AIMSUN: New ITS Capabilities

J. Barceló, P. Barceló, J. Casas and J.L. Ferrer
TSS-Transport Simulation Systems
info@tss-bcn.com

Summary

This paper summarises the main modelling and interface developments made recently in the AIMSUN microscopic traffic simulator to provide a better response to the requirements for the assessment of ITS systems and advanced transport analysis. The description addresses three main areas: improvements on the dynamic assignment capabilities, in interfacing traffic control systems, as those based on TRANSYT, and in the interface between EMME/2 and AIMSUN. In this last case a special attention is paid to the manipulation of OD matrices to define inputs to AIMSUN models.

1. Introduction

Microscopic traffic simulators are very likely the most powerful and versatile traffic analysis tools. Its ability to reproduce to a significant level of accuracy the observed traffic conditions in a broad variety of circumstances makes that the skilled users become very demanding, asking for new features and functionalities in the never ending process of fitting better the increasing complexity of traffic phenomena. Demand that namely reaches its highest point when ITS applications are involved. Consequently AIMSUN is a continuously evolving traffic simulator, as well as GETRAM, its supporting software modelling platform. Among the many features and functions added last year to GETRAM/AIMSUN (12) (the detailed information can be reached visiting the web page http://www.tss-bcn.com) we would like to highlight three in this paper:

1. Improvements on the dynamic assignment abilities
2. The interface with TRANSYT
3. Improvements on the interface with EMME/2

2. Heuristic Dynamic Assignment

The advent of the Intelligent Transportation Systems made possible by the new technologies, and the transportation R&D Programmes in the European Union (DRIVE I and II, ATT Programme and 5th Framework Programme), USA (IVHS) and Japan (AMTICS and VICS), seeking innovative solutions based on the applications of new computer and telecommunication technologies, has raised new requirements, and set new challenges, in the domain of traffic modeling and simulation. Advanced driver information systems, adaptive traffic control systems, real-time traffic management systems, and so on, are among the applications of these new technologies currently under development or, in some cases, already being tested in field trials. Some common requirements of these applications are: Practical methods of measuring the degree of change in activity flows resulting from system modifications, real-time identification of imbalance situations in the use of available capacity of the road network, definition and assessment of suitable strategies, and implementation of real-time management decisions and control measures

The development of systems successfully fulfilling these conditions requires traffic models that efficiently represent interactions in the transport system dynamically. Therefore, they should take proper account of the effects of time-varying demand, time dependent queuing and so on. These requirements have prompted great interest and effort in recent years in research on dynamic traffic assignment models, both stochastic and deterministic, as well as workable solution algorithms. So far, in spite of the research efforts, no analytical algorithms exist whose computational performance enables their use for practical applications. The currently available algorithms have a set of drawbacks making them impractical for real life applications.

Microscopic traffic simulation has proven its usefulness in many different areas of application dealing with complex traffic systems (2), (3), (4), (5) and (6). Simulation has consequently been proposed as numerical procedure, heuristic in nature, able of providing approximate solutions to dynamic traffic systems, not only due to its ability to capture the full dynamics of time dependent traffic phenomena, but also for being capable of dealing with behavioral models accounting for drivers' reactions when exposed to Intelligent Transport Systems (ITS). To achieve these objectives a microscopic simulator should be able of:

a. Updating timely the routes from origins to destinations depending on changing conditions of traffic over time
b. Assigning the vehicles to routes from origins to destinations at each time period
c. Dynamically re-route vehicles en-route when better alternative routes from their current position to their destination exist.

This type of simulation assumes that the demand is defined in terms of Origin-Destination matrices whose entries represent the number of trips from an origin to a destination as a function of time (1). The routes are calculated according to specified travel costs and the assignment to the routes is based on modeling driver's decisions by means of route choice models. The heuristic dynamic assignment procedure works as follows:

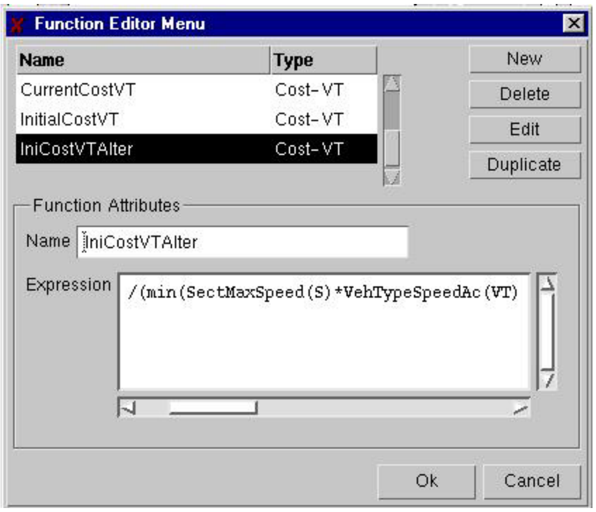
1. Calculate initial shortest routes, taking the estimated initial costs.
2. Simulate for a period using available routes information and obtain new costs as a result of the simulation.
3. Recalculate shortest routes, taking into account the new costs.
4. Add the new information calculated in 3 to the knowledge of the drivers.
5. Go to step 2.

At the beginning of the simulation, shortest path trees are calculated from every section to each destination centroid, taking as arc costs the specified initial costs. During simulation, new routes are recalculated in every time interval, taking the specified arc costs updated for each arc after the statistics gathered during the last interval. The experience and computational tests have shown the importance of accounting for the users perception in defining the available routes, to accomplish this objective the new version of AIMSUN offers three possibilities: (a) Historical Fixed Routes (HFR): Predefined fixed routes, manually by means of the network editor or imported from the output of other traffic simulators (macroscopic, i.e. EMME/2, or microscopic); (b) Historical Shortest Path Trees (HSPT): Predefined shortest paths, which can be imported from the output of previous simulations with AIMSUN or another traffic simulators, and (c) Calculated Shortest Path Tree (CSPT): Shortest path tree calculated using the cost functions (There are two types of CSPT: Initial Shortest Path Tree (ISPT): for each destination centroid, it gives a shortest path tree, using the initial cost function for each turning movement, and Computed Shortest Path Tree (SSPT): shortest path trees computed at each time period for each destination centroid, using cost functions that depend on the statistical data gathered during the simulation.

A vehicle with vehicle type v_t travelling from origin O_i to destination D_j, could choose the route among the following possible paths: The N predefined Historical Fixed Routes: $HFR_k(O_i,D_j)$ $k=1..N$, the M predefined Historical Shortest Path Trees: $HSPT_k(D_j)$ $k=1..M$, the 1 Initial Shortest Path Trees at the beginning of the simulation: $ISPT(D_j)$, and the P Computed Shortest Path Trees: $SSPT_k\ (D_j)$ $k=1..P$

The user may define the time interval for recalculation of the computed shortest paths and the maximum number of path trees to be maintained during the simulation. When the maximum number of path trees (K) is reached, the oldest paths will be removed as soon as no vehicle is following them. It is assumed that vehicles only choose between the most recent K path trees. Therefore, the oldest ones will become obsolete and disused. From the point of view of the practitioner the answers provided to two main questions could heavily condition the use of the heuristic dynamic assignment as analysis tool: the concept of cost used in updating the routes, and the route choice model used in assigning vehicles to available routes

Assuming that route cost is the sum of the costs of the arcs composing the route, a wide variety of arc costs can be proposed: travel times at each simulation interval, toll pricing, historical travel times representing driver's experience from previous days, combinations of various arc attributes as for instance travel times, length and capacity, etc.The improved version of AIMSUN provides the user with two alternatives: use default arc costs or use the Function Editor included in TEDI (Network Editors in the GETRAM modeling environment) to define his/her own arc cost function using as arguments any of the numerical attributes, statistical values or vehicle characteristics. Calculation of shortest paths is carried out per vehicle type, taking into account reserved lanes. Therefore, the set of paths from which a vehicle may select, either when entering the network or when being re-routed, may be different for different vehicle types even though they travel to the same destination, depending on the presence of reserved lanes. Also the travel time used in the cost function for recalculation of shortest paths is taken as the travel time per vehicle type.



Name	Type
CurrentCostVT	Cost-VT
InitialCostVT	Cost-VT
IniCostVTAlter	Cost-VT

Name: IniCostVTAlter

Expression:

```
/(min(SectMaxSpeed(S)*VehTypeSpeedAc(VT)
```

Figure 1. Example of User defined arc cost function per vehicle type

As an alternative to the Default Initial and Cost Functions users can define their own Cost Functions. This is done via the Function Editor in TEDI. If no User-Defined Function is assigned to an arc, the Default Cost Function is applied. To define an Initial or Cost Function, the user can use any of the most common mathematical functions and operators, and as arguments any of the numerical attributes available in the simulator. Figure 1, illustrates an example of a simulation model for which alternative arc cost functions have been defined by the user. The open window shows part of the algebraic expression for a cost function per vehicle type.

In a similar way when using these dynamic assignment abilities it could be raised the question of which is the most suitable route choice function. Route choice functions represent implicitly a model of user behaviour, representing the most likely criteria employed by the user to decide between alternative routes: perceived travel times, route length, expected traffic conditions along the route, etc. The solution implemented in the most recent version of AIMSUN also provides the user two alternatives: use the default functions or define his/her own route choice function by means of the Function Editor. The most used route choice functions in transportation analysis are those based on the discrete choice theory, i.e. Logit functions assigning a probability to each alternative route between each origin-destination pair depending on the difference of the perceived utilities. A drawback reported in using the Logit function is the exhibited tendency towards route oscillations in the routes used, with the corresponding instability creating a kind of flip-flop process. According to our experience there are two main reasons for this behavior. The properties of the Logit function and the inability of the Logit function to distinguish between two alternative routes when there is a high degree of overlapping. The instability of the routes used can be substantially improved when the network topology allows for alternative routes with little or no overlapping at all, playing with the shape factor of the Logit function and re-computing the routes very frequently. However, in large networks where many alternative routes between origin and destinations exist and some of them exhibit a certain degree of overlapping (see Figure 2), the use of the Logit function may still exhibit some weaknesses (7), (8).

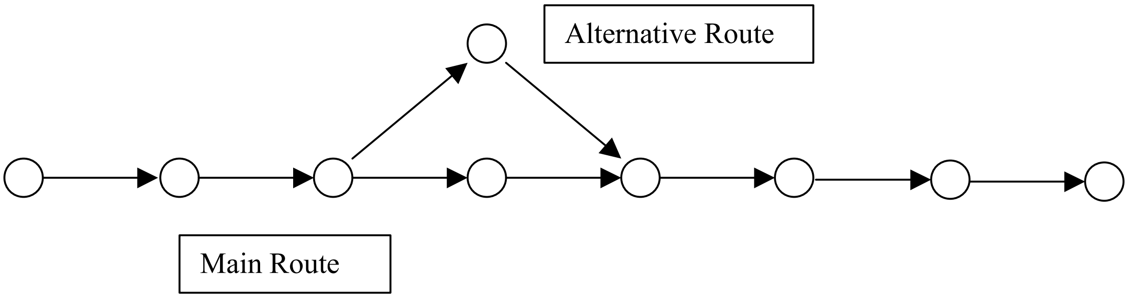


Figure 2: Overlapping Routes

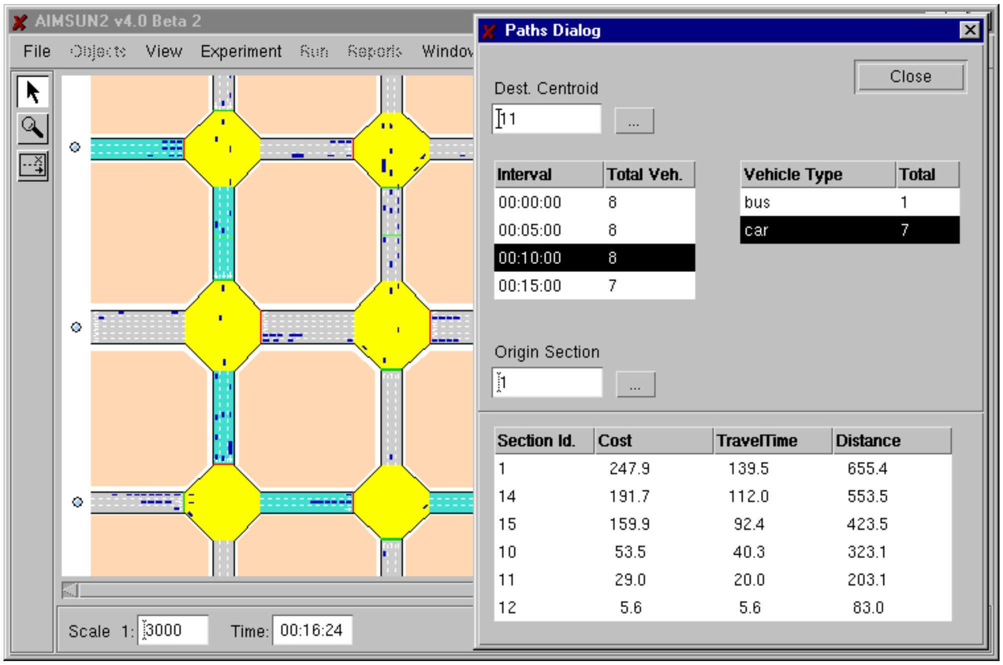
To avoid this drawback the C-Logit model (a variation of the Logit model introduced by Cascetta) has been implemented. In this model, the choice probability P_k, of each alternative path k belonging to the set I_{rs} of available paths connecting an O/D pair, is expressed as: $P_k = \frac{e^{\theta(V_k - CF_k)}}{\sum_{l \in I_{rs}} e^{\theta(V_l - CF_l)}}$ where V_i is the perceived utility for alternative path i , and θ is the scale factor, as in the case of the Logit model. The term CF_k, denoted as ‘commonality factor’ of path k, is directly proportional to the degree of overlapping of path k with other alternative paths. Thus, highly overlapped paths have a larger CF factor and therefore smaller utility with respect to similar paths. CF_k is calculated as follows: $CF_k = \beta \cdot \ln \sum_{l \in I_{rs}} \left(\frac{L_{lk}}{L_l^{1/2} L_k^{1/2}} \right)^{\gamma}$ where L_{lk} is the length of arcs common to paths l and k, while L_l and L_k are the length of paths l and k respectively. Depending on the two factor parameters β and γ, a greater or lesser weighting is given to the ‘commonality factor’. Larger values of β means that the overlapping factor has greater importance with respect to the utility Vi; γ is a positive parameter, whose influence is smaller than β and which has the opposite effect.

To get the insight on what is happening in a heuristic dynamic assignment for the proper calibration and validation of the simulation model the user should have access to the analysis of the used routes. To support the user in this analysis process AIMSUN includes a path analysis tool. Figure 3 depicts the path dialogue window. The path list box contains the list of section identifiers composing the path and the following information is displayed for each section:

- The cost in time (seconds) from each of the sections in the path to the destination centroid. This can be calculated as either the sum of *IniCost(a)* or *Cost(a, vt*) of all the arcs composing the path.
- The travel time in seconds from each of the sections in the path to the destination centroid. This is equal to the cost only if the capacity weight parameter is set to zero.
- The distance (metres) from each of the sections in the path to the destination centroid.

In the example shown in Figure 3, the shortest path from section 1 to centroid 11 goes through sections 14, 15, 10, 11 and 12. The cost of the whole path is 247.9 units (depending on the definition of cost), the travel time is 139.5 seconds and the distance is 655.4 meters. In this case Cost and Travel Time are different, as the Capacity Weight has been set to 1.25.

Given an OD pair (r, s) with origin O_r and destination D_s, if P_{rs} is the set of feasible paths from O_r to D_s and I_T is the set of time intervals to account for, the path analysis tool makes available to the user information on: the current travel time on path k at time interval t, where $k \in P_{rs}$ and $t \in I_T$; the historical travel time on path k at time interval t, where $k \in P_{rs}$ and $t \in I_T$; the current flow on the arcs along the path at time interval t, where $k \in P_{rs}$ and $t \in I_T$, the saturation index on path k at time interval t. This information allows to have: Time Plots of path travel times, time plots of path saturation indexes, and makes also possible the calculation of utilities associated to all paths for the analysis of day-to-day and within-day traffic variations



Dest. Centroid: 11

Interval	Total Veh.
00:00:00	8
00:05:00	8
00:10:00	8
00:15:00	7

Vehicle Type	Total
bus	1
car	7

Origin Section: 1

Section Id.	Cost	TravelTime	Distance
1	247.9	139.5	655.4
14	191.7	112.0	553.5
15	159.9	92.4	423.5
10	53.5	40.3	323.1
11	29.0	20.0	203.1
12	5.6	5.6	83.0

Scale 1: 3000

Time: 00:16:24

Figure 3: Path Dialog Window

3. Interface with TRANSYT

i) The main objectives of this interface are the following: Convert a GETRAM network into a TRANSYT/10 one, use a TRANSYT/10 control plan into AIMSUN simulator, and display TRANSYT/10 units and measures of performance produced by AIMSUN. The figure 4 shows the structure of the interface. According to this structure, the GETRAM-TRANSYT Interface consists in the following software components:

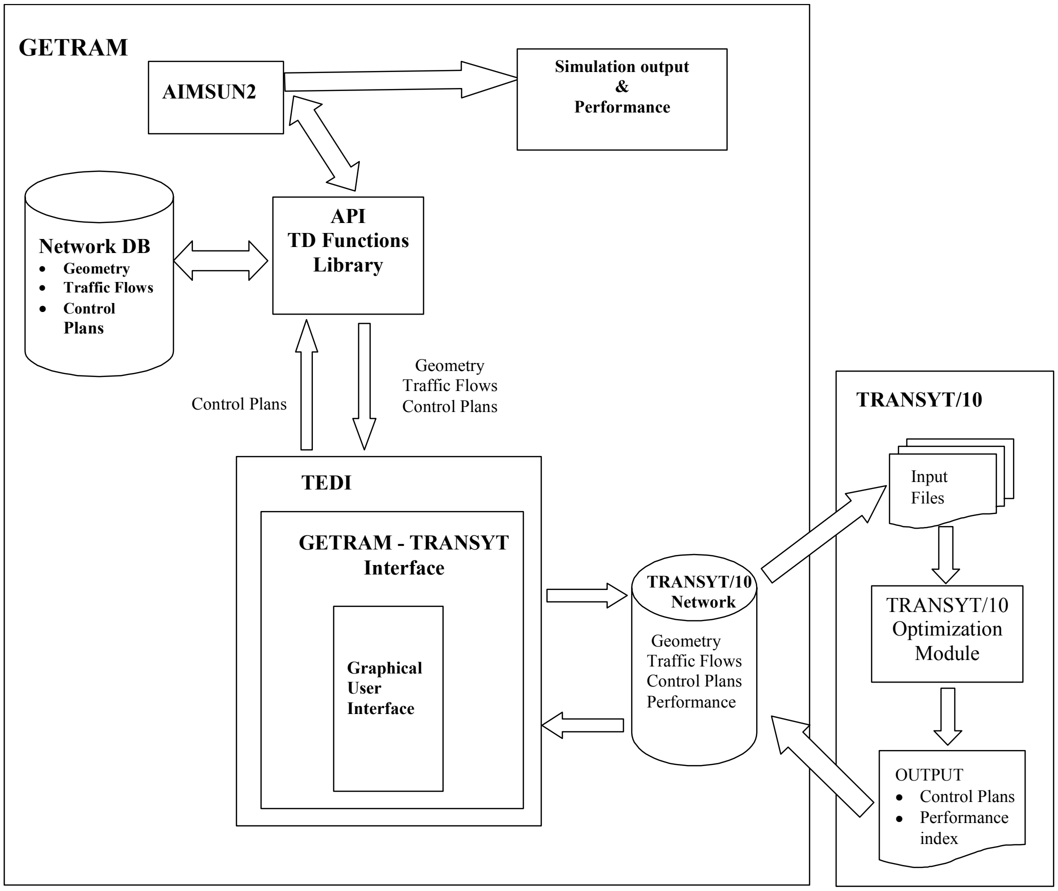


Figure 4: GETRAM – TRANSYT Interface

- Enhanced TEDI, the GETRAM editor, which includes the GETRAM-TRANSYT Graphical User Interface, through which the user is able to handle the different interfacing functions and access to TRANSYT data
- TRANSYT Network Database. It is a set of ASCII files where the TRANSYT data corresponding to a GETRAM network is stored. It is for the exclusive use of the GETRAM-TRANSYT Interface and can be accessed through the Enhanced TEDI.
- Translation Module. These are a set of procedures to translate complete GETRAM networks (geometry, traffic flows and control plans) into TRANSYT ones and TRANSYT control plans into GETRAM ones.
- Enhanced AIMSUN, which includes a module to calculate TRANSYT Performance Indexes.

The operations that are involved in the interface process are:

ii) Extract from the GETRAM Network database the data required by TRANSYT/10 and produce a preliminary version of a TRANSYT Network. The following data can be obtained:
    - Geometry: links, nodes, turnings, lanes, length, etc.
    - Traffic Flows
    - Initial Control Plan definition

iii) As not all the TRANSYT/10 required data is available from GETRAM, the user may need to complement the network data with some information, resulting into a new and consistent TRANSYT Network now containing all data required by TRANSYT/10. A Graphical User Interface that is embedded into the TEDI editor is built for that purpose. The GUI consists basically on a set of dialogues to facilitate the user the detection of lacking data, inconsistencies, etc. and the input of new data.

iv) TRANSYT/10 Input data file is automatically created. Using the data from the TRANSYT/10 Network database, the ASCII files in the format required by TRANSYT/10 are produced.

v) Run TRANSYT/10 from TEDI, using as input the TRANSYT/10 Input data file created.

vi) Control Plans calculated by TRANSYT/10 are imported directly into TEDI and can be saved as GETRAM Control Plans in order to use them as input for AIMSUN simulation runs.

AIMSUN can produce the same type of performance indexes as TRANSYT/10 and either stores them in the TRANSYT/10 database or printout reports. The user is able to see the performance indexes, both the ones calculated by TRANSYT and the simulated ones, through the TEDI Interface.

4. Interface with EMME/2

Traffic assignment models based on the user equilibrium approach (10) are one of the most widely used tools in transportation planning analysis. All the modeling hypotheses lead to nice mathematical models for, which there are efficient algorithms that provide solutions in terms of the expected flows on network links. Modeled flows offer a static average view of the expected use of the road infrastructure under the modeling hypothesis. This information has usually been sufficient for planning decisions. However, the evolution of advanced technologies and their application to modern traffic management systems require in most cases a dynamic view complementing the static estimates provided by the assignment tools. The planned infrastructure is probably sufficient for average demand, but time-varying traffic flows, i.e. at peak periods, combined with the influence of road geometry, can produce undesired congestion that can not be forecasted or analyzed with the static tools. This is a clear case for changing in the analysis methodology (11) and a sound reason for the combination of a well-known traffic assignment tool, the EMME/2 model (9), with a microscopic traffic simulator as AIMSUN. The convenience of interfacing a macroscopic approach for transportation modeling based on traffic assignment models, as implemented in EMME/2, and a microscopic approach, as implemented in the micro simulator AIMSUN, was already identified by many practitioners some time ago. There are also other reasons for requiring such an interface. In the case of a typical microscopic simulation based on generating traffic flows at input sections in the model and use turning proportions for emulating the behavior at intersections, quite frequently not all input flows and turning proportions are available by lack of the corresponding measurements. The result of a calibrated assignment model can in this case provide default values for these missing measurements. A direct communication between EMME/2 and AIMSUN would provide the user with a friendly tool to overcome the cumbersome task of inputting manually the data. Another reason to interfaces both systems is to exploit the powerful matrix calculator tools in EMME/2 for the benefit of AIMSUN to generate the adjusted Origin-Destination time sliced matrices required by the route based microscopic simulation, providing through the interface the means for exporting and importing trip matrices between both systems. To interface the representation of the network in EMME/2 with the representation of the network in AIMSUN requires that both network representations at macroscopic and microscopic levels are consistent, and that the same Origin-Destination matrix is used in both models. A way of satisfying accurately this requirement is use a model building tool that automatically translates the network representation at the disaggregate level, that of the microscopic view, into the aggregate level of the macroscopic view. This is the main objective of the new and improved interface between GETRAM/AIMSUN and EMME/2 included in Version 4.0. The interface has three main functionalities:

1. Translation of the a GETRAM road network model into an EMME/2 model
2. Generation of input files to AIMSUN from an EMME/2 assignment
3. Generation of a GETRAM network model mockup from an EMME/2 model

4.1 From GETRAM models to EMME/2 models

The network representation in GETRAM is very detailed representing the road in all its complexity as correspond to the higher level of disaggregation of the model information. An EMME/2 model simplifies the network representation in terms of a graph. This corresponds to the lower level of aggregation. The GETRAM-EMME/2 interface maps the GETRAM model into the EMME/2 model in a consistent way ensuring a coherent correspondence of the information between both levels. The logic of the model translation from GETRAM to EMME/2 is depicted in figure 5.

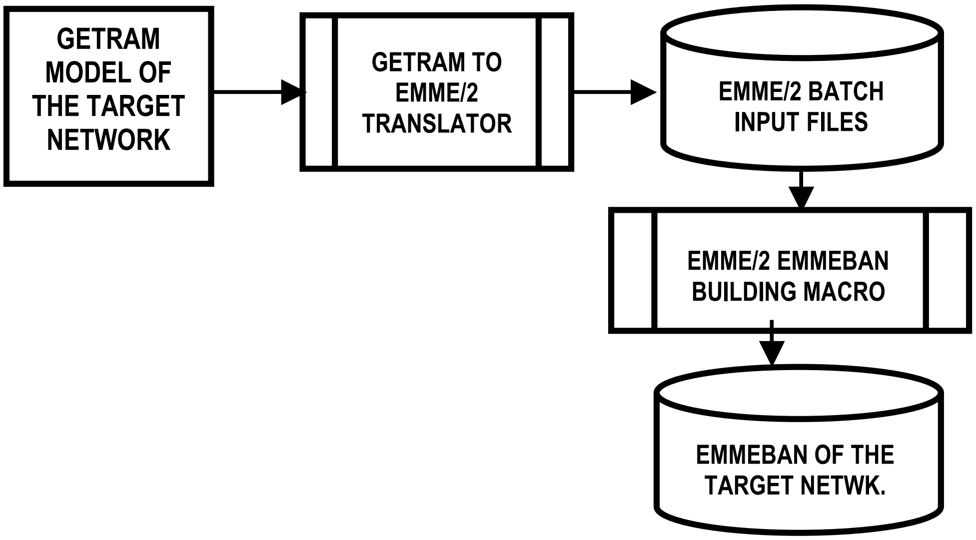


Figure 5: Conceptual logic of the GETRAM to EMME/2 translation process.

In the GETRAM environment working with TEDI when the translation process is activated it creates the batch input files to generate the EMME/2 model of the target network. The interface provides an emmeban building macro that reads the EMME/2 input files and generates the emmeban for the target network, then the user can continue the network analysis in the EMME/2 environment. The process works in the following way: The GETRAM network model is loaded in the TEDI graphic environment of GETRAM, and then clicking on the Files menu the function

Export is shown, as depicted in Figure 6a. Through the Export Function the user can identify the output directory where the EMME/2 environment is located, set up the equivalencies between centroids, set up the correspondences between vehicle types in GETRAM and transportation modes in EMME/2, and between sections in GETRAM and links in EMME/2, and their corresponding attributes (capacities, default volume-delay functions, etc.).

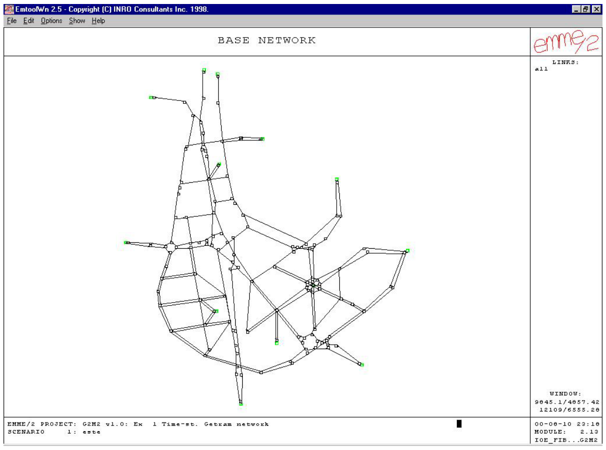


Figure 6a

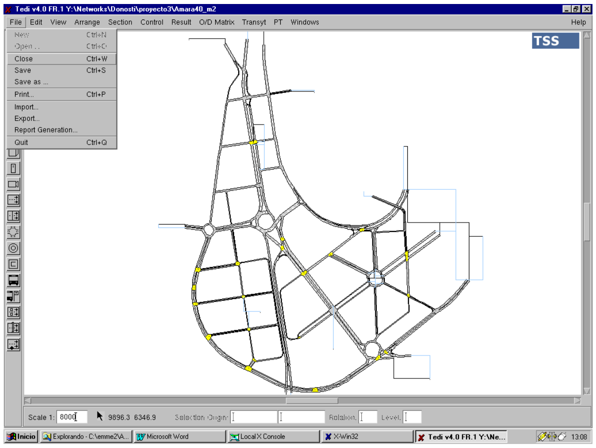
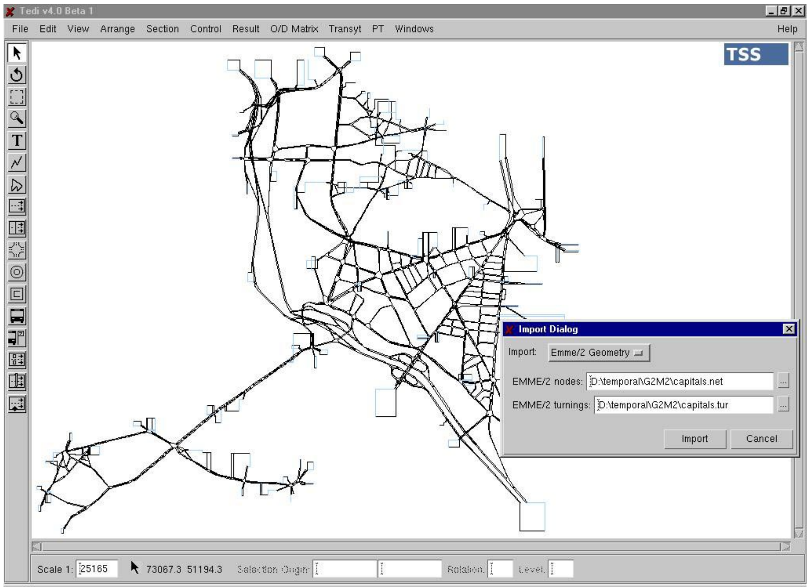


Figure 6b

Once the emmebank is created the user can continue the model analysis in the EMME/2 environment, i.e. adjusting the Origin-destination matrices, and once the process is completed import the new results into the GETRAM environment. In Figure 6b the automatically generated EMME/2 model for the GETRAM network in Figure 6a, is depicted.

4.2 From EMME/2 models to GETRAM models



Import: Emme/2 Geometry

EMME/2 nodes: D:\temporal\G2M2\capitals.net

EMME/2 turnings: D:\temporal\G2M2\capitals.tur

Scale 1: 25165

Figure 7. Import dialogue from EMME/2 to GETRAM

The reciprocal process, that is the creation of a GETRAM model from an EMME/2 model is also included in the new interface. However, it should be noticed that this process could not produce a complete GETRAM model. The reason is quite obvious. The way from a fully disaggregated level (GETRAM Network) to a aggregated level (EMME/2 model) can be defined univocally, but there is not a unique way of disaggregating the aggregated information, on the other hand if it happens that the GETRAM Network has an excess of information with respect to the information required at the EMME/2 model, the converse is not true. The EMME/2 model misses some of the information required by the GETRAM Network. Bearing this in mind the operation of the interface can be described as the creation of a "mockup" GETRAM model from an EMME/2 model, that later on can be refined using the TEDI graphic editors. Figure 7 depicts the generation dialogue and the result of the translation.

5. References:

1. J. Barceló, J.l. Ferrer, R. grau, M. Florian, I. Chabini and E. Le Saux (1995), A Route Based variant of the AIMSUN2 Microsimulation Model. Proceedings of the 2nd World Congress on Intelligent Transport Systems, Yokohama.
2. J. Barceló, J.L. Ferrer and R. Martin (1999a), Simulation Assisted Design and Assessment of Vehicle Gudance Systems, International Transactions in Operational Research, Vol. 6, No.111, pp. 123-143.
3. J. Barcelo and J. Casas (1999b) The use of neural networks for short-term prediction of traffic demand, in: Transportation and Traffic Theory , Avi Zeder editor, Pergamon Press.
4. J. Barceló, J. Casas, J.L.Ferrer and D. García (1999c) Modeling Advanced Transport Telematic Applications with Microscopic Simulators: The case of AIMSUN2, in: Traffic and Mobility, Simulation, Economics, Environment, W. Brilon, F. Huber, M. Schreckenberg and H. Wallentowitz (Eds.), Springer.
5. J. Barceló, J. Casas and J.L. Ferrer (1999d) Analysis of Dynamic Guidance Systems by Microsimulation with AIMSUN2, Proceedings of the 6th World Conference on ITS, Toronto,.
6. J. Barceló, (2000), The Role of Simulation in Advanced Traffic Management Systems, Proceedings of the 7th World Conference on ITS, Torino.
7. M. Ben-Akiva and M. Bierlaire (1999),Discrete Choice Methods and Their Applications to Short Term Travel Decisions, in: Transportation Science Handbook, Kluwer.
8. E: Cascetta, A. Nuzzolo, F. Russo and A. Vitetta (1996), A Modified Logit Route Choice Model Overcoming Path Overlapping Problems, in: Proceedings of the 13th International Symposium on Transportation and Traffic Flow Theory, Pergamon Press.
9. INRO Inc (1999), EMME/2 Release 9.0 User's Manual.
10. M. Florian and D. Hearn (1995), Network Equilibrium Models and Algorithms, in M.O. Ball et al. Eds. Handbooks in OR & MS, Vol.8, Elsevier,
11. L. Montero, E. Codina, J. Barceló and P. Barceló (Autumn 2000), Combining Macroscopic and Microscopic Approaches for Transportation Planning and Design of Road Networks, Accepted to be published in Transportation Research C.
12. TSS-Transport Simulation Systems (2000), GETRAM/AIMSUN Version 4.0 User's Manuals.