii) the translation of the data going to (detector bits) and from (force bits) BALANCE into something DRACULA could understand. This was performed by the INTERFACE program.

As mentioned above, it proved impossible to incorporate the XDataOnTCPIP function directly within DRACULA. Instead a simple program called SPRUCE.EXE was written which did incorporate the function. This program was run at each simulation step by DRACULA using the standard system function. At each simulation step, DRACULA would write detector data to a file called network.ddd and read signal settings from a file called network.bbb. The SPRUCE program would then be called and it would read the data in the network.ddd file and transmit it to the INTERFACE program and receive the signal settings back from the INTERFACE program at the same time. SPRUCE would then write the signal settings to the network.bbb file. (see Figure 7)

The INTERFACE program performed the following functions:

- Translation of the BALANCE force bits into appropriate signal settings.

- Translation of the DRACULA detector data into the stream of detector bits required by BALANCE
- Synchronisation of the tasks



**Figure 7: Communication between DRACULA and BALANCE**

BALANCE uses the detector data it receives to optimise the signal settings. The detector bits are received and the signal force bits transmitted using the XDataOnTCPIP function.

Every second, DRACULA produced a list of SCOOT nibbles for all of the detectors in the network.

Every second the INTERFACE program sends detector bits to BALANCE and signal aspects to DRACULA.

Every second BALANCE outputs stage force bits messages, which are full 16-bit UTC control bit patterns, but only using the stage force bits. To cater for the multiple node architecture an item was added to the message to define which controller the following control bits relate to as in the following table:-

Byte No.	Data Content
1	Number of controllers (1 - 20 say)
2	1st controller id (0 - 19)
3	Control bit states 0 - 7, where 0 = bit inactive & 1 = bit active
4	Control bit states 8 - 15, where 0 = bit inactive & 1 = bit active
5	.....
?	Nth controller id (0 - 19)
?	Control bit states 0 - 7, where 0 = bit inactive & 1 = bit active
?	Control bit states 8 - 15, where 0 = bit inactive & 1 = bit active

A single four arm junction is SW London was used to test the operation of the DRACULA / BALANCE interface.

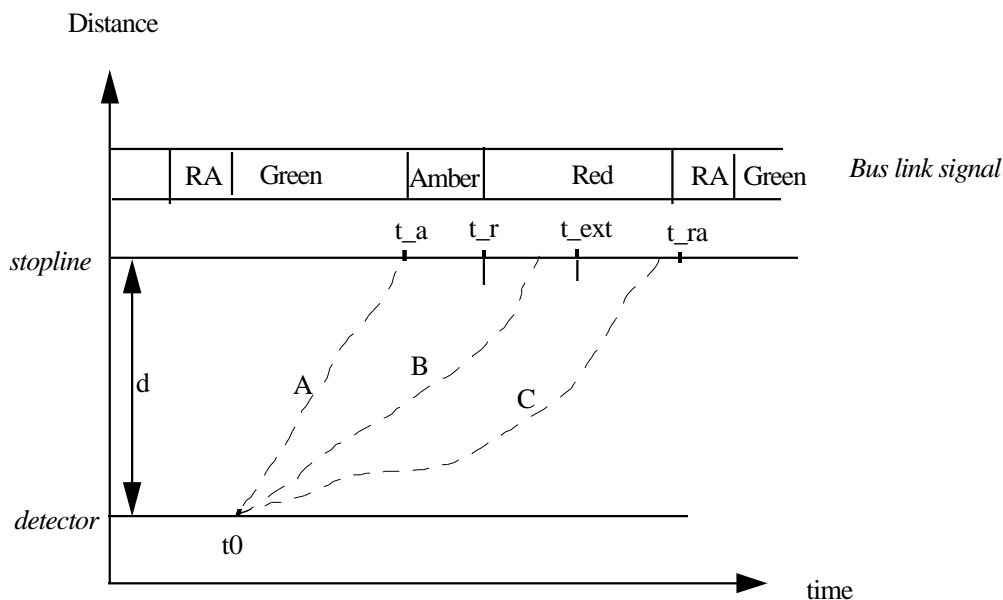
#### 4. Public Transport Priority

Apart from providing public transport with special reserved lanes, public transport is also given priority at signalised intersections. When a public transport vehicle is detected at time  $t_0$  and predicted to arrive at the stopline at time  $t_a$ , one of two actions may be performed:

- Extension, which extends the bus green period in order to allow the bus to exit;
- Recall, which terminates the bus red stage earlier in order to reduce the bus waiting time.

Public transport signal priority requires the modelling of selective vehicle detectors (selected to detect public transport vehicles only) and use of a simulation control parameter to switch the priority scheme on.

Figure 8 shows schematically the signal priority in a space-time diagram. The signals for the bus link are shown on the top, with  $t_r$  and  $t_{ra}$  representing the start and end time of the red aspect.  $ra$  denotes red/amber.  $t_{ext}=t_r+E_{max}$ , where  $E_{max}$  is the user specified maximum allowed extension (in second). The distance from the detector to the stopline is  $d$ . Three bus trajectories from the detector to the stopline are drawn in dashed lines.



**Figure 8: Space-time representation of bus signal priority**

If a bus is predicted to arrive at the stopline just after the start of the red signal (case B in Figure 8), the bus green aspect will be extended by just enough time to allow the bus to exit. The amount extended depends on the predicted bus arrival time, subject to a user-defined maximum ( $E_{max}$ ) and to minimum greens for the subsequent stages affected.

If a bus is predicted to arrive during the red, but an extension is not appropriate (i.e. requires more than the maximum permitted extension, case C above), then the duration of the bus red aspect may be reduced by a constant amount of 5 seconds. The length of other stages remains unchanged, so the length of the current cycle is decreased temporarily.

Otherwise the signals will not be changed (case A in Figure 8).

### 5. Detectors

Detectors in DRACULA have been modified to allow them to output the data produced by SCOOT detectors, which are common in the UK. The data consists of quarter second occupancy bits which are sent out every second as blocks of four bits.

Each detector in the network is defined by the following data:

- A unique integer identifier

- An 8 bit bit-mask to identify which of the eight DRACULA vehicle types the detector can detect
- Upstream node of the link on which the detector lies
- Downstream node of the link on which the detector lies
- Position of the detector on the link
- The lane in which the detector lies

The front and rear ends of a vehicle are compared to the location of a detector in the current lane the vehicle is travelling in, a detection is triggered if a vehicle passed or is stopped on a detector. The exact timing, in quarter second intervals, when the vehicle passed the detector is extrapolated based on the current speed the vehicle. This is because the simulation time increment is one second.

At every simulation time step the program loops through all detectors in the network and outputs all detections. The detector data is in the form of SCOOT nibbles. This is quarter second occupancy data which is sent every second. For each detector four bits of information are sent every second. The information is passed using bytes (i.e. 8 bits), so two detectors worth of data are sent with each byte. The SCOOT nibbles are created left to right, so the leftmost bit is for the first quarter second, the rightmost bit for the last quarter second.

#### 6. *Traffic Calming*

Traffic calming is represented as a special speed-controlled region in a link.

A traffic calming region is specified by:

- Upstream node of the link the traffic calming region locates
- Downstream node of the link
- Starting position of the region, measured from the link entry
- End position of the region, measured from either the link entry to from the stopline;
- Maximum speed over the region.

When approaching a traffic calming region:

If the current speed is more than the maximum speed of the region, then

Decelerate at a normal deceleration to the maximum speed of the region

else

Move at the car-following speed

end if

#### 7. *References*

Friedrich, B., Sachse, T., Hoops, M., Jendryschik, W. and Reichert, G. (1995) "BALANCE and VARIA Methods for Traffic Adaptive Control", Proceedings of the Second World Congress on Intelligent Transportation Systems, Yokohama, Nov 9th-11th 1995, pp 2356-61.

Toomey, C., Clark, M. and Friedrich, B. (1998) "Tipping the BALANCE: A European Trial of Advanced UTC", Traffic Technology International, Dec 97/Jan 98, pp51-54.

## Enhancement of the NEMIS micro-simulation model features

Gianni Canepari and Carlo Di Taranto

Mizar Automazione SpA, Via Vincenzo Monti, Torino, Italy

e-mail: [GianniCanepari@miz.it](mailto:GianniCanepari@miz.it)

### *Abstract*

Micro-simulation models are becoming an increasingly important tool for traffic control research and development. They permit the traffic engineer to have a “bird’s eye” view of the traffic in an urban area and an instant feel for current congestion problems and possible solutions. Controversial or new techniques can be tried and tested without any disruption to traffic in the real network.

As more complicated traffic control methods and systems are developed, the problem of testing and calibration becomes more and more acute. On-street evaluation is notoriously difficult for two reasons:

1. the day to day variability of traffic
2. the difficulty and expense of collecting enough data to arrive at a certain result.

NEMIS was designed as a specific solution to the problem of on-street testing. Its ability to model urban networks in microscopic detail (individual vehicles, single intersections or road sections etc.) makes it a valuable tool for testing traffic control strategies or techniques at local and area levels.

NEMIS is a scientific software package and, since its creation, it has been used principally for research and development work and for the technical assessment of traffic control strategies.

NEMIS has been developed for the micro-simulation of urban traffic (private and public) in large scale networks. It is capable of modelling urban networks and vehicle behaviour in considerable detail, and is well structured to meet a variety of application needs. Its usefulness has been demonstrated for the following tasks:

- analysis of the effects of regulation and network modification on traffic mobility
- evaluation of different traffic light control strategies
- testing of traffic assignment techniques
- simulation and evaluation of route guidance strategies and variable message systems.

Other areas where NEMIS may be used include:

- evaluating the effects of improved public transport facilities on inner city traffic flow
- testing the effectiveness of parking management systems
- examination of strategies aimed at reducing fuel consumption/exhaust emission.

### *1. Development history*

The combination of a detailed network topology and the ability to track the progress of each vehicle through the network accounts for the effectiveness of NEMIS as a powerful simulation

tool. Its accuracy has been verified through field trials and its application to large-scale projects in traffic control.

Like all useful systems, it has evolved over the years to meet changes in the urban environment and user requirements. The following is provided as a short history of its development:  
1974-1976: The NEMIS basic modules were developed by a team of researcher at IENGF (Istituto Elettronico Nazionale “Galileo Ferraris”) in Turin. Some of the team became part of the R + D staff at MIZAR Automazione S.p.A. and completed and upgraded the package.

1977: Analysis using NEMIS of the effects on urban traffic of regulation and network modifications in the central area of Turin.

1978: Analysis using NEMIS of the effects on urban traffic of regulation and network modifications in the entire urban network area of Alessandria, Italy.

1978-1980: Feasibility analysis of the “Progetto Torino” system. The latter is an innovative hierarchical and decentralised traffic control system operating in Turin. NEMIS was used to evaluate traffic light control strategies.

1988: The NEMIS package was made portable by transferring it from a UNIVAC mainframe onto IBM PS/2.

1990: Evaluation of a decentralised control strategy to support route guidance. This study required modification of NEMIS in order to simulate beacons and floating-cars.

1990: Evaluation by the Institute of Transport Studies, Leeds, UK of a control strategy to reduce blocking back during oversaturation.

Current research at MIZAR requires the use of NEMIS to evaluate local control strategies.

## 2. *Key Developments within SMARTTEST*

Within the framework of SMARTTEST some important enhancements have been introduced in NEMIS.

The main efforts have been spent to provide NEMIS with an improved and standardised interface suitable for the real time simulation of the following external Transport Telematic Applications:

- *Adaptive Traffic Signals*
- *Public Transport Priority*
- *Variable Message Signs*
- *Dynamic Route Guidance*

The new interface has been based on a standard TCP/IP communication protocol, which helps to connect the computer where the model runs to the network of computers where the external strategies operate.

Furthermore, two models have been improved:

- *Public Transport Services*
- *Vehicle Detectors*

All the enhancements have been implemented taking care of transferability aspects and have been tested onto specific cases gathered in the Torino test site.

In particular:



- Public Transport Services model has been enhanced with the introduction of a lay-by PT stops.
- Vehicle detector improvement concerned performance, error rate and breakdown occurrence.
- The validation of the standard interface concerns mainly operational aspects.

In the following sections attention is paid to the new standard interface solely, which represents the main objective of the developments. Description of the developments is provided making reference to the application that resulted more complex: interfacing of Adaptive Traffic Control and Public Transport Priority strategies to the NEMIS model.

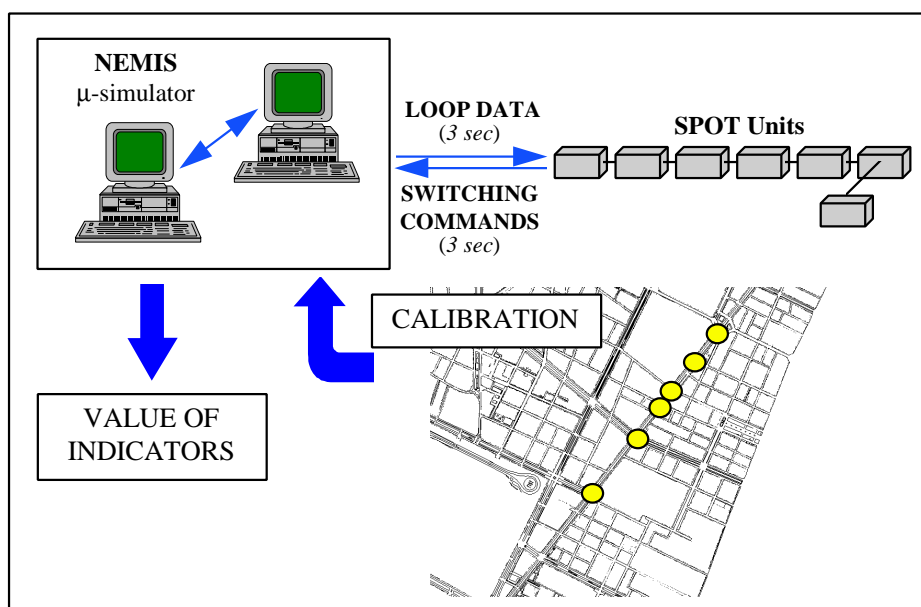
### 3. The New Interface

The calibration and setting up of UTC system on field can be extremely time-consuming and difficult unless intensive off-line tests are performed in a controlled environment.

The NEMIS software package was designed specifically as a tool for testing urban traffic control strategies prior to or in parallel with on-street testing.

NEMIS already supported an interface with external road-side processors (e.g. SPOT or SCOOT OTU) to test the effectiveness of urban traffic control systems as well as to screen strategies and tune system parameters before field installation.

*The old interface package needs to be installed on a MS-DOS PC connected by a serial line with the NEMIS computer and by another serial line with the WYHETS SCOOT system or to the SPOT MFOs<sup>1</sup> of the UTOPIA Network (Figure 9).*



**Figure 9 : Use of NEMIS as a system evaluation tool before field installation**

The old interface communication protocol operates using several serial connection (RS232) between the NEMIS computer, the MS-DOS PC where the interface package runs and the real system. This solution presents the following limits:

<sup>1</sup> MFO = Multi Functional Outstation

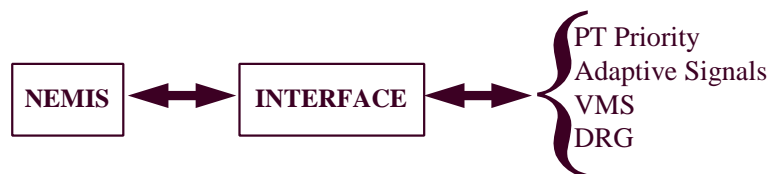
- limited communication capabilities (limited speed of serial connections)
- non-standard communication protocol (proprietary protocol "ad hoc" to interface NEMIS with SPOT and SCOOT)
- limited simulation capabilities (dedicated hardware is needed for the intersection controller)

The new interface overcomes these limits using a standard protocol based on TCP/IP in order to connect NEMIS with a LAN/WAN where the intersection controllers implementing external Traffic Control Strategies (e.g. MFOs SPOT) operate.

The new NEMIS interface provides users with a simplified and high efficiency micro-simulator suitable to investigate the impact of Advanced Transport Telematics functions (adaptive and co-ordinated traffic signals, public transport priority, VMS and Dynamic Route Guidance) on big road networks.

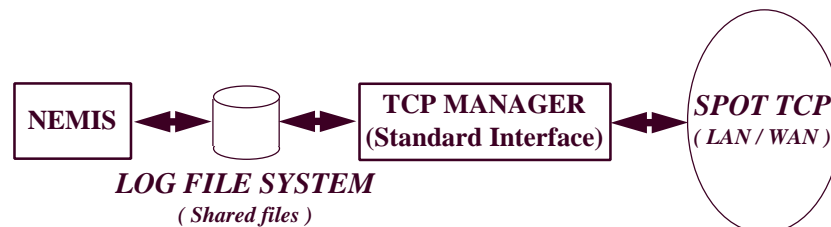
The main objective (as shown in Figure 10) was to have a new interface based on standard communication protocol TCP/IP and suitable to directly interface NEMIS with several external control strategies embedded in the UTOPIA SPOT units. In fact, the new interface is able to manage messages written directly in the UTOPIA format.

Anyway, it is also possible to use NEMIS in order to test any "ad-hoc developed" external control strategy. If this is the case, the external control strategy developed needs to exchange appropriated messages with NEMIS using for each message the proposed format.



**Figure 10: Target of the new interface package**

Starting from the physical approach shown in figure 2 the new interface has been developed following the architecture proposed in figure 11.



**Figure 11: Implementation of the new interface package**

The new interface approach developed for NEMIS is based on the following components:

- TCP MANAGER tools that manages the communications between NEMIS and the network where external control strategies are located
- LOG FILE SYSTEM set of circular files<sup>2</sup> where the messages that need to be exchanged between NEMIS and the TCP MANAGER are stored.

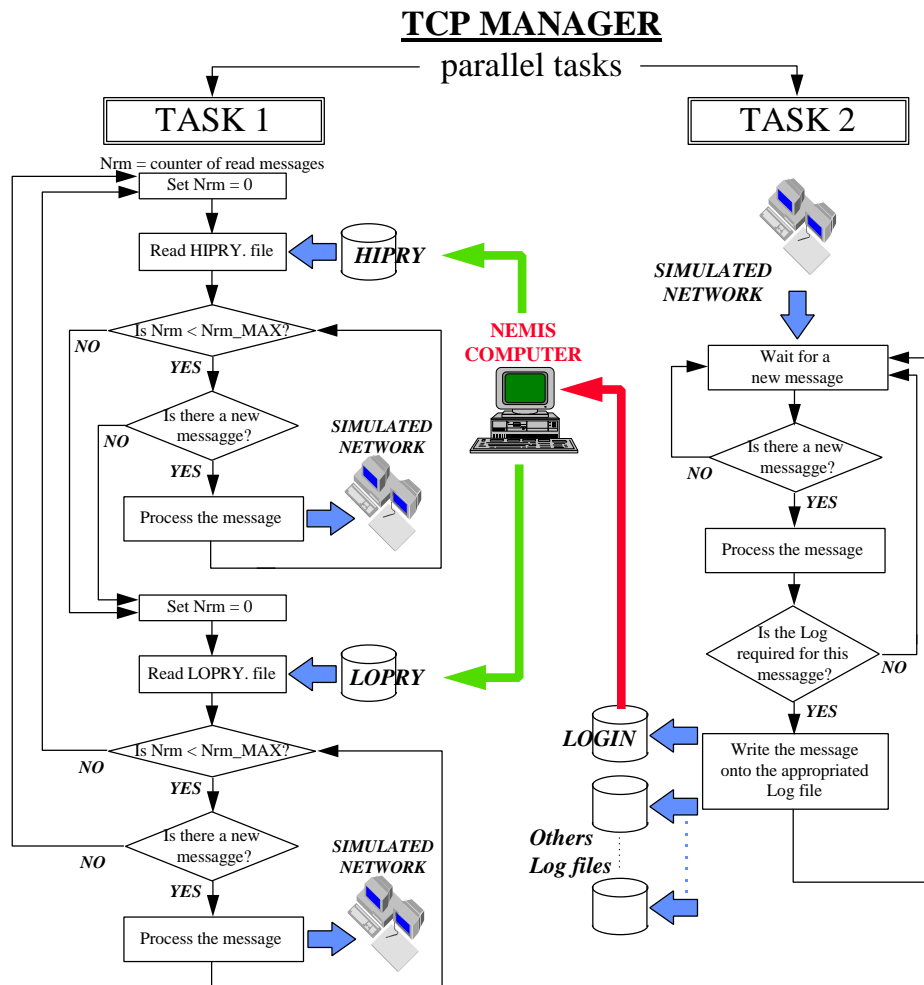
The operation of the new interface can be easily described by means of the following steps:

- The TCP MANAGER receives from the external control functions that reside within SPOT units or within others users developed packages, all the messages containing the elaborated control strategy. The communications between the TCP MANAGER and the external control functions are based on TCP/IP standard protocol.
- When a new message is received by TCP MANAGER, the message itself is processed and then written onto the appropriated file into the Log File System.
- The TCP MANAGER also reads the circular files HIPRY and LOPRY (two command files that belong to the Log File System - see following sections) where NEMIS writes the messages required by the external control functions.
- When a new message is written by NEMIS onto a command file of the Log File System, the TCP MANAGER, processes the message and then send it out towards the appropriated SPOT unit or to the appropriated external control function.

The flow chart in figure 4 shows in schematic way, the operation of the TCP MANAGER two main tasks and the interaction existing between NEMIS, the TCP MANAGER and the whole simulation network.

---

<sup>2</sup> Circular files format and operation are described with great details in the following section “4. *adaptive traffic signals and public transport priority*”



**Figure 12: TCP MANAGER operation**

The messages are written onto the Log File System using the LFS management functions provided together with the interface package. The same LFS management functions have to be used also to read the messages previously written onto the Log File System by NEMIS itself and/or by any other external operating tools.

Comparing the new interface to the old one, we can underline the following advantages:

- it uses a standard communication protocol
- the communication speed is limited only by the communication network features (rates greater than 10Mb/s can be achieved using optical fibre)
- the simulation does not need a dedicated hardware (a standard PC network can be used)

The new interface is described in more details in the following section “4. *adaptive traffic signals and public transport priority*”.

#### 4. Adaptive Traffic Signals and Public Transport Priority

##### 4.1 Introduction

Adaptive control strategies for private traffic and public transport priority services, are surely the basic aspects of the main systems operating in urban traffic control area.

Taking care of the growing interest showed by users for micro-simulator application related to the capabilities to simulate the main telematic functions for traffic management, it comes that new millennium micro-simulators must provide users with all the instruments needed for the simulation of these fundamental aspects.

It also clearly appears that state of the art technology proposed for the implementation of adaptive control and public transport priority strategies, as the control strategies themselves, are far from being standardised.

Hence, together with the need of micro-simulators able to investigate adaptive control and public transport priority strategies performance, grows the necessity to keep externally to the micro-simulator itself all the software module implementing the control strategies.

From this consideration the idea was born to provide users of the NEMIS micro-simulator with a tool able to interface, directly and in a easy way, the external control strategies embedded in the SPOT unit (local multifunctional unit of the UTOPIA UTC system). The approach adopted provides to the micro-simulator NEMIS the capability and the tools necessary to evaluate the impact of particular adaptive control and public transport priority strategies (those implemented by UTOPIA) without prevent the simulation of control strategies ad-hoc developed by the user itself.

In the following a brief introduction to the UTOPIA integrated system and to the adaptive control and public transport priority strategies implemented by the SPOT peripheral units at the intersection level is provided.

#### 4.2 Some notes about the UTOPIA UTC system

UTOPIA (Urban Traffic Optimisation by Integrated Automation) is a specific concept designed to improve urban travel conditions by the application of fully automated control principles. The concept is based on a particular system architecture and particular control strategies. It was originally conceived to provide an innovative response to two fundamental requirements of wide-area traffic control systems:

- significant improvements in private vehicle mobility in all traffic conditions
- the assignment of absolute priority to selected public transport vehicles at traffic light intersections

UTOPIA control strategies aim to reduce significantly the total time lost by private vehicles during their trips within the controlled area, subject to the constraint that public vehicles requiring priority shall not be stopped at intersections with traffic lights.

Field trials have shown that systems based on the UTOPIA concept can meet both the above requirements simultaneously<sup>3</sup>.

The UTOPIA architecture is hierarchical and decentralised: optimal control strategies are determined at the higher level on the basis of area traffic prediction, while traffic light control is actuated at the lower level according to the actual traffic conditions encountered at the intersections.

UTOPIA control strategies overcome the unmanageable complexity of the area optimal control problem by a process of decomposition into several simpler, interrelated sub-problems. Control

---

<sup>3</sup> UTOPIA - Technical Reference Manual

problem decomposition, the choice of suitable functionals for the resulting sub-problems and the introduction of rules for sub-problem interrelation permit both a hierarchical and decentralised architectural solution to the original control problem and provide control actuation that is really close to optimum. Decomposition is performed by following a topological rule: firstly, the area is subdivided into ‘overlapping’ zones, where each zone is logically centred on a controlled intersection and includes neighbouring intersections. Then an optimal control problem is defined for each zone.

The zone control problem concerns the traffic control to be actuated at the central intersection only, but it provides a consistent interrelation with the control of the neighbouring intersections and takes into account traffic information and traffic control data concerning the whole zone. The functional to be optimised consists of the terms relating to the traffic observed on the incoming link of the central intersection and those which implement the following two fundamental interaction principles:

- a *strong interaction principle* is implemented by taking into account the time lost at the downstream intersections by vehicles leaving the central one;
- a *look-ahead principle* is implemented by taking into account traffic forecasts defined on the whole optimisation horizon (120 sec) for all the central intersection’s incoming links.

In order to guarantee stability and robustness at the network level, interactions are provided with a higher level, where an area optimal control problem is defined on the basis of the area traffic macroscopic model.

The final result is a series of problems which can be classified as belonging to two different classes: the ‘intersection level’ and the ‘area level’. At both levels the problem formulation is based on the definition of two fundamental modules: the *state observer* and the *controller*.

Control problems are solved by a network of interconnected control units which apply suitable Open Loop Feedback Optimal (OLFO) techniques.

Step back to the new interface approach, it is based onto the following issue:

- the LogFile System set of circular files that are used to storage the messages exchanged between NEMIS, the TCP Manager and the SPOT units network
- the Library for the Log File System management Library that contains the FORTRAN functions used by NEMIS in order to manage the access to the circular files set, and the operation of reading/writing messages onto circular files
- the TCP MANAGER a front-end TCP that, manages the communication interface of the whole UTOPIA system and the data exchange between NEMIS and the local controller unit (SPOT) of the simulated network

#### 4.3 Processing

From the point of view of the NEMIS model adaptive control and P.T. priority are external strategies, that is tasks embedded within the local unit SPOT of the UTOPIA System.

In this section a description of the new interface is provided that allows users to implement the information exchange between the external modules where the control strategies reside and the micro simulator NEMIS

This section contains the following four subsections:

- The Circular files
- The exchanged messages and their format
- The Log File System Management Library
- The simulation process

4.5 The circular files

Figure 13 describes the interface between the SPOT units (unit where the software module that implement the external adaptive control and public transport priority strategies resides) and NEMIS.

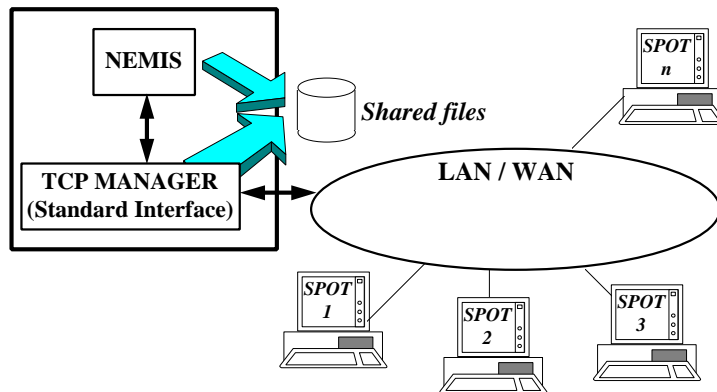


Figure 13: Scheme of the new interface

The communication take place by means of messages logged by the TCP Manager onto a set of circular files: the Log File System. The term circular files is referred to a set of logical structures that allow sequential accesses, in which the first record logically follows the last one. Circular files are used to storage into each record the messages that are exchanged between the various elements of the whole simulation system, that are all the messages produced by NEMIS, by the SPOT units and by possible other running tools of the UTOPIA system (users interface tools).

Within the circular files, records are sorted on the basis of logged messages date and time.

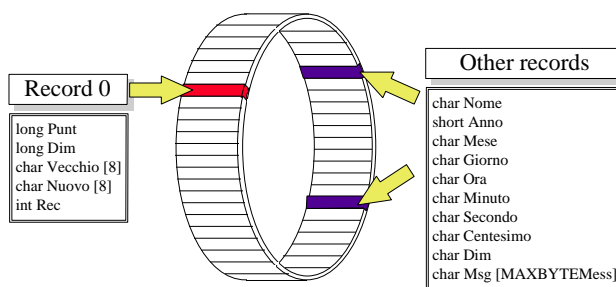


Figure 14: Circular file elements

Within the Log File System, NEMIS and the TCP manager exchange information using the three following circular files:

- **HIPRY.**      High PRioritY messages      contains messages 78 (PT forecasts) written by NEMIS.  
The TCP Manager reads the records in this file with a time period user defined (default = 0.5 sec)
- **LOPRY.**      Low PRioritY messages      contains messages 93 (detector counts) written by NEMIS  
the TCP Manager reads the records written in this file with a time period user defined (default = 0.5 sec)
- **LOGIN.**      LOG INput messages      contains messages 91 (“planned” signal plan) and messages 111 (synchronisation messages) written from the TCP Manager.  
NEMIS reads this file searching for synchronisation messages (MSG 111) and “planned” signals plan messages (MSG 91)

The exchanged messages and their format

The messages required to implement the simulation of adaptive control and public transport priority strategies are the following:

- **MESSAGE 78: PUBLIC TRANSPORT FORECASTS (for P.T. PRIORITY)**  
This message communicates to the external control strategy the arrival time forecast for the P.T. vehicle. It can also contain data related to the travel time between detector and stop line.  
It is logged onto the circular file “HIPRY.” of the Log File System
- **MESSAGE 93: PRIVATE TRAFFIC DETECTORS (for ADAPTIVE TRAFFIC SIGNALS)**  
This message communicates data related to the traffic counts of the detectors directly connected to the SPOT unit.  
It is logged onto the circular files “LOPRY.” of the Log File System
- **MESSAGE 91: STAGE PLANNED (for ADAPTIVE TRAFFIC SIGNALS / P.T.PRIORITY)**  
This message is sent out to the upstream intersection controller (that needs to know the future strategy planned by the downstream intersections in order to achieve the strong interaction principle) and to NEMIS, so that it can vary the matrix for the traffic light description taking care of the adjustment in the control strategy.  
It is logged onto the circular file “LOGIN.” of the Log File System
- **MESSAGE 111: SYNCHRONISATION MESSAGE (for ADAPTIVE TRAFFIC SIGNALS)**  
This message is sent out to NEMIS to start a new simulation period.  
Every simulation period lasts 3 seconds (1 STEP) and next simulation period starts when a new message 111 is received from the TCP MANAGER.



It is logged onto the circular file "LOGIN." of the Log File System

All the messages in UTOPIA format are preceded by a 4 byte header that contains the following information:

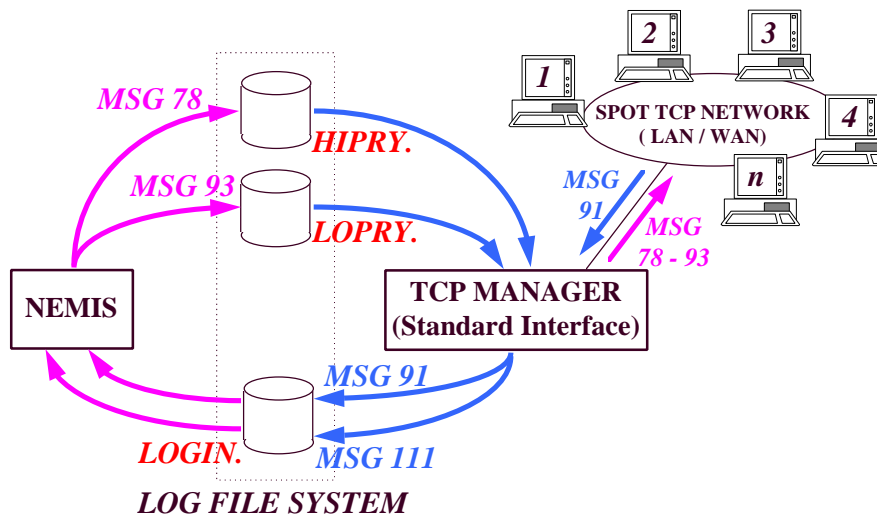
Message Code	1 byte
Origin Micro	1 byte
Destination Micro	1 byte
Step/Cost	1 byte

Many run-time messages contains, after the header of the message itself, the send time of the message on 3 bytes. This time is expressed in seconds by midnight.

If there are no different indications (message 93) the time bytes are ordered as follows:

LSB - MSB - MSB.

Figure 15 summarises the concept expressed above.



**Figure 15: Description of the Log File System and exchanged Messages**

#### 4.6 The Log File System Management Library

The Log File System LIBrary manages the Log File System files. When a message has to be logged on a file of the Log File System, the function of the LFS LIBrary control the consistency of the recording date and time.

The system date / time variation are also managed by the library in order to avoid Log files corruption.

The main function of the LFS LIBrary are the following:

- LFSOpen
- LFSClose
- LFSLock
- LFSUnLock
- LFSInfo

- LFSRead
- LFSWrite

4.7 The Simulation Process

The flow chart of Figure 8 shows in a schematic way, the operation of the TCP MANAGER two main tasks and the interaction existing between NEMIS, the TCP MANAGER and the whole simulation network.

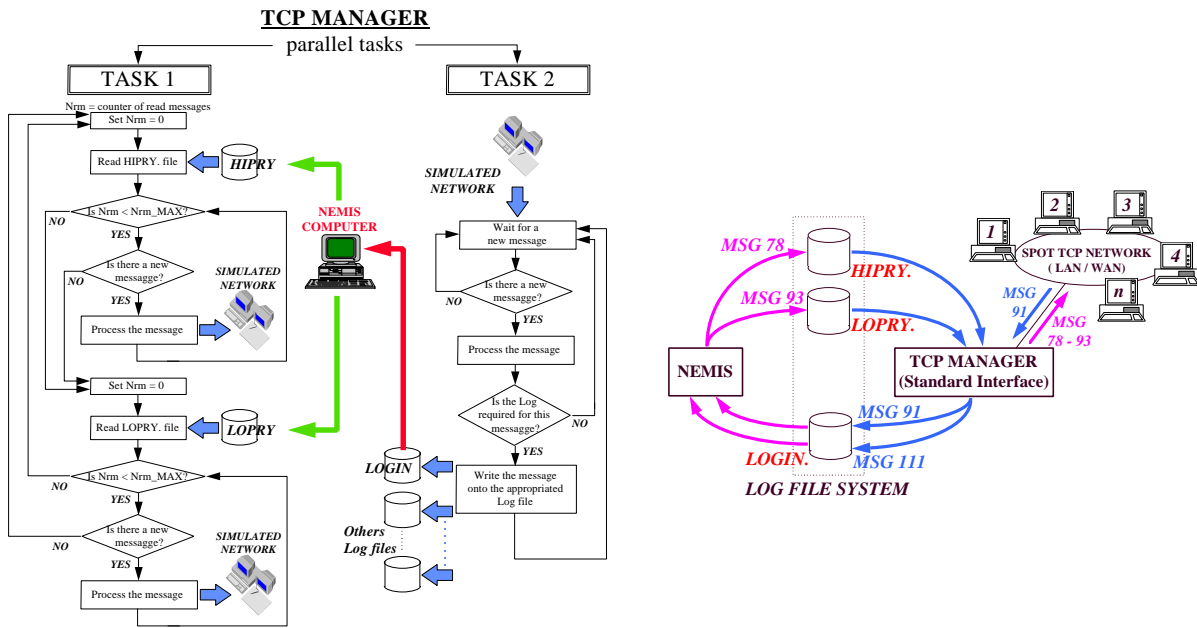
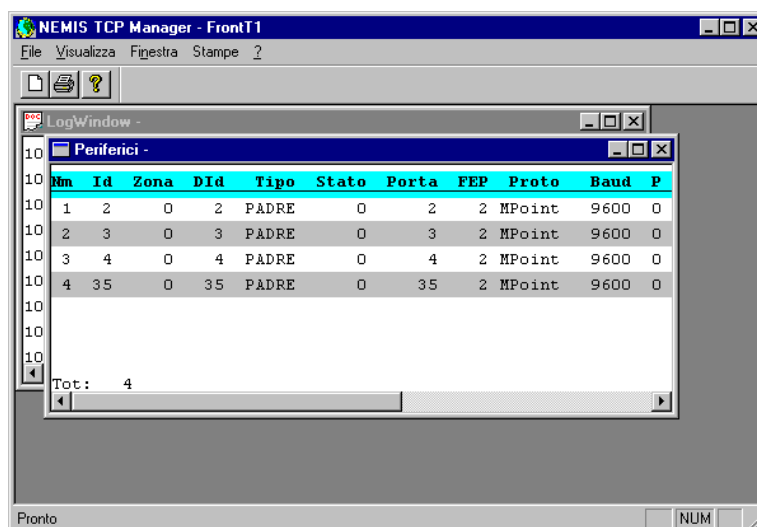


Figure 16: TCP MANAGER behaviour, Log File System and exchanged Messages

The communication between the TCP MANAGER and NEMIS, can be summarised through the following steps:

- The TCP MANAGER receives from the external control functions that reside within SPOT units or within others users developed packages, all the messages containing the elaborated control strategy (Message 91).
- The communications between the TCP MANAGER and the external control functions are based on TCP/IP standard protocol.
- When a new message is received by TCP MANAGER, the message itself is processed and then written onto the appropriated file into the Log File System.
- The TCP MANAGER also reads the circular files HIPRY and LOPRY where NEMIS write the messages needed by the external control functions (Messages 78 and 93).
- When a new message is written by NEMIS onto a command file of the Log File System, the TCP MANAGER, processes the message and then send it out towards the appropriated SPOT unit or to the appropriated external control function.

Figure 9 shows a screen-shot of the TCP Manager user interface.



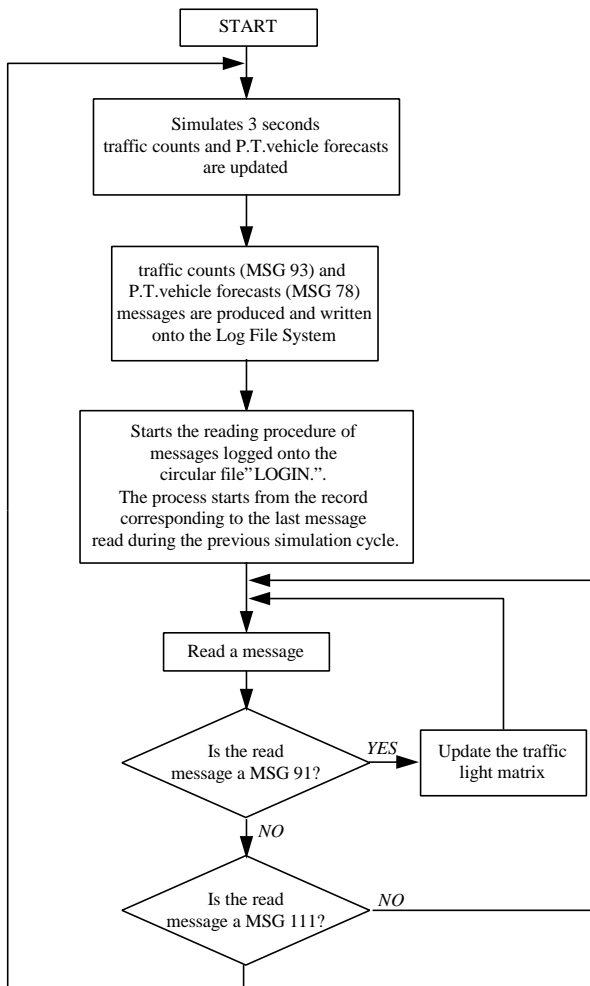
**Figure 17: User interface of the NEMIS TCP Manager**

The way in which the simulation process changes when NEMIS is linked with external control strategies can be described as follows:

1. NEMIS simulates the three seconds and then:
  - prepares the messages 93 (traffic counts for external adaptive control strategy)
  - prepares the messages 78 (P.T. vehicle forecasts for external P.T. priority strategy)
2. NEMIS writes onto the Log File System the messages previously prepared
  - all the messages 93 are written onto "LOPRY." file
  - all the messages 78 are written onto "HIPRY." file
3. At the end of the simulation period, NEMIS read the circular file "LOGIN." starting from the last message read and looking for messages 91 coming from the SPOT local units and for message 111 coming from the TCP Manager.

4. When a message 91 is detected the SEM6 matrix (containing the traffic light information) is updated with the new planned plan information.
5. When a message 111 is found, the new simulation period (that lasts three seconds) starts and the simulation process returns to the step 2.

Figure 18 shows a flow chart for the simulation process.



**Figure 18: NEMIS simulation process**

## The SITRA-B+ Microscopic Traffic Simulation Model - Examples of Use and Future Developments

Jean-François GABARD\* and Laurent BREHERET\*\*

\* ONERA/CERT - 2 av. Edouard Belin, 31055 TOULOUSE Cedex - FRANCE

\*\* SODIT - 2 av. Edouard Belin, 31400 TOULOUSE - FRANCE

### Abstract:

SITRA-B+ is a microscopic traffic simulation model which was primarily developed by the research scientists of ONERA in order to provide them with a tool capable of testing, comparing and tuning Urban Traffic Control Systems ; its use was then extended to complementary application fields, like Public Transport Priority, Collective or Individual Route Guidance Systems. As one of the few selected tools of the SMARTEST EEC Project, it has benefited of some improvements and extensions identified from users and developers needs, such as a more flexible and universal interface with UTCS systems and a better modelling of traffic in roundabouts.

### 1. Introduction

The use of simulation techniques, which appeared in the early 1950s in the field of transportation science, can now be considered as a necessary step as far as people who deal with control strategies design or operation want to know if those strategies are worth implementing.

This is particularly relevant for traffic management, because of the nature of the process to be controlled : the high level of complexity and high degree of interaction between its elementary components - the vehicles - leads to a quasi impossibility of using classical analytical approaches to produce optimal solutions.

Among other key features of the traffic simulation techniques, one worth to be mentioned is obviously its ability to be used for making unbiased and cheaper comparative assessment of various strategies, before implementing them in the field : one of the first studies conducted in CERT with the SITRA-B simulation model was dealing with the comparison of traffic responsive and adaptive fixed-time traffic lights control strategies (Gabard *et al.*, 1982).

The first version of SITRA-B, which was developed in FORTRAN in the 1970s (Correge *et al.*, 1977), was able to model two types of vehicles (cars and buses) on simple urban networks (2-lane links, 4-arm intersections) ; the car-following law is a linear one, coupled to a lane changing model to enable bus stops modelling. A simple graphical display was already introduced in this early version, which enabled to visualise vehicle movements and state of traffic lights, and therefore to check the model correctness ; it is worth noting that, at this stage, the strategies to be tested were completely embedded in the simulation code.

It then rapidly appeared that, because of the increasing number of strategies to be tested, the new application fields linked with the emerging Telematics functions, and of course the need to address more complex network layouts, the simulation program architecture had to be renewed. An opportunity to fulfil those objectives was given by European DRIVE Project ASTERIX

(Barcelo, 1991), the aim of which was to develop a general purpose traffic simulation software, including different modelling levels.

The C++ object oriented language was chosen for the new release of SITRA-B, which so became SITRA-B+ ; among its new associated features, which will be described in more details in the next chapter, we can especially quote the ability to design complex network layouts, to consider more vehicles categories (as for example guided and unguided vehicles), to model various types of detectors. A Sun station and the UNIX environment were chosen to host this new version.

The third step which was achieved with the SITRA-B+ development process took place within the frame of the SMARTTEST DG VII European Project (Hugosson *et al.*, 1997) ; after having investigated the user needs and matched them against identified gaps of existing models, this project led at enhancing a set of representative micro-simulation tools and incorporating the resulting findings in a « best practice manual » for dissemination throughout Europe.

For SITRA-B+, which was one of those selected tools, CERT and SODIT were working together in order to achieve a set of complementary goals : on one hand, the basic modelling capabilities of SITRA-B+ were increased (roundabouts and Public Transport services modelling, improved host architecture for adaptive traffic signals and PT priority strategies), and on the other hand, the user interface was expanded and a new implementation now enable to run SITRA-B+ on PC computers operating under Windows NT.

After a brief description of the main characteristics of the simulation tool, we give a series of representative application examples, addressing both UTC and route guidance strategies ; then we present two of the main achieved enhancements which were held in the frame of the SMARTTEST Project, related to roundabouts modelling and adaptive traffic signals management.

## 2. GENERAL FEATURES OF THE MODEL

We describe here the main characteristics, of the SITRA-B+ model ; a User's Manual (Pommier *et al.*, 1992) gives complementary information on the model and data files structure.

### 2.1. Network description

Network layouts addressed by SITRA-B+ are essentially urban ones ; they include complex intersections, traffic lights, vehicle detectors and communication beacons ; the size of typical applications ranges from 10 to 30 intersections, 30 to 100 links.

A two-level structure is used to describe a network ; first, a « macroscopic » description considers nodes and links, where a node can be either an intersection or an entry or output node ; then, at the « microscopic » level, connection points and lanes enable to define traffic rules ; two types of lanes are used by the simulation model : those belonging to a link, which can be reserved lanes (bus lanes for example), and where lane changing can occur, and those associated to an intersection, called « internal lanes » : figure 1 shows two representations of the same intersection, with and without the internal lanes structure (the dots represent traffic lights) ; internal lanes enable to specify the vehicles routes inside the intersection, and lane changing is not possible between such lanes ; conflicts between internal lanes are managed by the model.

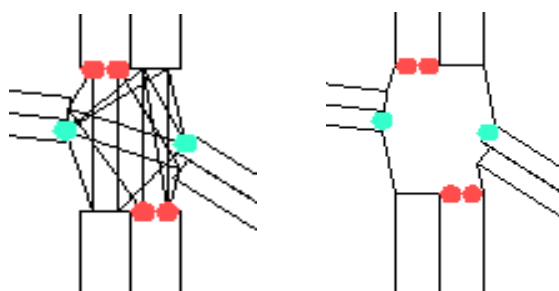


Figure 1

## 2.2. Car-following and lane changing model

Different categories of vehicles can be modelled by SITRA-B+ ; they differ on one hand by their geometric and kinematics description and on the other hand by their equipment level (guided and non-guided cars).

Among the wide area of car-following models studied and proposed by researchers (see Gabard, 1991), the Helly model (Helly, 1961) was selected to describe the behaviour of the driver-vehicle system. With this model, the acceleration of the follower car  $\dot{x}_{n+1}$  is given by the following expression :

$$\dot{x}_{n+1}(t+T) = c_1[\dot{x}_n(t) - \dot{x}_{n+1}(t)] + c_2[x_n(t) - x_{n+1}(t) - D]$$

where  $T$  is the reaction time of the vehicle-driver system (typically 1 second in SITRA-B+, which is also the elementary time step of the model),  $c_1$  and  $c_2$  are the relative velocity and headway control parameters and  $D$  the desired headway ; the desired headway is given by :

$$D = l_n + t_{n+1} \dot{x}_{n+1}(t)$$

with  $l_n$  being the length of vehicle  $n$  and  $t_{n+1}$  the time headway for vehicle  $n+1$ .

In combination with this car-following model, two kinds of cinematic parameters are associated to each vehicle : deterministic ones (acceleration and deceleration rates and vehicle length, which enable to distinguish the different vehicle categories), and stochastic ones (as the desired speed, which enable to model speed variability among drivers).

As far as lane changing is concerned, the model takes into account headways and relative velocity of neighbour vehicles on the destination lane ; the choice of the destination lane depends on the possibility for the vehicle to achieve its next turning movement at the downstream intersection.

## 2.3. Vehicle generation and traffic description

For each vehicle category, traffic demand can be described by the two following ways :

- a set of input flows at entry nodes and turning movements at each intersection
- an Origin-Destination matrix, which then has to be processed either by a simplified internal assignment procedure, or by an external one, in order to feed SITRA-B+ with a set of proposed routes and associated flow ratios.

## 2.4. UTC and guidance strategies

Except in the case of simple fixed time plans, we chose to consider traffic light control strategies as well as routing strategies as external software modules ; this choice is motivated by the wide variety of possible strategies to be tested.

For data exchange between the simulation model and the external strategy, a simple communication protocol, using synchronisation files was selected ; this decision was taken in order to avoid constraining strategy developers by setting too strong development rules ; on the other hand, dedicated interfaces have to be developed on both sides.

## 2.5. Inputs and outputs

If no network editor is yet available for scenario building (a set of input files, associated to both macro- and micro-description of the network have to be filled by the user), a dynamic and interactive graphical display is provided : it enables to check and monitor the progress of each scenario, by offering the possibility to access all parameters of each static or dynamic entity ; two display modes can be chosen by the user : a macroscopic one, where mean vehicle concentration is visualised per lane, and the usual microscopic one, showing individual vehicles movements.

A set of complementary output files puts together the various results and usual traffic indicators, in order to achieve performance analysis.

### 3. APPLICATION EXAMPLES

A great number of research works were conducted by the Transport Research Group of ONERA/CERT, using SITRA-B+ as an evaluation tool, and more recently by SODIT ; they can be separated in two main categories : the first one deals with UTC strategies, and more precisely with the PRODYN (Farges *et al.*, 1991) and PRODYN-BUS (Henry *et al.*, 1994) real-time traffic light control strategies, and the second one is related to the evaluation of information and guidance strategies.

Among recent studies, we can for example quote, concerning PRODYN, the investigation of new control approaches using neuro-fuzzy techniques (Henry *et al.*, 1997), the evaluation of queue length estimation algorithms (Gallego *et al.*, 1997) and of turning movement ratios (Kessaci *et al.*, 1991), and, for bus priority strategies, a study conducted within the LLAMD EEC Project (Farges *et al.*, 1994).

In the field of driver information and guidance strategies, the first studies involving SITRA-B+ as an assessment tool occurred in 1992 (Arzan *et al.*, 1992), which were then extended towards the bi-directional route guidance systems using flow and travel time data fusion (Barbier *et al.*, 1997).

Three illustrative examples, related to recent or on-going work, are presented below ; screen copies of the Sun-Motif interface showing the network layouts associated to each of those three applications can be found in the Annex.

The objective of the first application example was to study the feasibility of a new bus traffic management scheme in the city of « Les Lilas » in the Ile de France area, in order to ease and make more secure passenger exchange between buses and metro ; PRODYN and PRODYN-BUS strategies were compared to fixed time plans ; simulation results showed that the new layout was able to deal with the traffic demand, and pointed out that this proposed layout was rather sensible to driving rules obedience (risk of intersection blocking, due to the presence of short links - see figure A1 in Annex) ; this study was sponsored by the Conseil Général Seine St Denis.

With the second example, the French Ministry of Transport wanted to evaluate the performance degradation caused by the loss of messages in a bus priority system used in the city of Pau ; figure A2 shows an axis, including complex intersections, where a bus priority traffic light control strategy was tested, with and without messages loss, and compared to a strategy without priority ; the conclusion of this study was that the performance degradation was low, and that traffic lights stages sequencing considerations could lead to significant benefits.

Figures A3 and A4 illustrate a study conducted in the frame of the ATT project CLEOPATRA (1996-1998), the aim of which was to develop and test methods and strategies for driver information and guidance ; figure A3 shows the Toulouse urban subnetwork which has been modelled with SITRA-B+, and figure A4 its southern part, with the « Grand-Rond » roundabout ; the traffic demand description, including initial assignment, was calibrated using surveys conducted by the city of Toulouse. Two strategies were tested with SITRA-B+ before their on-site demonstration : a global Variable Message Sign strategy, and an In-Vehicle Information System. The first one computes guidance recommendations, expressed as « turn right » or « turn left » messages to be displayed at signs installed at the entrances of the modelled network, and the second one uses communication beacons for short range transmission towards vehicles equipped with Euroscout guidance devices ; in this last case, the simulated scenarios consider several ratios of equipped vehicles (from 0.1% to 5%).

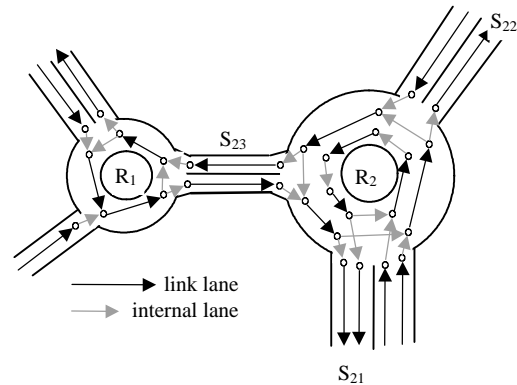


#### 4. Achieved improvements

##### 4.1. Roundabout modelling

Roundabout modelling was possible in the previous version of SITRA-B+, but in an imperfect way, both from the network description point of view and from the driver-vehicle behaviour one. It was then decided to improve those both aspects with the following specifications.

Concerning roundabout layout description, figure 2 shows an example of two interconnected roundabouts, a single-lane one ( $R_1$ ) and a 2-lane one ( $R_2$ ): the two types of lanes are represented on this network configuration: the link lanes, which can be found on entry and output links and also in the roundabouts, between which lane changing is possible, and the internal lanes which are used to specify the authorised turning movements: for example, on roundabout  $R_2$ , a vehicle approaching the 2-lane exit  $S_{21}$  on the inner roundabout lane can exit, provided that it will use the left lane of  $S_{21}$ , whereas exiting is forbidden for a vehicle approaching exits  $S_{22}$  or  $S_{23}$  on the same inner roundabout lane.



Two separate models are considered to ensure a realistic driver-vehicle behaviour :

- a lane choice model, in order to first distribute vehicles on entry arms taking into account their destination in the roundabout and also queue balance in case of congestion, and second for lane changing inside the roundabout (in relation with the turning movement constraints defined above)
- a gap acceptance model, whose principle is illustrated by figure 3 : two distance thresholds,  $D_{min}$  and  $D_{max}$ , which are calculated considering stopping distance of approaching vehicles, define an « acceptance area » for the incoming vehicles ; an « aggressiveness » parameter is attached to each incoming vehicle, increasing with waiting time, and leading to smaller gaps in case of congested traffic conditions.

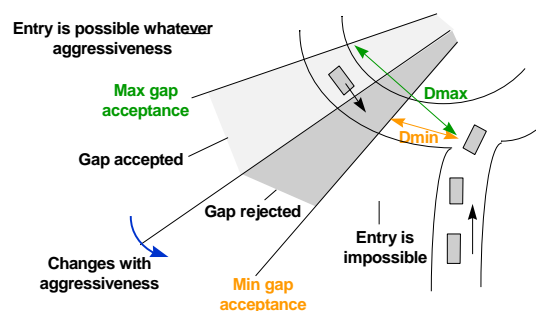


Figure 3

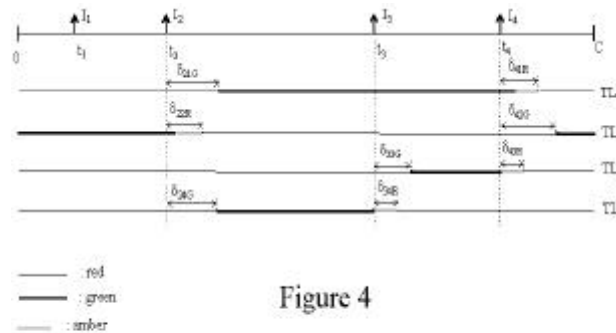
The graphical interface was modified in order to ease the roundabout description, and curvilinear links can now be displayed ; complementary traffic indicators, such as statistical distribution of speed in the roundabout and of waiting time in the entry links were produced.

##### 4.2. Adaptive traffic signals

In order to allow SITRA-B+ to be linked with the widest range of traffic lights control strategies, we proposed new specifications for traffic lights modelling and sequencing inside the simulation process ; the idea was to consider, as potential strategies, not only the most sophisticated real-time ones, but also simpler ones like plan changing strategies, as previous studies already showed that the choice of plan changing methods could have a significant influence on the traffic.

Considering first traffic light modelling, the initial set of « colours » (red, amber, green) was extended : traffic lights can for example display an « arrow » enabling right turn, blinking amber, combination of elementary colours, ... Each of those traffic lights states can be associated to a given driver-vehicle *behaviour*, which includes for example the associated give-way rules.

In order to cope with any traffic signals management method, we proposed a traffic lights sequencing description illustrated on figure 4 :



This representation, close to the stage-based approach, but offering more flexibility in order to cover adaptive signals, uses the following notation :

$C$  : cycle length, in seconds

$I_j$  : stage impulse number  $j$

$t_j$  : time of stage impulse number  $j$

$TL_i$  : traffic light number  $i$

$\delta_{jiG}$  : time-lag between stage impulse number  $j$  and effective green colour on traffic light  $i$

Two types of traffic light colours can be distinguished : « stable » colours (to which belong the major part of them) and « transient » colours, an example of which is given on figure 4 with amber : a « transient » colour is a colour which can occur at the transition between two « stable » colours (from green to red in this example) ; a given « transient » colour duration can be associated to an intersection or to a traffic light ; in order to cope with particular sequencing situations, a supplementary « stable » colour identifier has to be considered : the « no change » colour, which means to keep with the present colour.

Finally, to achieve the description of the complete sequencing, we associate to each couple ( $TL_i$ ,  $I_j$ ) two parameters : the next stable colour to be installed, and the corresponding time-lag ; in the case of traffic light #1 on figure 4, we would obtain :

$(TL_1, I_1) \rightarrow \text{red}, d_{1R}$

$(TL_1, I_2) \rightarrow \text{green}, d_{1G}$

$(TL_1, I_3) \rightarrow \text{green}, d_{31G}$

$(TL_1, I_4) \rightarrow \text{red}, d_{1R}$

Colours distribution and sequencing over traffic lights must comply with a set of constraints which include *amber duration*, *conflict clearance times*, *minimum* and *maximum green times*, which is usually made by the intersection controller. For example, time-lag parameters appearing on figure 4 must verify :

$$d_{22R} \geq \text{amber duration}$$

$$d_{1G} - d_{22R} \geq \text{conflict clearance time}$$

The external strategy has anyway to produce traffic lights settings which verify the above mentioned constraints, and therefore to know them ; we thus assume that, except for *amber duration*, the other constraints are embedded in the external UTC strategy.

In order to be able to cope with the widest set of UTC strategies, a high degree of flexibility has to be offered by the data exchange architecture and interfacing system ; this could mean that a strategy would be able to control each traffic signal at the lowest level ; in order to avoid, when it is possible, an unnecessary and too dense data flow, it seems better to introduce the traffic controller entity on the traffic simulation side, which will be able to store and execute a given sequence of predefined stages through the impulse sequence defined above.

The adaptive signal specifications given above allow to run any default fixed time plan on the traffic light intersections : the controllers associated to each intersection simply execute its current stage settings, which is in fact the initial working procedure .

Concerning the data flow exchange between SITRA-B+ and the external UTC strategy, two types of messages can be sent by the strategy to the simulator : « impulse » type messages and « plans » type ones :

- when the SITRA-B+ traffic controller receives an impulse type message, this means that an adaptive strategy is now running on this controller, so that it should no more execute its predefined stage settings, but wait for the next impulse message.

- in case of an adaptive strategy based on a series of fixed time plans, and in order to minimise the data exchange rate, the plan type tells to the controller that it now has to take control of the intersection by executing the new stage settings defined in this message, until reception of a new « impulse » type message, which will in this case announce a new plan change.

## 5. CONCLUSION

The application field of microscopic traffic simulation models, which was first mainly dedicated to UTC strategies testing, is now extending towards a wider range of applications, such as all dynamic traffic management problems linked to incidents, congestion, road works, information and guidance strategies, and also to network layout evaluation such as reserved lanes, bus priority systems and intersection design.

Some of those models, like SITRA-B+, who were first designed as pure research tools, needed to evolve and to meet identified user needs, in order to attract a larger audience among them. Research projects like SMARTTEST offer a good opportunity to achieve these goals.

## Acknowledgements

This work is based on research conducted in the European ASTERIX and SMARTTEST EEC Projects.

## References

- Arsan M.I., Charbonnier C. and J.L. Farges (1992). Développement et Test en Simulation d'un Système Dynamique de Guidage des Véhicules. *Recherche - Transports - Sécurité (Revue de l'INRETS)*, n° 33, pp.53-60.
- Barbier M. and J.L. Farges (1997). Flow and Travel Time Fusion in Bi-directional Route Guidance Systems. *Proceedings of the 8<sup>th</sup> IFAC / IFIP / IFORS Symposium on Transportation Systems*, Chania, Greece, Vol. 1, pp. 421-426.
- Barcelo J. (1991). Software Environments for Integrated RTI Simulation Systems. *Proceedings of the DRIVE Conference Advanced Telematics in Road Transport*, Brussels, Vol. II, pp. 1095-1115.
- Correge M., Gabard J.F., Henry J.J. and J. Tuffal (1977). Programme de Simulation Microscopique de Trafic Urbain SITRA B. *Journée AFCET « Simulation dans les Transports »*.

- Farges J.L., Kamdem I. and J.B. Lesort (1991). Realization and Test of a Prototype for Real Time Urban Traffic Control. *Proceedings of the DRIVE Conference (CEC-DG 13 (Ed.))*. Brussels, Vol. 1, pp. 527-542.
- Farges J.L. and J.J. Henry (1994). CELTIC Bus Priority in Lyon. *7<sup>th</sup> International Conference on Road Traffic Monitoring and Control*. IEE, London
- Gabard J.F., Henry J.J., Tuffal J. and Y. David (1982). Traffic Responsive or Adaptive Fixed-Time Policies ? A Critical Analysis with SITRA-B. *Proc. Int. Conf. Road Traffic Signalling*. Institution of Electrical Engineers, London, pp. 89-92.
- Gabard J.F. (1991). Car-Following Models. *Concise Encyclopedia of Traffic & Transportation Systems*, M. Papageorgiou (Ed ;°, Pergamon Press, Oxford, 1991.
- Gallego J.L., Farges J.L. and J.J. Henry (1997). Traffic Queue Estimation. *Sociedad de Estadística e Investigación Operativa, Top*, Vol. 5, No. 1, pp. 81-93.
- Helly W. (1961). Simulation of Bottlenecks in Single-Lane Traffic Flow. In Herman R.C. (Ed.), *Theory of Traffic Flow, Proc. Symp. Theory of Traffic Flow*. Elsevier, Amsterdam, pp. 207-238.
- Henry J.J. and J.L. Farges (1994). PT Priority and PRODYN. *Proceedings of the 1<sup>st</sup> World Congress on Application of Transport Telematics and Intelligent Vehicle-Highway Systems*, Vol. 6, ERTICO (Ed.), pp 3086-3093.
- Henry J.J., Farges J.L. and J.L. Gallego (1997). Neuro-Fuzzy Techniques for Traffic Control. *Proceedings of the 8<sup>th</sup> IFAC / IFIP / IFORS Symposium on Transportation Systems*, Chania, Greece, Vol. 2, pp. 719-724.
- Hugosson B., Lind G. and S. Algers (1997). Evaluating ITS Strategies : User Needs and Survey of Existing Models. *Proceedings of the 4<sup>th</sup> World Congress on Intelligent Transport Systems*, Berlin, 21-24 October 1997.
- Kessaci A., Farges J.L. and J.J. Henry (1991). Turning Movement Ratios Estimation Using Inductive Loop Detectors and Information from Route Guidance Systems. *Recherche - Transports - Sécurité (Revue de l'INRETS)* English issue, n° 6, pp.39-44.
- Pommier L. and M. Barbier (1992) SITRA-B+ User's Manual, V 1.1. *ONERA/CERT report*.

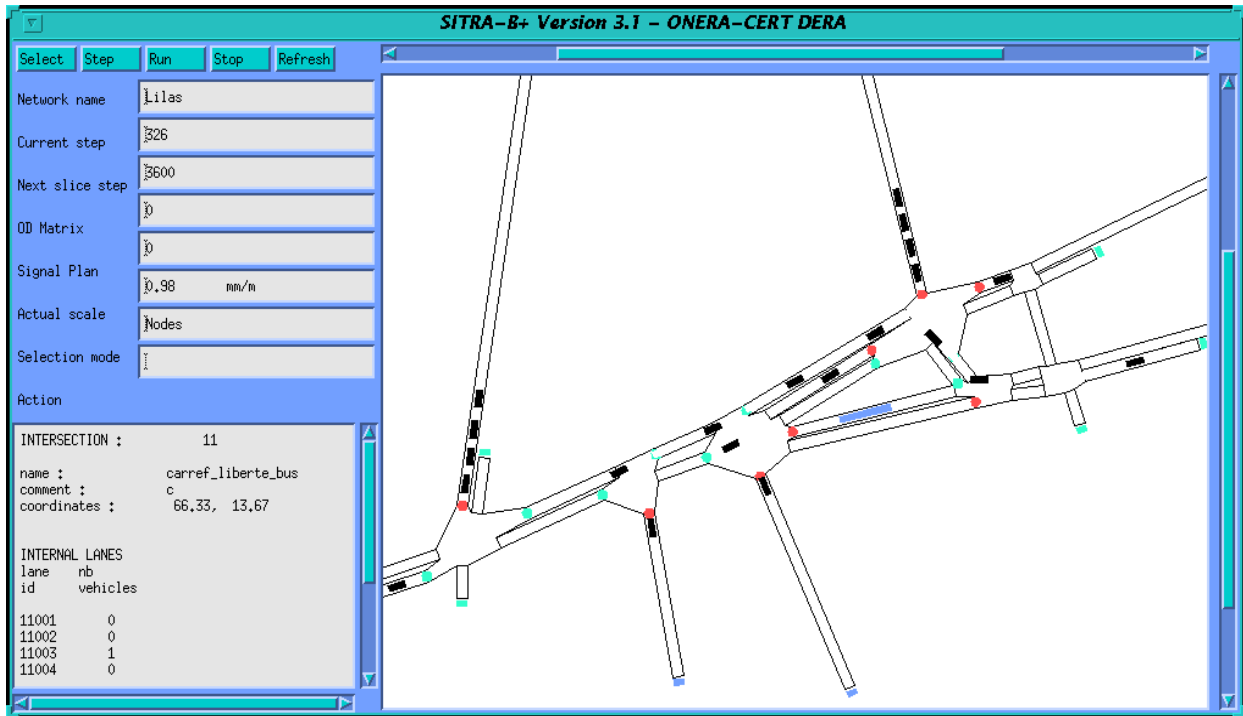


Figure A1 : PRODYB-BUS experiment in Les Lilas

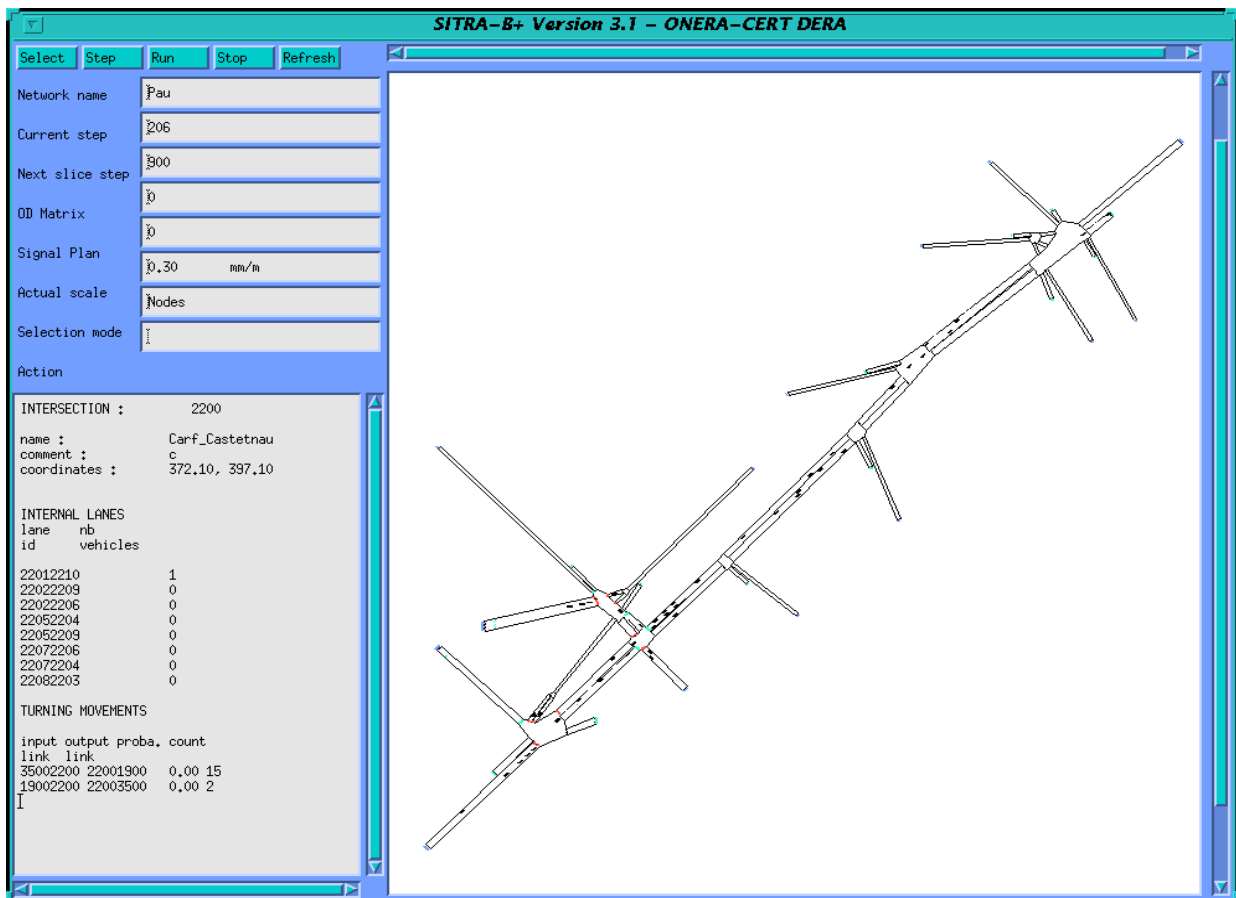


Figure A2 : study of robustness of bus priority algorithms in Pau

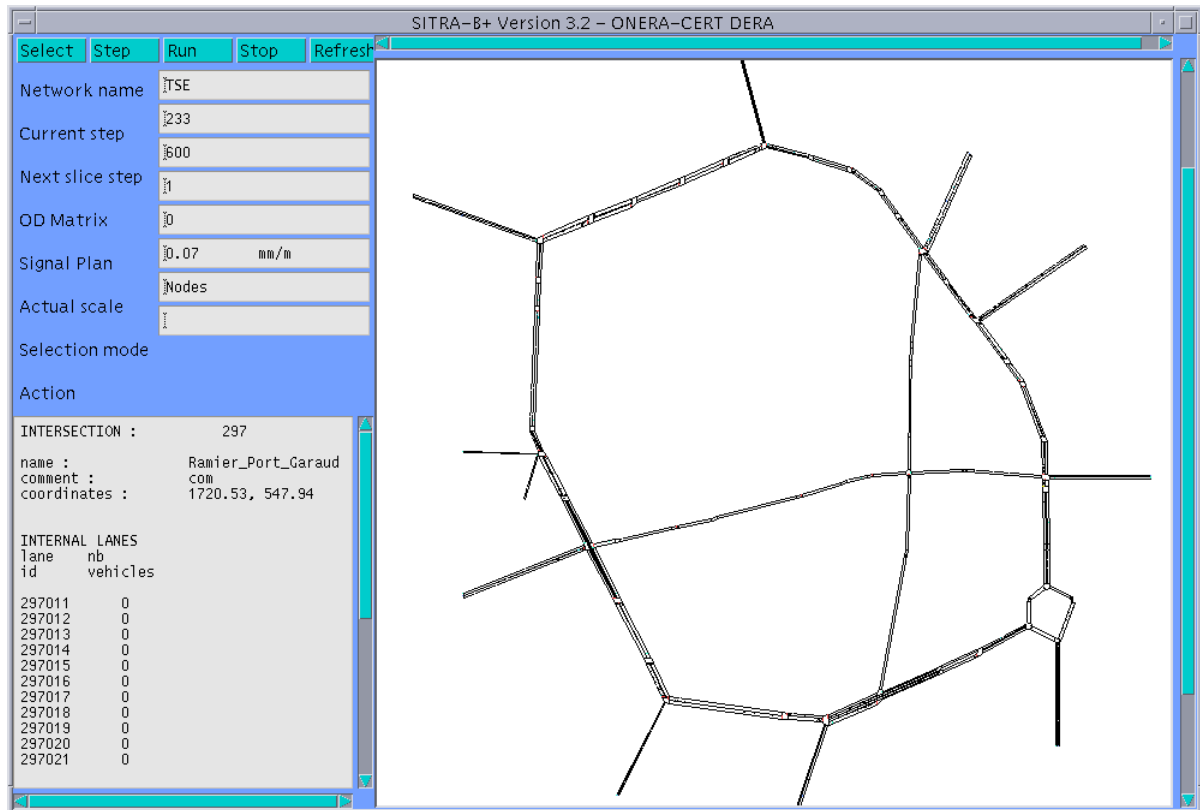


Figure A3 : assessment of guidance strategies in Toulouse test site

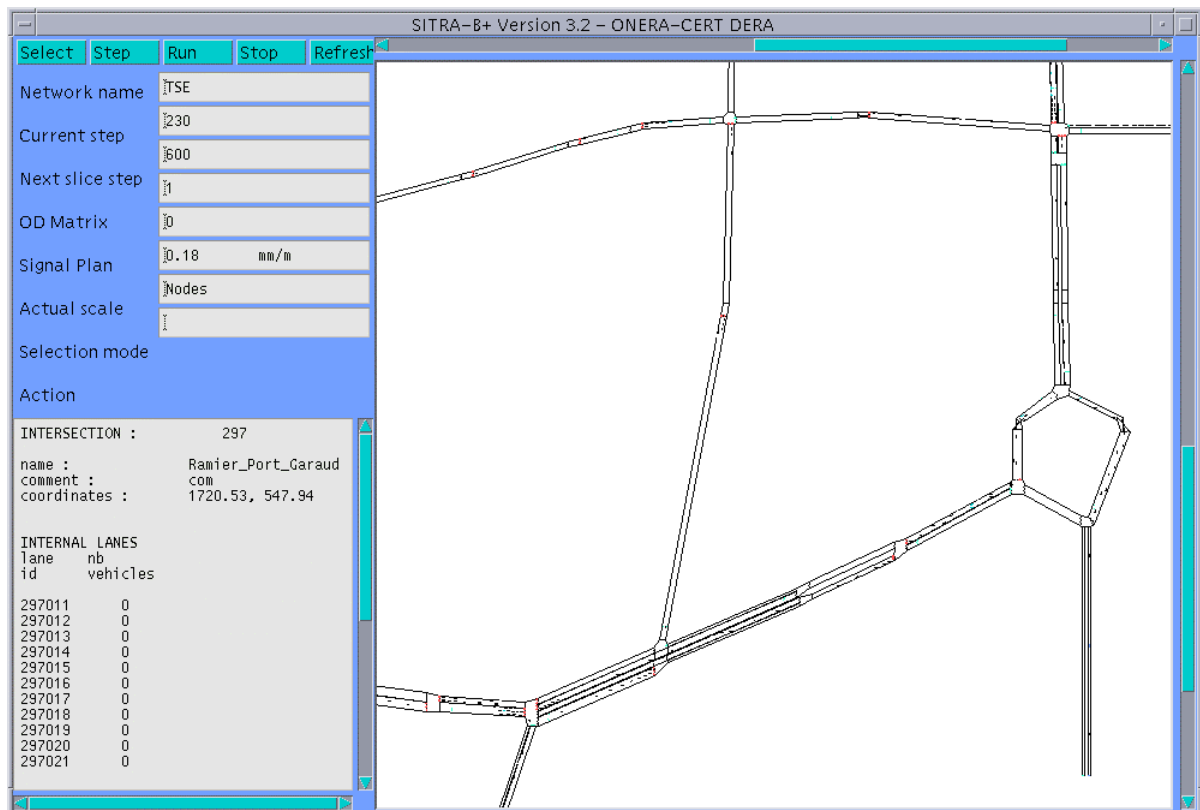


Figure A4 : assessment of guidance strategies in Toulouse test site

## Transferability of urban micro-simulation methods

Ken Fox

Institute for Transport Studies, University of Leeds, Leeds, LS2 9JT, UK

e-mail: [K.A.Fox@leeds.ac.uk](mailto:K.A.Fox@leeds.ac.uk)

### *Abstract*

Transferability is a key issue in the global environment that transport modellers often find themselves working in today. If a model has been developed using data at one location, can this help with the analysis at another site? This paper addresses this transferability problem for urban micro-simulation models. It reports on work carried out in the SMARTTEST project that has been part funded by the European Commission DGVII.

Three micro-simulation tools have been used in the study, namely:

- DRACULA - developed by ITS, Leeds,
- NEMIS - developed by Mizar Automazione, Turin and
- AIMSUN2 - developed by UPC, Barcelona.

All three tools were used to model a road network in Leeds. Comparisons were made between the outputs of each of the tools and data collected from the real road network.

Other issues addressed include:

- the ability of each of the tools to model all the features found in the test network,
- whether default values for calibration parameters such as those for car following are valid at the test site and
- the sensitivity and robustness of the results.

### 1. INTRODUCTION

The Transport Directorate (DGVII) of the European Commission is part funding the SMARTTEST project which is aiming to improve road traffic micro-simulation models. One of the issues being tackled in this project is transferability. In the context of road traffic micro-simulation models, transferability analysis needs to answer the following two key questions:

1. If we have a model developed using data at one site in Europe, can this help with the analysis at another site elsewhere in Europe?
2. How confident can we be that any conclusions we draw at one site, after evaluating a scheme with a micro-simulation model, are valid at another site?

This paper covers work done in the SMARTTEST project to answer these questions. The project has done this by attempting to model a small test network in Leeds with three different micro-simulation packages developed by partners within the SMARTTEST project consortium. The paper starts with a description of each of the micro-simulation models. It then goes on to describe the test network and the data collected from the street for comparison with the model outputs. Next, the different modelling approaches used by each of the micro-simulation models are examined and compared. The assumptions made in order to use each model to simulate the test network are detailed.

## 2. THE MICRO-SIMULATION MODELS

The SMARTEST project consortium consists of eight partners from four European countries. Four different micro-simulation models are being enhanced by the project, three of which have been used in this study. These are AIMSUN2, DRACULA and NEMIS. Each model has been developed in a different European country; AIMSUN2 in Spain, DRACULA in the UK and NEMIS in Italy.

### 2.1 AIMSUN2

AIMSUN2 (Advanced Interactive Microscopic Simulator for Urban and Non-urban Networks) has been developed by the Universitat Politècnica de Catalunya (Ferrer and Barcelo, 1992). It is a software tool capable of modelling real traffic conditions in an urban network which may contain both expressways and arterial routes. AIMSUN2 is a combined discrete-continuous simulator: there are some elements of the transportation system (vehicles, detectors) whose state changes continuously over the simulated time period, while there are other elements (traffic lights, entrance points) whose state changes discretely at specific points during the simulation time. It provides very detailed modelling of the traffic network: it distinguishes between different types of vehicles and drivers; it can deal with a wide range of network geometries; it can also model incidents, conflicting manoeuvres, etc.

AIMSUN2 needs three types of input data: the network description, the traffic signal control plans and the traffic conditions. The network description contains information about the geometry of the network, turning movements, layout of links (or sections) and junctions and location of detectors. The traffic control plans define the signal stages and their durations for signal controlled junctions and the priority definition for unsignalized junctions. Two options are available to define the traffic flows in the network. Either the traffic flows for the input links and the turning proportions at junctions or an O/D matrix can be used.

The outputs provided by AIMSUN2 include a continuously animated graphical representation of the traffic network, a printout of statistical data (flows, speeds, journey times, delays, stops), and data gathered by the simulated detectors (counts, occupancy, speeds, queue lengths).

### 2.2 DRACULA

DRACULA (Dynamic Route Assignment Combining User Learning and Micro-simulation) is a microscopic traffic network modelling suite, conceived and developed at the Institute for Transport Studies, University of Leeds over a five year period (Liu et.al., 1995). The development, testing and validation of the model have been primarily funded by a large grant from the UK Engineering and Physical Sciences Research Council, although some early applications of the model were possible under funding from the EC's DRIVE II Telematics programme. Applications of this model in progress, or to commence shortly, include the study of congestion based road pricing, real time traffic signal control, dynamic route guidance, segregated busway design, emergency evacuation procedures (eg. Following chemical explosions, floods), and strategic (inter-urban) modelling.

### 2.3 NEMIS

NEMIS is a scientific software package developed by Mizar Automazione in Turin, Italy (Mauro, 1991). Since its creation, it has been used principally for research and development work and for the technical assessment of traffic control strategies. It has been developed for the micro-simulation of urban traffic (private and public) in large scale networks. It is capable of modelling urban networks and vehicle behaviour in considerable detail, and is well structured to meet a variety of application needs. Its usefulness has been demonstrated for the following tasks:

- analysis of the effects of regulation and network modification on traffic mobility

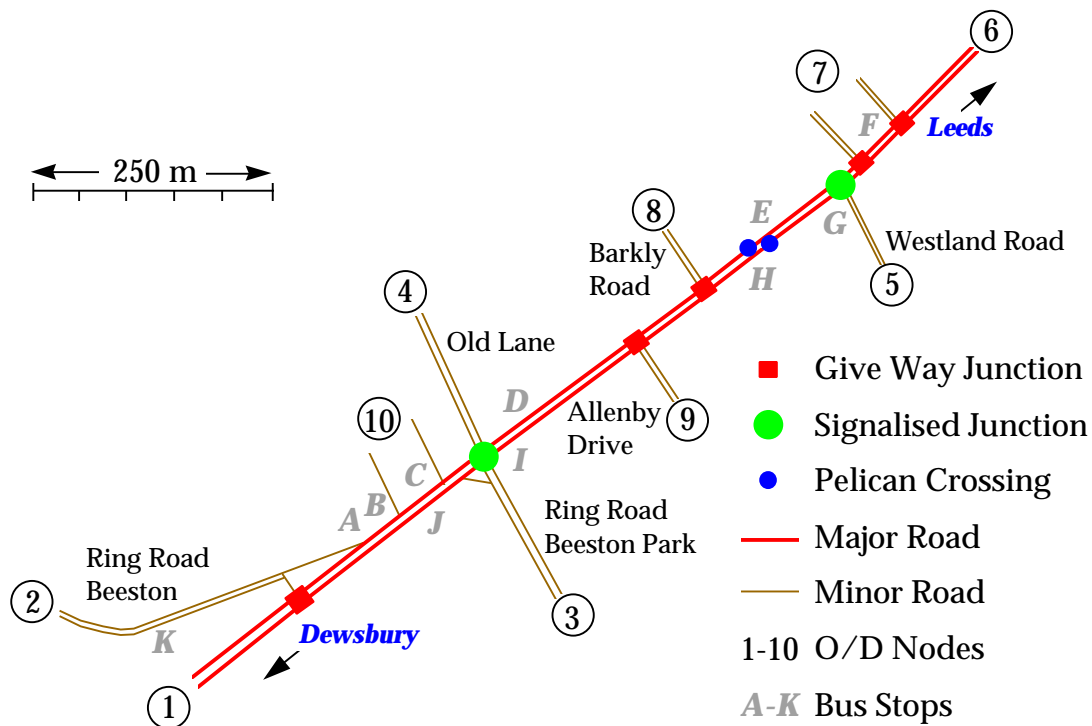


- evaluation of different traffic signal control strategies
- testing of traffic assignment techniques
- simulation and evaluation of route guidance strategies and variable message systems

Simulation occurs in increments of one second. Outputs include pollution emissions and fuel consumed by vehicles in the network.

### 3. THE TEST NETWORK

#### 3.1 Network Description

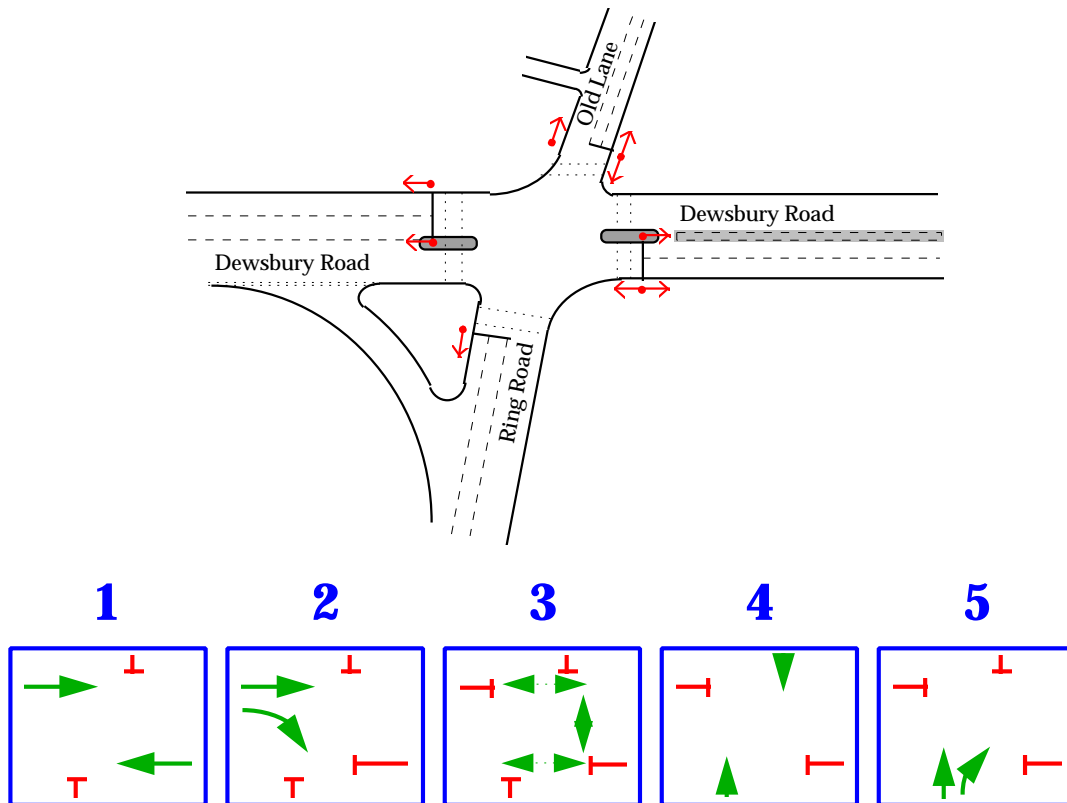


**Figure 19: The Dewsbury Road - Leeds**

A network in Leeds was chosen for this study because of the ready availability of suitable data to both define the network and to compare against model outputs. This data had been collected as part of the PRIMAVERA project (Fox et al., 1995). PRIMAVERA developed advanced traffic management strategies for urban arterial roads. These strategies were developed with the aid of the NEMIS micro-simulation tool and then the best strategy was implemented on-street. Much data was collected, firstly to calibrate and validate the initial micro-simulation model of the network, then to evaluate the effectiveness of the new strategies on-street. Data was collected for both AM and PM Peak periods. The full PRIMAVERA network in Leeds consisted of ten signalised intersections along 3km of an urban arterial, namely the Dewsbury Road, classified as the A653. This is one of the main radial routes into Leeds, carrying approximately 23,000 vehicles per day. It is also a heavily used public transport corridor, peak bus flows being in excess of 36 buses per hour.

To simplify the transferability tests carried out by the SMARTTEST project, a sub-network of the PRIMAVERA network was used. This consisted of a 1½ km segment of the Dewsbury Road, containing two signalised junctions and a pelican crossing (see Figure 19). The test network also contains a number of priority junctions, where minor roads join the main arterial. Bus stops are also present in the network. The network thus contains many features that are common in urban road networks in the UK. An additional feature of the test network is that there is only one route

between each of the origin destination pairs, therefore route choice is not an issue to cloud the model evaluation. It was also decided to only carry out simulation runs of the AM peak period.



**Figure 20: Old Lane / Ring Road Beeston Park Junction**

The main signalised junction in the network is where Old Lane and the Ring Road Beeston Park meet the Dewsbury Road. This is a four arm junction with pedestrian facilities (see Figure 20). All the approaches are signalised, however the left turn from the Ring Road onto the Dewsbury Road is an unsignalised slip road. There are bus routes along each of the four arms of the junction.

The other signalised junction in the network is a three arm junction where Westland Road joins the Dewsbury Road (see Figure 21).

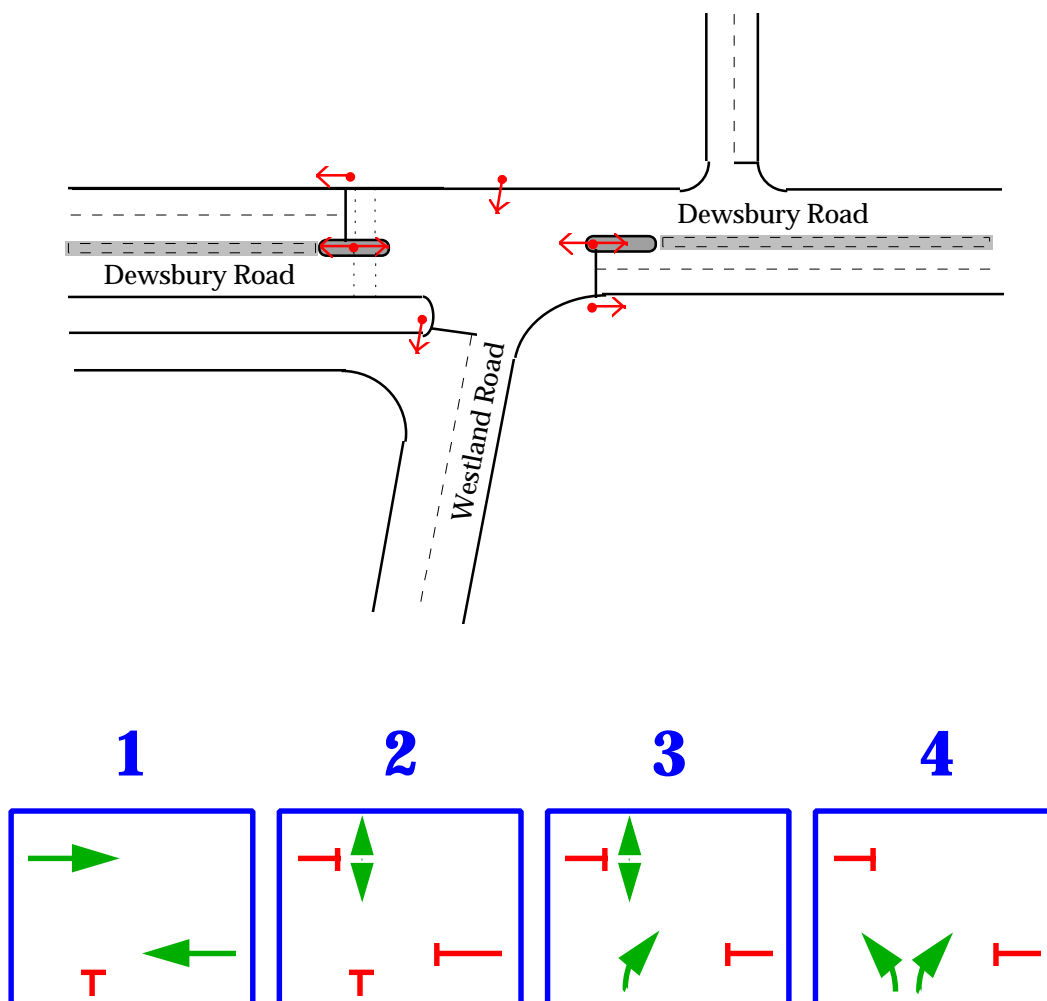


Figure 21: Westland Road Junction

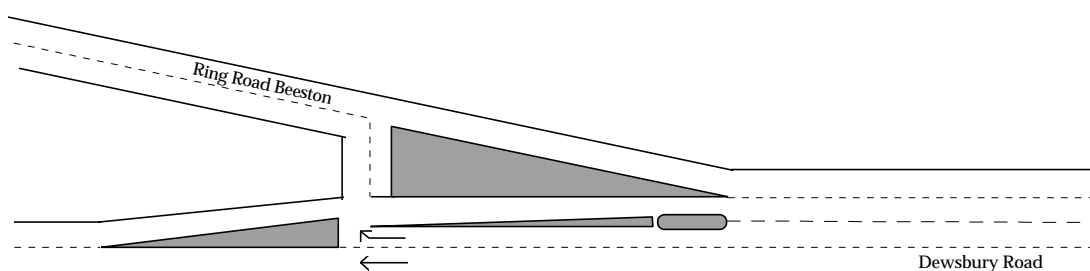
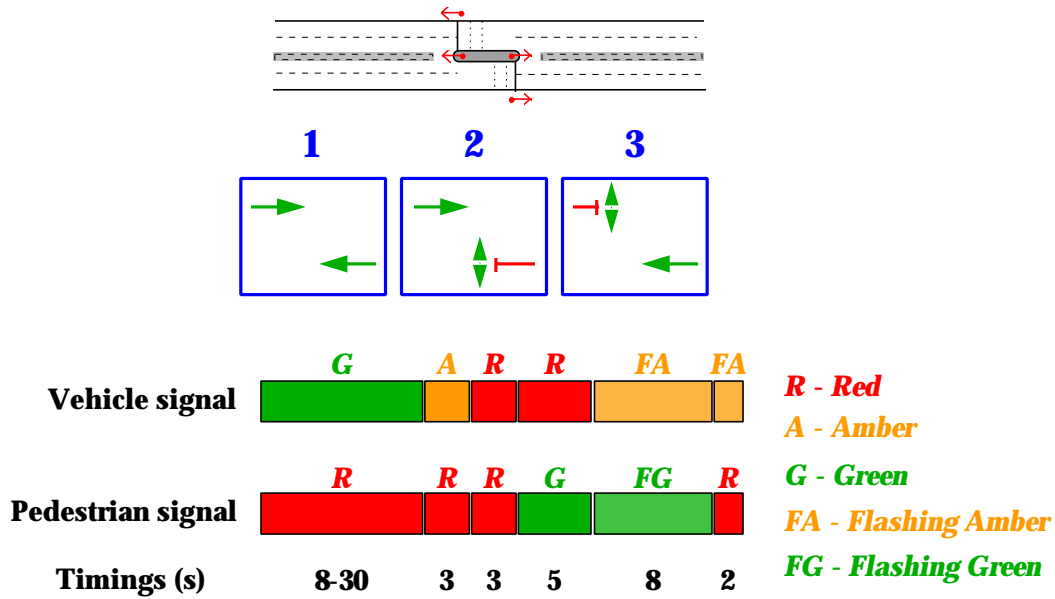


Figure 22: Junction with Ring Road Beeston

The major non-signalised junction in the network is at the south western end where the Ring Road Beeston merges with the Dewsbury Road. This is a quite complicated priority junction. There are separate lanes reserved for right turns from the Dewsbury Road into the Ring Road Beeston and for both left and right turns out of Ring Road Beeston on to the Dewsbury Road (see Figure 22).



**Figure 23: Barkly Road Pelican**

There is also a staggered pelican crossing in the network. It is on the Dewsbury Road close to Barkly Road (see Figure 23).

Although the PRIMAVERA project developed traffic management strategies that utilised the SCOOT and SPOT adaptive signal control systems, at times the network also operated using signals controlled by fixed time plans. The data used for the SMARTTEST transferability tests only related to those periods where the network was under fixed time control. These fixed time plans are given in Table 2. These include the cycle times and offsets and the durations of each phase and intergreen period in seconds. See each of the junction diagrams for details of movements associated with each phase.

Junction	Cycle Time	Offset	1	I	2	I	3	I	4	I	5	I
1	88	34	40	4	4	11	7	8	7	4	4	7
2	88	20	63	6	8	6			10	5		

**Table 2: AM Peak fixed time plans**

Junction 1: Old Lane. Junction 4: Westland Road.

Note that the Barkly Road pelicans are double cycled, i.e. they have a cycle time half of that for the rest of the network.

Bus routes go between many of the origin and destination nodes in the network. The bus routes associated with each origin and destination pair, along with their scheduled entry times at each origin during the AM Peak hour are shown in Table 3.

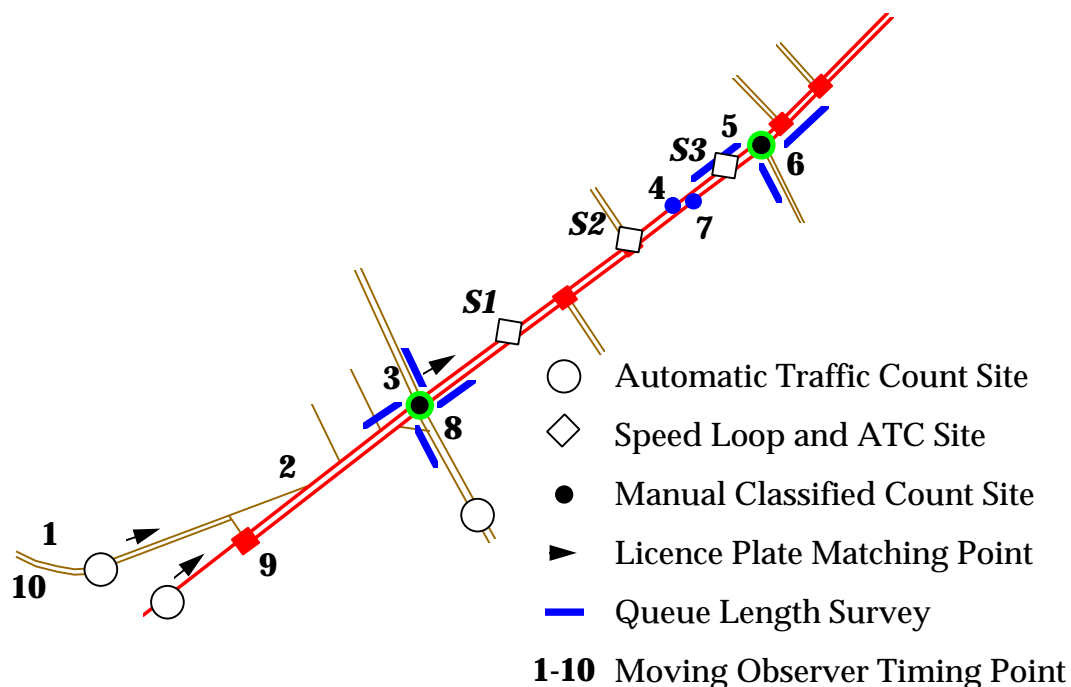
O	D	Buses	Route Number(Origin Start Times)
1	6	15	46(15,45), 117(5), 118(44), 201(10), 202(42), 203(25,55), 218(11), 220(41,51), 222(35), 226(26), X7(5), X16(20)
2	3	2	9(0,36)
2	6	4	2(0,15,30,45)
3	2	2	8(10,34)
3	4	2	74(0,30)
3	6	12	3(9,24,39,54), 24(4,34), 25(17,47), 77(11,41), 484(19,29)
4	3	2	74(19,54)
6	1	12	46(27,57), 117(43), 118(15), 201(26), 202(56), 203(11,41), 218(13), 220(43), 222(3), 226(0)
6	2	4	2(0,15,30,45)
6	3	10	3(8,23,37,52), 24(19,49), 25(4,34), 77(22,52)

**Table 3: Bus Routes, frequencies and starting times during the AM Peak hour**

3.2 Data Collected

3.2.1 Introduction

Much data was collected during the PRIMAVERA project. In addition, a digital map of the area was available in AutoCAD format, which allowed the network geometry to be easily and accurately measured. The surveys carried out are summarised in Figure 24.



**Figure 24: Field Trial Data Collection**

Statistical analysis has been used to estimate the accuracy of the collected data. When comparing simulation outputs with data collected from the real world it is important to ensure that sufficient data has been collected to estimate the values being compared to a desired accuracy. If the usual

statistical assumptions are made regarding the normality of the data then it is possible to determine the confidence interval for a population mean. The confidence interval is a range on either side of the sample mean. It is expressed as a function of a significance level,  $\alpha$ , which usually has a value of 95%, and is given by the formula:

$$(t_L, t_U) = \left( \bar{x} - z_{1-\alpha/2} \frac{s}{\sqrt{n}}, \bar{x} + z_{1-\alpha/2} \frac{s}{\sqrt{n}} \right) \quad (1)$$

where  $z_{1-\alpha/2}$  is that value in the standard distribution that has  $1-\alpha/2$  area to the left. For a 95% confidence level,  $\alpha = 0.05$  and  $z_{1-\alpha/2} = z_{0.975} = 1.96$ .

The surveys on the PRIMAVERA network included:

- Automatic Traffic Counts, using loops, to measure traffic flows in both directions at three points in the network.
- Automatic Speed and Flow measurements as vehicles passed over at three data collection points on the Dewsbury Road.
- Manual Classified Counts at two of the junctions, to obtain turning movements for seven categories of vehicle.
- Travel time surveys both by number plate matching at three points in the network and by moving observers travelling in cars and buses around the network.
- Queue length surveys were carried out at the two signalised intersections.
- Bus waiting times at stops were measured by observers.

### 3.2.2 Flow Data

The Automatic Traffic Counts and the Manual Classified Counts have been combined to produce two Origin/Destination matrices for the AM Peak hour. The first matrix is for cars, the second for heavy goods vehicles (HGVs). These can be seen in Table 4 and Table 5.

	1	2	3	4	5	6	7	8	9	10	Total
1	-	-	90	26	32	470	25	30	30	-	703
2	-	-	50	13	16	240	10	15	15	-	359
3	110	110	-	241	40	230	5	5	-	-	741
4	22	22	215	-	10	50	-	-	5	-	324
5	12	12	5	1	-	80	5	-	-	-	115
6	150	180	100	10	200	-	25	-	30	-	695
7	10	10	5	-	10	25	-	-	-	-	60
8	-	-	-	-	25	150	-	-	-	-	175
9	10	10	5	5	5	25	-	-	-	-	60
10	-	-	-	-	-	50	-	-	-	-	50
Total	314	344	470	296	338	1320	70	50	80	0	

**Table 4: The Car O/D Matrix for the AM Peak hour**

	1	2	3	4	5	6	7	8	9	10	Total
1	-	-	4	1	4	11	-	-	-	-	20
2	-	-	3	-	4	12	-	-	-	-	19
3	8	2	-	-	-	10	-	-	-	-	20
4	-	-	-	-	-	1	-	-	-	-	1
5	5	1	-	-	-	17	-	-	-	-	23
6	8	3	10	4	4	-	-	-	-	-	29
7	-	-	-	-	-	-	-	-	-	-	0
8	-	-	-	-	-	-	-	-	-	-	0
9	-	-	-	-	-	-	-	-	-	-	0
10	-	-	-	-	-	-	-	-	-	-	0
Total	21	6	17	5	12	51	0	0	0	0	

**Table 5: The HGV O/D Matrix for the AM Peak hour**

### 3.2.2 Car Travel Time Data

Two “moving observers” were used to collect travel time data in the network. Staff drove a car round two defined routes and noted times as they passed specific timing points (see Figure 24). Data was collected on five weekdays in July and three weekdays in October 1994. The data relevant for the AM peak is presented in Table 6. This identifies the times between the various timing points, the number of observations (n), the mean travel time (seconds), the standard deviation of the travel time (seconds), the 95% confidence interval as given by equation (1), and the distance between the timing points (metres).

Link	n	Mean (s)	s.d.	C.I.	Length (m)
1-2	50	41.0	9.79	(38.3,43.7)	469
2-3	51	63.0	32.39	(54.1,71.9)	196
3-4	98	36.1	4.37	((35.2,37.0)	493
4-5	100	24.6	11.75	(22.3,26.9)	185
6-7	99	13.7	2.91	(13.1,14.3)	193
7-8	100	56.3	23.53	(51.6,60.9)	480
8-9	98	34.0	15.53	(30.9,37.1)	270
9-10	100	55.6	19.59	(51.7,59.4)	430

**Table 6: Car Travel Times AM Peak (Moving Observers)**

### 3.2.3 Bus Travel Time Data

Moving observers were also used to collect bus travel times in the network. Unfortunately in the sub network used for the transferability study they only covered times between the Westland

Road junction and the Barkly Road pelican. These journey times can be seen in Table 7. It should be noted that there is bus stop between these junctions in both directions.

Link	n	Mean (s)	s.d.	C.I.	Length(m)
6-7	25	41.2	12.8	(36.1,46.2)	193

**Table 7: Bus travel times AM peak**

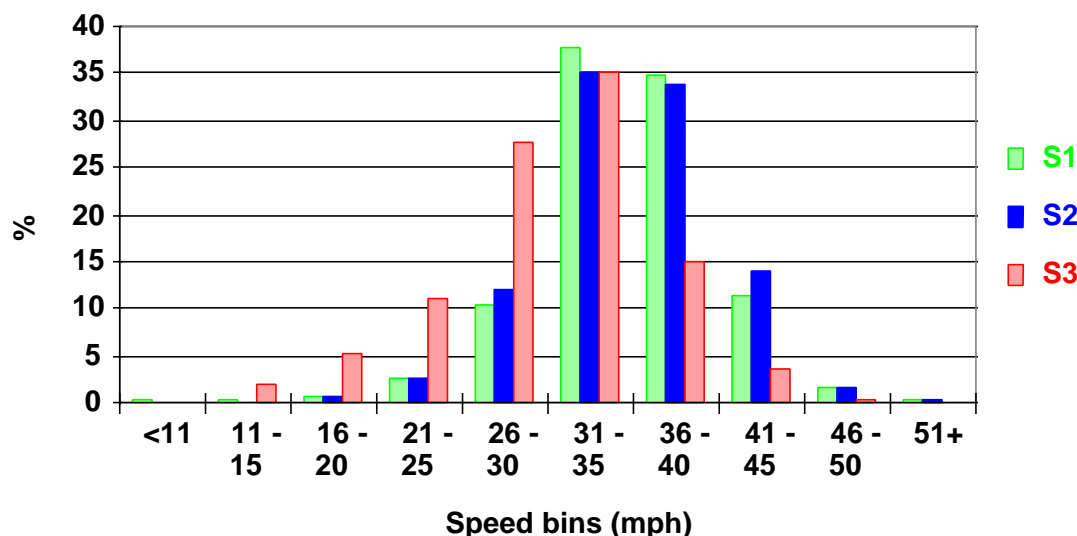
Data was also collected on waiting times at most of the bus stops in the network this can be seen in Table 8 (see Figure 19 for the bus stop identifiers).

3.2.4 Speed Data

	B	C	D	E	F	G	I	J	K
Mean (s)	7.0	7.7	34.1	14.6	29.7	11.0	5.0	18.0	6.0

**Table 8: Mean Waiting Times at Bus Stops**

Speed count data was collected at three points in the network, either side of Barkly Road pelican. These are designated S1, S2 and S3 in Figure 24. The speed count detectors counted the number of vehicles travelling in given speed bins. The speed profiles thus generated during the AM peak are given in Figure 25. The speed limit for this section of the Dewsbury Road is 40 mph. As can be seen, a significant number of vehicles exceed the speed limit here (13% at S1, 16% at S2 and 4% at S3).



**Figure 25: Speed Profiles at Barkly Road**

The mean speeds at each of the three speed count detectors during the morning peak hour is given in Table 9, as is the standard deviation (s.d.), number of observations (n) and the confidence interval (C.I.) given by equation (1).

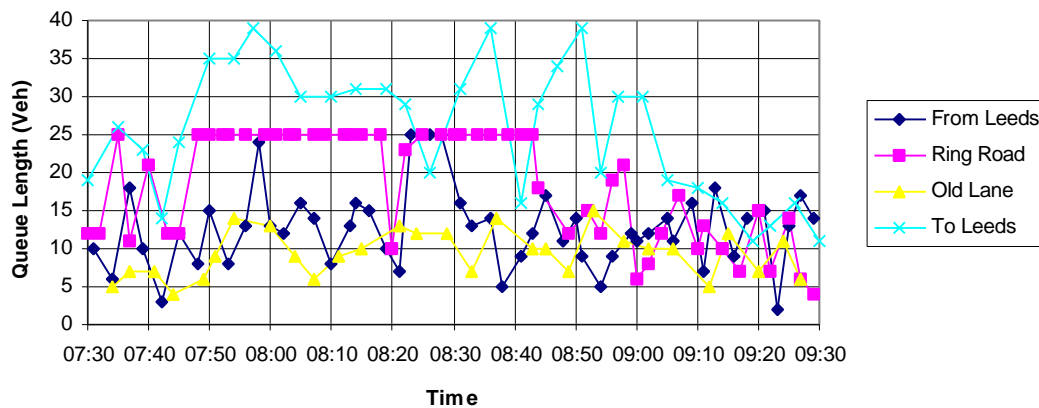
Detector	Mean Speed (mph)	s.d.	n	C.I.
S1	35.1	5.62	2970	(34.9,35.3)
S2	35.4	5.45	3603	(35.2,35.6)
S3	30.5	6.31	1800	(30.2,30.7)

**Table 9: Mean speeds at the detectors (AM Peak)**



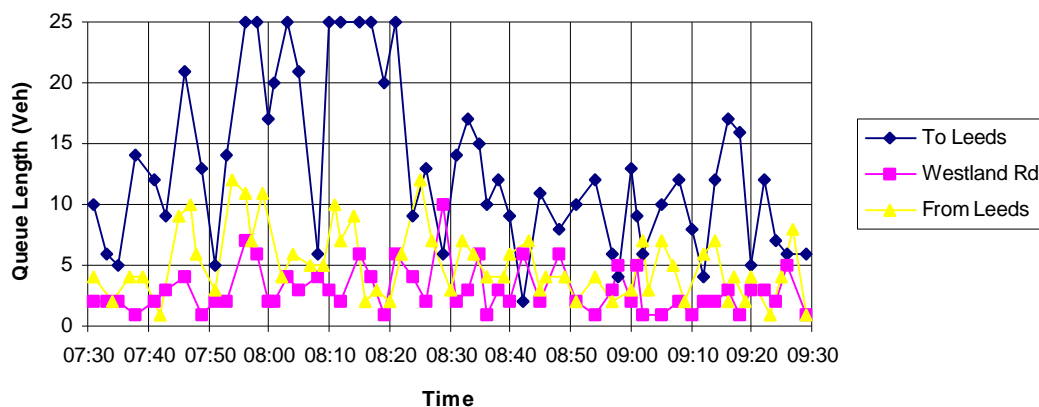
3.2.5 Queue Data

Queue surveys were carried out at the two signalised intersections in the network during the AM peak. Queue lengths were defined by the number of vehicles waiting at the moment the signals changed to green. The queues for the Old Lane / Ring Road Beeston Park junction can be seen in Figure 26. Unfortunately the queue on the Ring Road arm often extended beyond the visibility of the observer.



**Figure 26: Queues on the Old Lane / Ring Road Beeston Park junction (AM Peak)**

Figure 27 shows the queues on the Westland Road junction. Once again there were times when the queues on one of the arms (Dewsbury Road traffic going to Leeds) extended beyond the visibility of the observer.



**Figure 27: Queues on the Westland Road junction (AM Peak)**

The data relevant for the arms where the queue was always fully observed are shown in Table 10. This identifies the various arms, the mean queue (vehicles), the standard deviation of the queue (vehicles), the number of observations (n) and the 95% confidence interval as given by equation (1).

	Mean	s.d.	n	C.I.
<b>Old Lane Junction</b>				
To Leeds	25.6	8.75	31	(22.5,28.7)
From Leeds	12.6	5.17	51	(11.2,14.1)
Old Lane	9.3	3.04	29	(8.2,10.5)
<b>Westland Rd Junction</b>				
From Leeds	5.2	2.87	53	(4.4,6.0)
Westland Rd	3.0	1.92	55	(2.5,3.5)

**Table 10: Mean Queue Lengths**

#### 4. MODELLING APPROACHES

##### 4.1 Network Geometry

All three micro-simulation models used similar network representations. Nodes represent junctions, and nodes are connected by links, each with a number of lanes.

Separate links are required for travel in each direction, i.e. none of the models allowed two-way movement on a link. This limitation can be important as it prevents overtaking via the oncoming lane if there is a suitable gap.

NEMIS has a minor limitation in that it can only model road networks where traffic usually drives on the right, so to model the UK network a mirror image has to be used. NEMIS also has a limit of four arms to a junction.

NEMIS is the only model that has provision for on-street parking.

AIMSUN2 has a very user friendly network builder that allows AutoCAD maps to be used as backgrounds. The road network model is then drawn over the top of this map. This allows an accurate network geometry to be specified without fear of error. AIMSUN2 was therefore the first model used to code up the Dewsbury Road network. The link lengths and their positions obtained from the AIMSUN2 model were then used to code up the NEMIS and DRACULA network models.

##### 4.2 Car-Following and Lane Changing

Driver behaviour is modelled via a car following rule and gap acceptance and overtaking rules. These usually have parameters which characterise desired headways, reaction times, aggressiveness, awareness and acceptable gaps for lane changing and turning across opposing traffic flows. Due to difficulties in measuring these parameters few of them are ever measured directly. The modeller relies on indirect measurements such as average headways, lane usage or saturation flow measurements to justify the values used.

AIMSUN2 uses a car following law based on that suggested by Gipps (1981) and a lane changing rule based on Gipps (1986). NEMIS uses a different car-following law, based on a study by Donati and Largoni (1976). Key parameters for four different vehicle types have been determined.

##### 4.3 Vehicle Types

Both NEMIS and DRACULA have a limit on the number of vehicle types allowed. DRACULA is limited to six types, namely Cars, Buses, Guided Buses, Taxis, High Occupancy Vehicles and

Heavy Goods Vehicles. NEMIS allows seven types, namely five different types of private vehicle, plus buses and trams. AIMSUN2 allows multiple vehicle types to be specified.

Each vehicle type has associated with it a fixed set of parameters, such as acceleration and deceleration rates, vehicle length and car following parameters. Table 11 gives some of the default parameters provided for the various vehicle types used by each of the models.

<b>DRACULA</b>	Car	Bus
Maximum Acceleration (m/s/s)	2.5	2.5
Maximum Deceleration (m/s/s)	2.5	2.5
Length (m)	3.5	7.5

<b>AIMSUN2</b>	Car	Truck	Bus	Long Truck
Maximum Acceleration (m/s/s)	2.8	1.0	2.0	1.0
Maximum Deceleration (m/s/s)	4.0	3.5	3.0	3.5
Length (m)	4.0	8.0	9.0	12.0
Desired Speed (km/h)	90	70	60	70

<b>NEMIS</b>	All vehicles
Maximum Acceleration (m/s/s)	3.0
Maximum Deceleration (m/s/s)	5.0
Maximum Speed (km/h)	50.4

**Table 11: Some default micro-simulation motion parameters**

#### 4.4 Public Transport

The main drawback of AIMSUN2 is that it does not currently directly model public transport. Although it is possible to model a bus vehicle type, it is not possible to specify routes, timetables or bus stops. This can be very important in urban networks where it is often difficult for other traffic to overtake buses at stops. Buses can therefore have a significant effect on traffic flow in the network.

DRACULA and NEMIS allow both bus routes and bus stops to be specified. Both specify the routes by defining a list of links to be followed. Both use a start time and a generation frequency to produce the bus schedules.

For DRACULA bus stops are associated with bus services. For NEMIS the stops on a route can be used by any of services that use the route. Both allow multiple stops on a link. DRACULA uses a simple wait time model for the bus stops based on a passenger arrival rate, although this is not service dependent. NEMIS just has a stop time based on a sample from a normal distribution of a fixed mean and standard deviation.

#### 4.5 Traffic Flows

All three models have the ability to accept traffic flow data in the form of Origin / Destination (O/D) matrices. AIMSUN2 and NEMIS have built in route choice models. DRACULA uses the SATURN assignment model (Van Vliet, 1982) to calculate vehicle routes.

The vehicle generation models in DRACULA and AIMSUN2 assign an origin, destination and route to each vehicle as they are generated. NEMIS uses the results of its assignment model to produce turning percentages at each junction. So when a vehicle arrives at a junction, a random choice is made, based on the known turning proportions, to choose the direction the vehicle is to make.

AIMSUN2 is the only model that allows different O/D matrices for different vehicle types. This could be an important factor in the Leeds network, where HGVs have a slightly different O/D pattern to other vehicle types.

#### 4.4 Traffic Signals

All the models have the capability of modelling traffic signals operating under fixed time control.

The pelican crossing can be modelled as a two arm junction with 1 stage and a long intergreen. None of the models directly allow the demand response feature of the pelican crossing to be modelled (or other demand responsive features that may be present at other signalised junctions in the network). Pelican crossings only show red to the traffic if a pedestrian has pressed a button to register their desire to cross the road. AIMSUN2 does have the ability to allow an external module to be developed to control signals in the network so it would be possible to write such a module (as a dynamic link library) to model the correct actions of a pelican crossing. Time constraints have however not allowed such a development. It has therefore been decided that as during the AM peak it is likely that the pelican crossing will be used nearly every cycle, it can be modelled as if it was used every cycle.

### 5. SIMULATION RESULTS

The Leeds network has been coded up for each of the three micro-simulation models. Calibration and validation has been carried out.

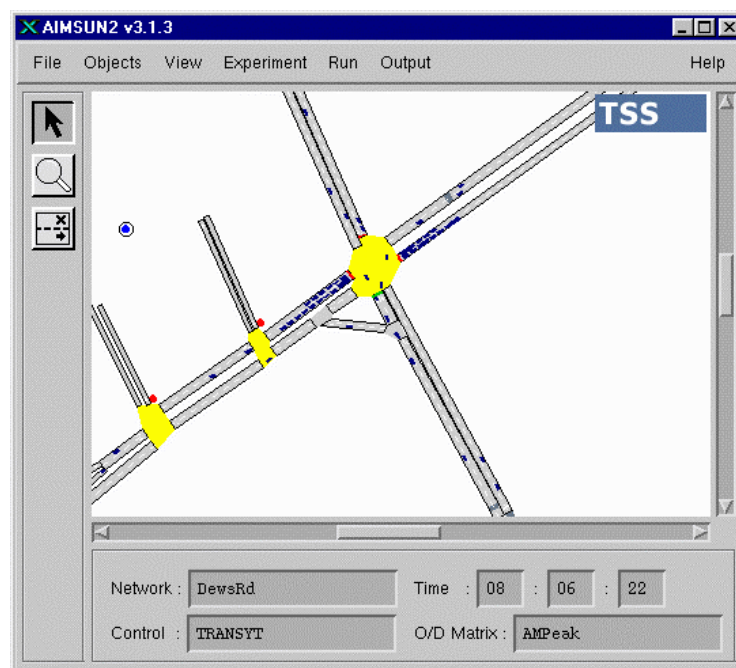
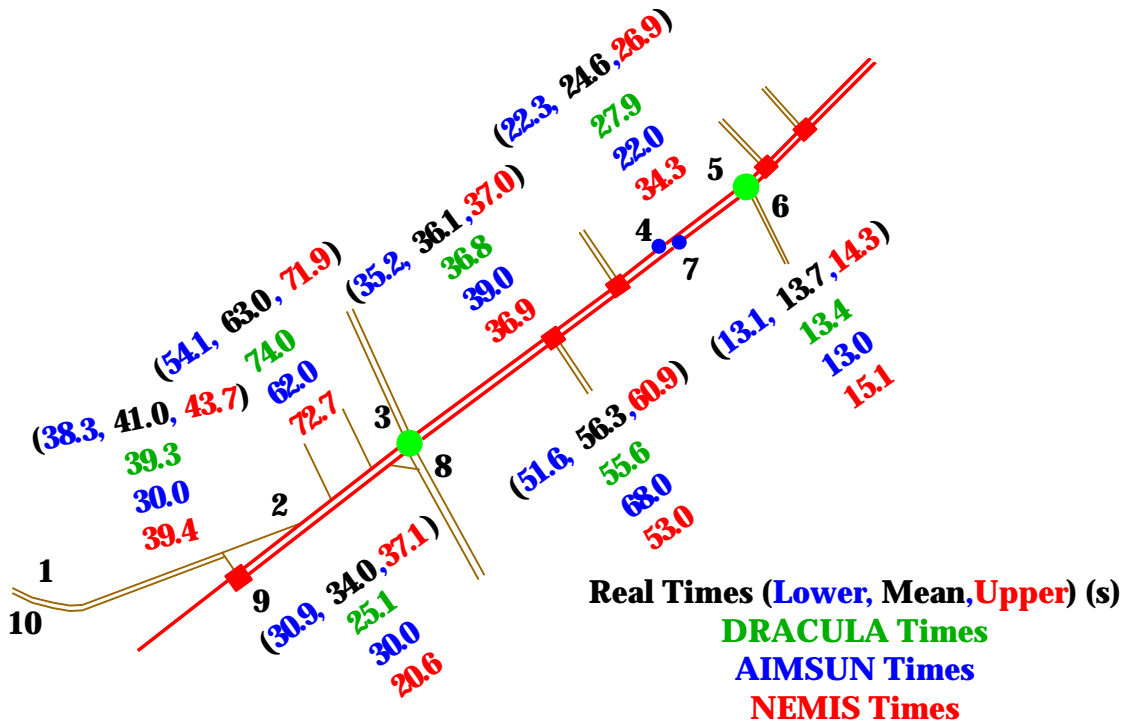


Figure 28: AIMSUN2 simulating the Leeds network

Averages from four different runs using different random number seeds for each run were used.



**Figure 29: Link travel times from the different models**

Figure 29 shows the link travel times from each of the micro-simulation models. These are compared with the actual travel times, as measured by moving observers and detailed in section 3.2.2. For each link, only the travel times for vehicles making the same turning movement at the end of the link as the moving observers were used in the analysis. As can be seen there is reasonably good agreement between the observed travel times and those output by the models.

Figure 30 shows a comparison of the queue lengths from each of the models. Here the agreement between reality and the model outputs is not so good. In particular the queue lengths from DRACULA and NEMIS are much longer than those observed for the Dewsbury Road link going into the Old Lane / Ring Road Beeston Park junction towards Leeds. This is a critical junction. The result indicates that both NEMIS and DRACULA have problems modelling junctions operating close to capacity.

Table 12 shows the times (in seconds) for each of the simulation models to model one hour in the AM Peak period. All the runs were performed on the same computer, which was a 200 MHz Pentium PC with 32Mb of memory. Runs have been carried out both with the graphics switched on and with them switched off. As can be seen, having animated outputs significantly slows down the simulation for all of the models. With the graphics switched off, both NEMIS and DRACULA are slightly faster than AIMSUN2. This can be partly explained by the fact that the AIMSUN2 runs were performed using the default step length of 0.75 seconds, whereas the DRACULA and NEMIS runs used a 1 second timestep.

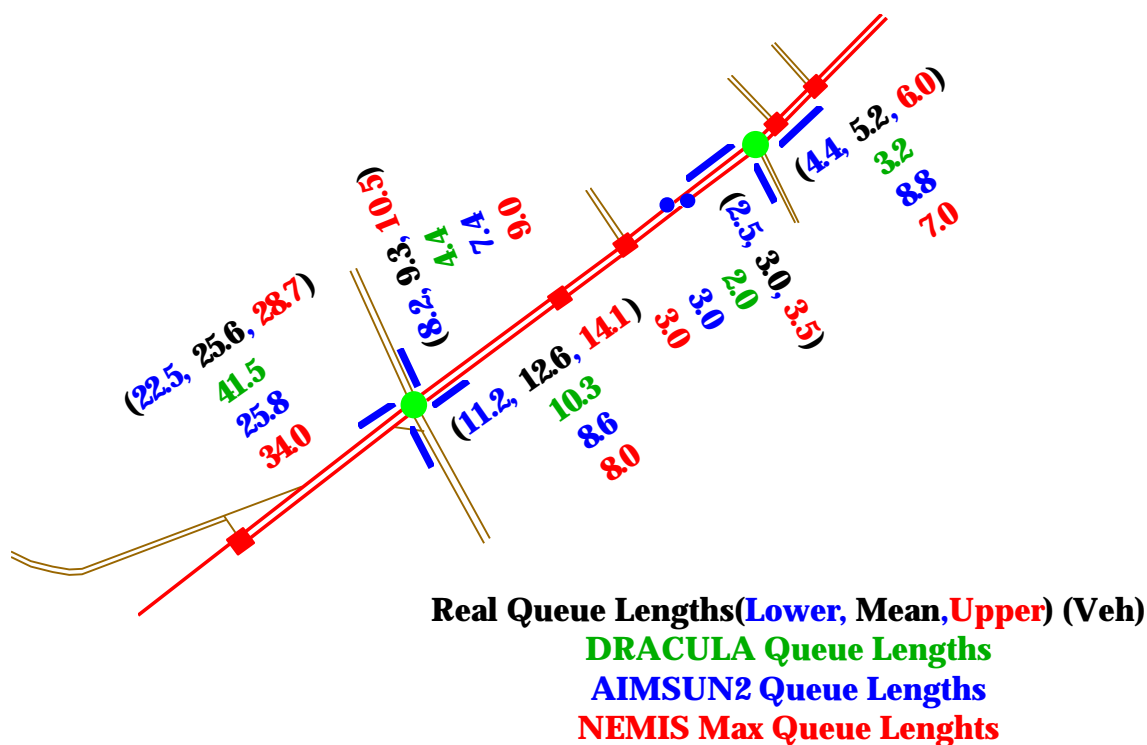


Figure 30: Queue lengths from each of the models

	Graphics on	Graphics off
DRACULA	68	24
AIMSUN2	375	39
NEMIS	64	24

Table 12: Simulation run-times (s) for one hour in the AM peak.

## 6. CONCLUSIONS

A variety of traffic data has been collected from a small urban road network in Leeds. This data has been processed and analysed so that it can be used in the calibration and validation of road traffic micro-simulation models.

Three micro-simulation models, initially developed to model traffic in different parts of Europe, have been used to model the traffic in the Leeds network. None of the models could represent all the features found in the test network, so some modelling assumptions had to be made to cover these cases.

The ability of the models to produce accurate representations of traffic behaviour has been investigated. All three models produce reasonably accurate outputs of travel times. Both NEMIS and DRACULA however have problems in modelling junction capacities accurately, which results in over estimations of queue lengths and travel times at junctions operating close to capacity.

## 7. REFERENCES

DONATI, F. AND LARGONI (1976) Analisi del comportamento di una colonna di autoveicoli in condizioni perturbate, *Riunione Annuale AEL*, Sorrento, 1976.

FERRER, J.L. AND BARCELO, J. (1992) *Modelos microscopicos de simulacion de sistemas de guiado de veh'culos*. Research Report. Laboratori d'Investigació Operativa i Simulació, Universitat Politècnica de Catalunya.

FOX, K., MONTGOMERY, F.O. AND MAY, A.D. (1995a) Integrated strategies for urban arterials: DRIVE II project PRIMAVERA. 1: Overview *Traffic Engineering and Control*, **36**(5), pp268-271.

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

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

LIU, R., VAN VLIET, D. AND WATLING, D.P. (1995) DRACULA - Microscopic Day-to-Day Dynamic Modelling of Traffic Assignment and Simulation, Paper presented at the *Fourth International Conference on Applications of Advanced Technologies in Transportation Engineering*, Capri, Italy, 27-30 June 1995.

MAURO, V. (1991) Evaluation of dynamic network control: simulation results using NEMIS urban micro-simulator *Transportation Research Board Annual Meeting*, Washington DC.

VAN VLIET, D. (1982) "SATURN - A Modern Assignment Model", *Traffic Engineering and Control*, **23**, pp578-581.