An Integrated Interface to Design Driving Simulation Scenarios

Salvador Bayarri[♣]*, Marcos Fernandez*[♦]*, Ignacio Pareja*[♣] *and Inmaculada Coma*[♦]
[♣]*Instituto Universitario de Trafico y Seguridad Vial (INTRAS).* [♦]*Instituto de Robotica.*
Universidad de Valencia. Hugo de Moncada 4, Entlo.Izq. 46010 Valencia. Spain.
Ph.: +34 6 3601472 Fax: +34 6 3607366
Salvador.Bayarri@uv.es, marcos@glup.eleinf.uv.es

Summary

The authors present the latest developments made at the INTRAS-LISITT driving simulation facility to integrate in a common platform the design tools, data formats and run-time processes required to manage the correlated layers of conventional simulation databases together with a scenario control system. These developments have been focused on the following aims: a) To integrate in a single tool the edition of the visual database, the logical structure of the network containing complex intersections, and accurate ground surface information, b) To develop friendly graphic interface function for designing the geometry of highways following engineering standards and automatically supporting design constraints; and c) To define a modular traffic simulation architecture able to support different behavior models. Examples of application are described.

INTRODUCTION

The research institutes in road safety and other driving simulation sites where complex experiments are frequently carried out need a stable software platform to design the different aspects of the experiment, ranging from the visual scene to the behavior of dynamic objects that are moving and changing during the simulated driving. Computer aid and modeling tools are a common approach to help the designer [*1*]. While there are some visual scene edition tools that can be used (some of them even include a road design module), no other software packages available in the market seem to be useful for the driving simulation experiment design purposes. This is the reason why many driving simulation and experiment developers have their own software components, data formats and, of course, run-time modules.

This was also the situation in the simulation environment of the Institutes of Road Safety and Robotics at the University of Valencia. Several tools were used to design different aspects of the driving simulation for experiments and other applications. Our aim now is to integrate as much as possible the functionality of these tools in a common environment and allow the programmers to use the resulting data output in a variety of applications. The kernel of the integration architecture is formed by a set of data formats and the associated software tools.

The following formats were used in previous versions of the simulator (see figure 1):

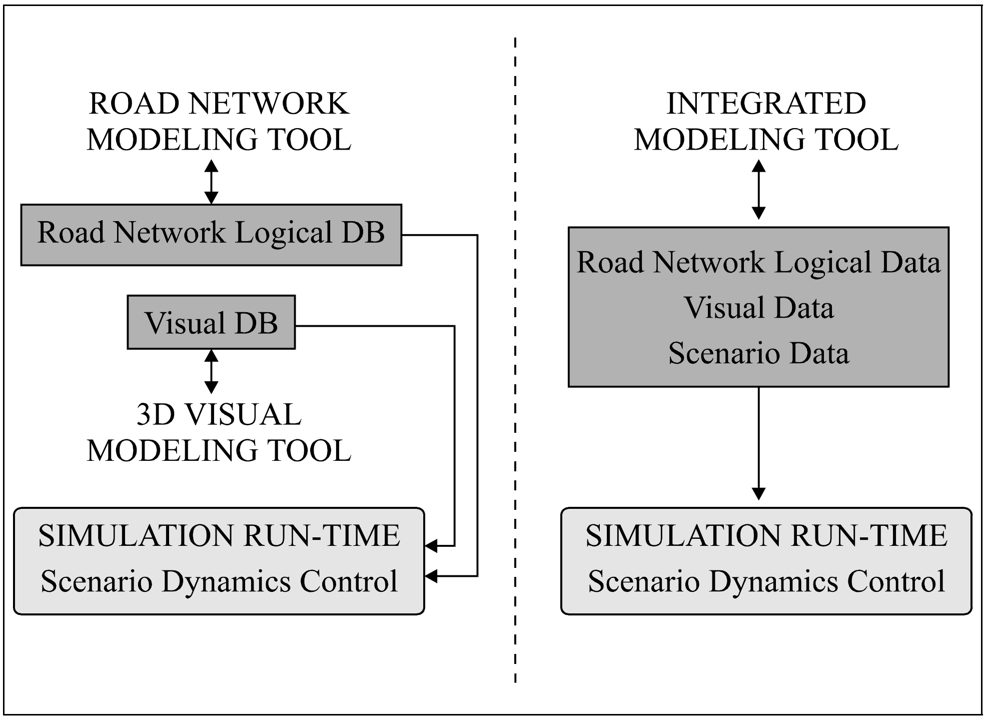

Figure 1:
Schema of data formats and tools before and after the integration process

Road Network Logical Format (.urb format): This kind of files contain a logical description of the road network, including the position of control points that define the geometry of road segments and intersection elements. All the connections among these elements are also defined here. In a more detailed level, some paths ('lanes') are created within the segments and intersection to help the vehicle motion simulation [*2*]. The tool associated to this data format, the road network designer was in charge of creating and very easily modify the road layout by using a simple 3D visualization environment. This tool can also generate a visual database.

Visual Scene Format (.mod format): This a 3D scene format including a hierarchical structure with features like levels of detail, transformations, and many other that are common to this kind of data files. It is completely equivalent to other real-time visualization formats like Multigen's OpenFlight format and can be loaded by Performer library to produce the visual output. The authors have developed their own modeling tool to edit the visual scene, a tool that has been used in many successful applications.

Later steps in the design process, as the definition of the scenario dynamics, were done in each specific case by programming them in the application, depending on the its requirements. One of the objectives of the integration plan was to put together the road network and the visual edition, as well as to increase some of its features. For instance, unlike other road tools, ours had a detailed representation of complex intersections, but it was using cubic piece-wise curves to approximate the road axis shape. We wanted to integrate real curves, like clothoid spirals in the road network.

Other aim of the integration was to include as much as possible some aspects concerning the scenario dynamics, experiment or application control and the simulation run-time architecture. As we will see, the work was also oriented by the demands of specific applications.

INTEGRATION OF SCENE DATABASE DESIGN TOOLS

The first task was to integrate the road network and the visual database edition. The reasons to do that seem obvious: to have an immediate perception of the final visual appearance of the road network and associated elements and keep automatically the coherence between the visual and the logical layers of the simulation database. Although some developers are using Multigen Road Tools to accomplish these aims, and it offers a good approximation to real freeways, we were not happy with it as a whole, and we were interested in including our intersection design functions, complete topology structure for road elements and paths, terrain functions, pagination functions and many other features. On the other hand, as we will describe later on, when we analyzed Multigen in order to use it in a large freeway design project we found that it did not offer the required flexibility in the road shape description and the user interface was really difficult to use.

A first problem is how to make compatible the hierarchical structure that is commonly used in real-time visualization with the existence of objects (road segments, intersections, lanes…) that are not a part of the visual data, although they must have a visual counter part. The second problem is how to represent the spatial conections, the information about topology, in this internal data structure.

To solve the first problem we have followed an approach that ourselves had previously used in order to integrate non-polygonal objects representations in a visual tree structure [*3*], and it is supposed to be the same that other tools have used. The solution is achieved by introducing new nodes in the visual structure that represent the road segments or intersections (see figure 2 up). These nodes do not produce any visual effect by themselves, but an automatic process is in charge of generating child nodes that contain new visual information each time the road element is modified, so that the road can be properly drawn.

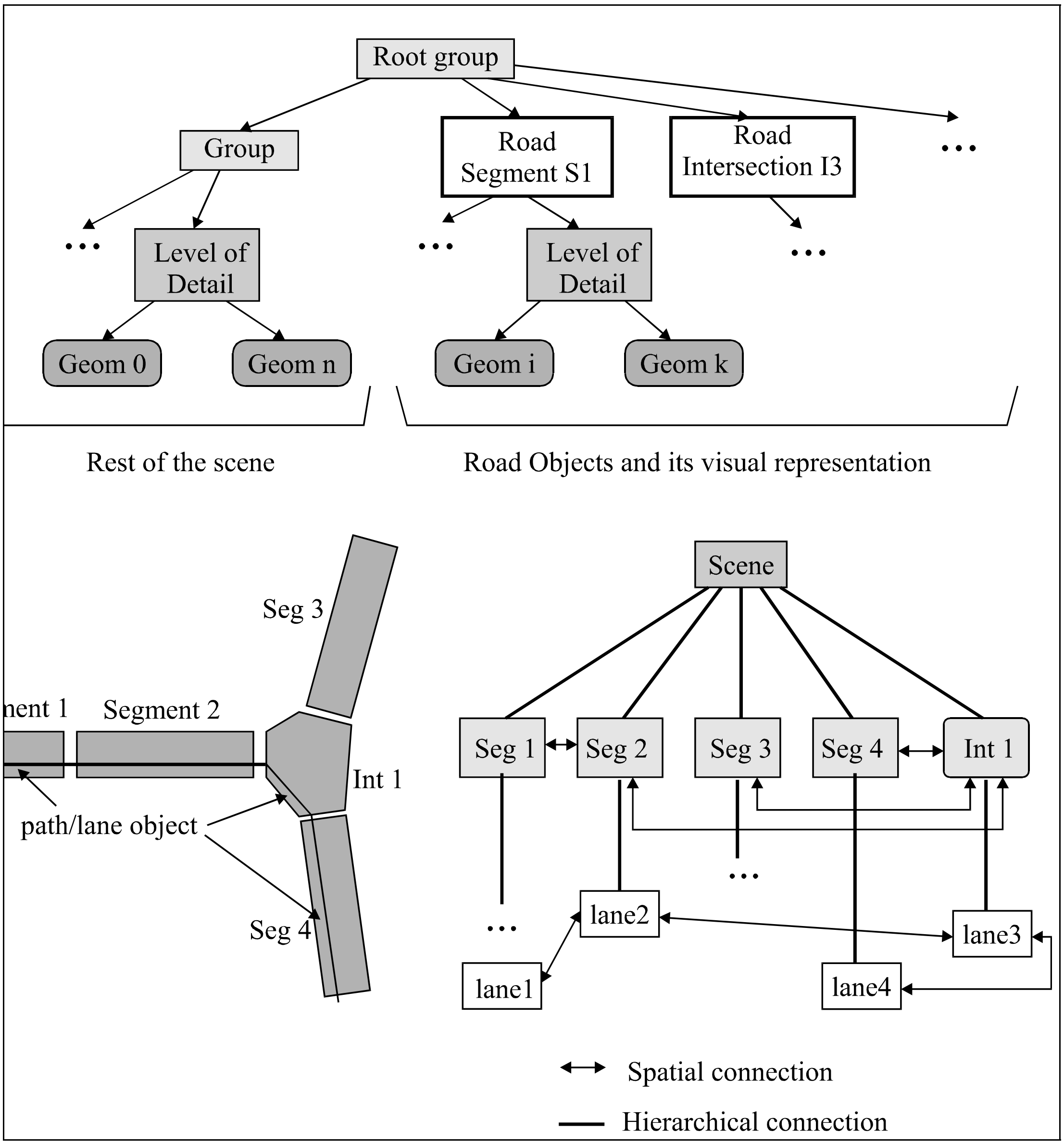

Figure 2:
Up: Integration of road element nodes and child visual nodes.
Down: Topology information at the road element and path levels in the internal data structure

The second issue, the integration of topology information, is approached by introducing references among the road element nodes. These references constitute a graph that joins all road network elements and that is used to maintain the continuity and coherence among them when edition operations are performed. In fact, there is a second level within the road graph, where the paths or lanes also appear as interconnected nodes (see figure 2 down). In this way, the road tool becomes a “what you see is what you get” design environment. Detailed changes in the polygon representation of the road elements are also possible, skipping the automatic generation mechanism, that must be locked in order to avoid their interference in the changes that are required to be made by hand. As a result, complex are realistic road elements can be generated.

The information of the visual and logical layers of the database are now stored together in a new, more sophisticated, format called ART (coming from the European ESPRIT project ARTIST from which the general ideas were taken). The tool can still read the old road network format so ready-made scenes can now be reused and improved.

THE PROBLEM OF A 3D INTERFACE FOR THE INTERACTIVE EDITION OF ANALYTICAL ROAD CURVES

While a polygonal decomposition of the road surface is needed to perform the visualization, an analytical description is required during the design process and also to perform accurate positioning on the road [*4*]. If we forget about the vertical profile, we can say that the road axis is described by segments of three types: straight, circle arc and spirals (clothoids). The main difficulty is caused by spirals, which are integral curves that cannot be defined parametrically, that is giving the position as an explicit function of the length, and thus must be integrated from the initial position and tangent depending on some geometric parameters, usually the total length and the initial and ending radius. Spirals are used to smoothly change the curvature radius between a straight segment and a circle arc, or between two circle arcs with different radius.

As far as we have seen, the existing tools (in fact, the Multigen one) have some limitations concerning the road shape edition. Multigen’s interface is based on the introduction of road sections including an initial and ending straight segment and a central circle arc preceded and followed by two spirals that interpolate the curvature to and from the straight segments (see figure 3 up). Thus, it is not possible to define a spiral that interpolates the curvature between two circle arcs of different radius, a configuration present is some high speed freeways where large radius arcs are used instead of straight sections.

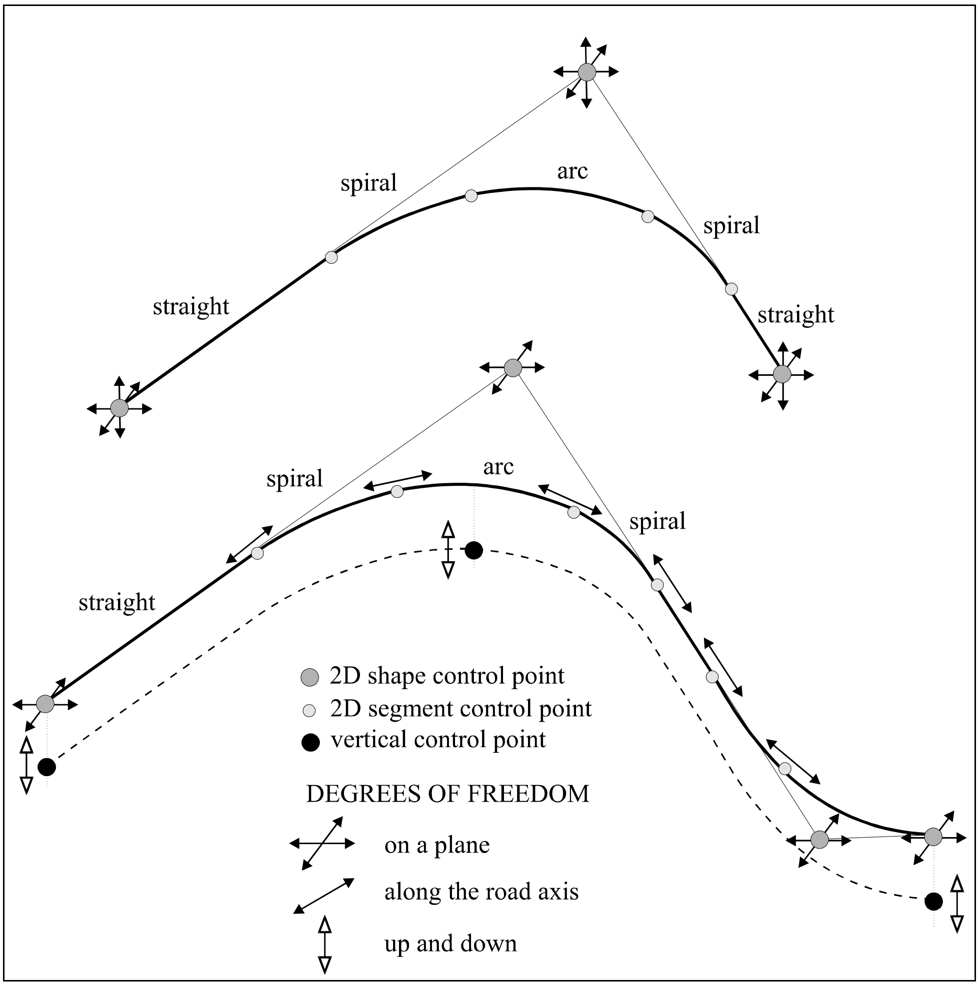

Figure 3.
Up: One limited approach to 2D road shape design, based on sections.
Down: A new approach for road shape control based on three sets of control points

Another major limitation concerns the management of incompatibilities among the parameters of the section. Spirals are defined using the total length and the ending radius, but these values maybe not compatible with the parameters that define the rest of the road section. The user can interactively move the three control points that define the section and change the length value of straight and spiral segments. Sometimes this causes strange results due to incompatibilities in the parameters.

We propose a different approach based on two sets of control points (see figure 3 down): the ones that define the sections and some others that allow direct control on the length of each curve segment. The compatibility is ensured by two simple rules. The first rule says that if the user moves the section control points, that have two degrees of freedom, the program tries to keep constant (or with their default values, if the section has been just added) the parameters, but the values are changed if any incompatibility appears, informing the user about this change (for instance, by blinking the element). The user can also move the secondary control points, with only one degree of freedom along the axis, to change the local parameters, but this motion will be limited by a second rule, so that no incompatibility appears. A new type of section including a spiral between two arcs will also be added, so that this type of element can be considered in the design.

Other main question in road design is how the 2D layout relates to the vertical profile and superelevation. Road engineers define the vertical profile as completely independent of the 2D layout, so that it is an unrealistic constraint to use the same control points for both, as it is done in Multigen. That is why to have a flexible model there is the need of an independent set of control points for the vertical profile. Concerning superelevation, although it is usually designed based on the curvature, it is convenient to allow the designer to change its specification by hand, since sometimes the default rules must be changed to accommodate special needs (for instance, rainfall water removal).

Other aspects that we are facing in the integrated tool are, of course, the connection of these curves to intersections, the option of easily insert and delete sections (or insert intersections) within the road sequence, and the introduction of different criteria that may define an automatic road tracing process in a given terrain. Some first results of the polygon generation functions can be seen in figure 4.

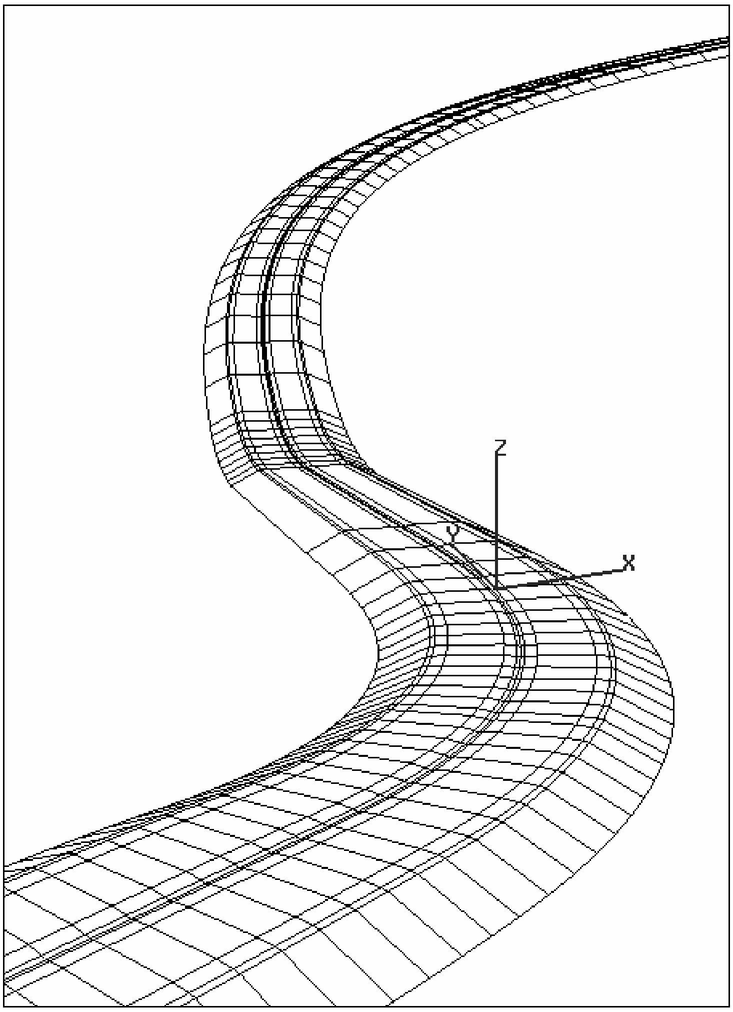

Figure 4:
An example of realistic road generated from by the independent control points model

INTEGRATION OF SCENARIO DYNAMICS DESIGN FUNCTIONS

Surrounding traffic is a key aspect in the use of driving simulation for human factors research, so it is very important to make easier the specification and configuration of the traffic behavior. We have been working in the development of a new graphical scenario design tool closely integrated with the scene design tool. Its aim is to provide us with a way to design traffic flow, as well as to include events to trigger specific actions in the traffic behavior.

The first part of the tool was developed at the Nissan Cambridge Basic Research Center in Boston MA. This part consists mainly of an interface to specify the main parameters of the *ambient* traffic (see figure 5) in some experiments in which we wanted to create a surrounding traffic that made the driver feel ‘comfortable’ with the driving environment. The researcher is able to select parameters as the distribution of vehicle types (heavy/sport/regular cars) as well as average speed, acceleration and maximum brake for each kind of vehicles. After setting these variables, the system will automatically generate a randomly distributed traffic [5].

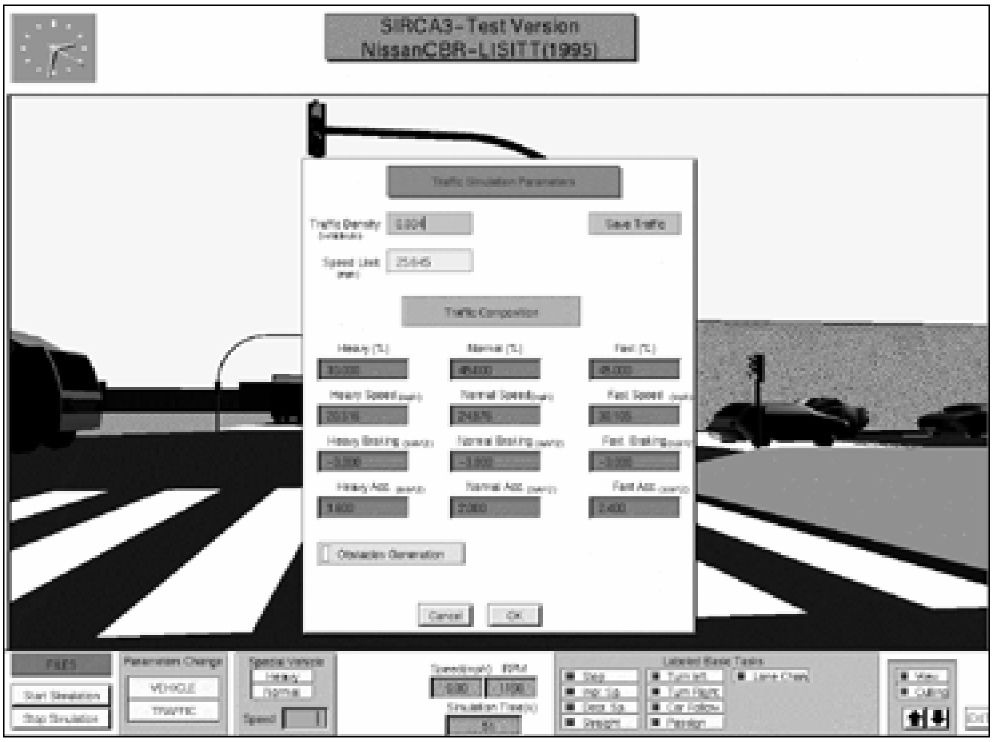

Figure 5:
Interface to control ambient traffic parameters in the application at Nissan CBR

This kind of control interface is suitable for experiments (for instance, the AUMAR application that we will mention later) where it is not required to have full control on the behavior of specific vehicles neither repeatability in the traffic configuration. To extend the scope of application of our traffic models we have now added new features to our vehicle control system. The main one is based on the description of pre-defined paths for simulated vehicles and other dynamic objects. A new module of the design tool allows the researcher to define the path to be followed by each object by clicking on the road surface, forming a sequence of points that is later user during run time to produce very efficient interpolation of the object motion. The points in the sequence can be also used to define the evolution in time of some variables of vehicle behavior, like the state of brake lights. Two different visual representations of the sequences in space and time edition windows constitute a flexible specification system similar to those used in computer animation (see figure 6).

A ‘situation’ can consist of any number of such sequences, and can be stored in an independent file. This is necessary, since many traffic situations can share the same scene, and so the same static database file. As we will see later, these ‘situations’ have special meaning in instructional applications of the simulation system.

The previous approach to traffic simulation presents the drawback of being not reactive to the driver behavior, so we are now working in a more advanced scenario tool that is based in the combination of both autonomous and pre-defined characteristics in the dynamic objects. It is possible to specify certain events that trigger specific actions upon vehicles to produce repeatable situations that the experiment want to evaluate. The location of the events is done by using the graphical tool and the events can be selected from a library of configurable events and behavior patterns (traffic lights change, aggressive lane change, etc.). There will be also an open API to extend the library of pre-defined events and behaviors.

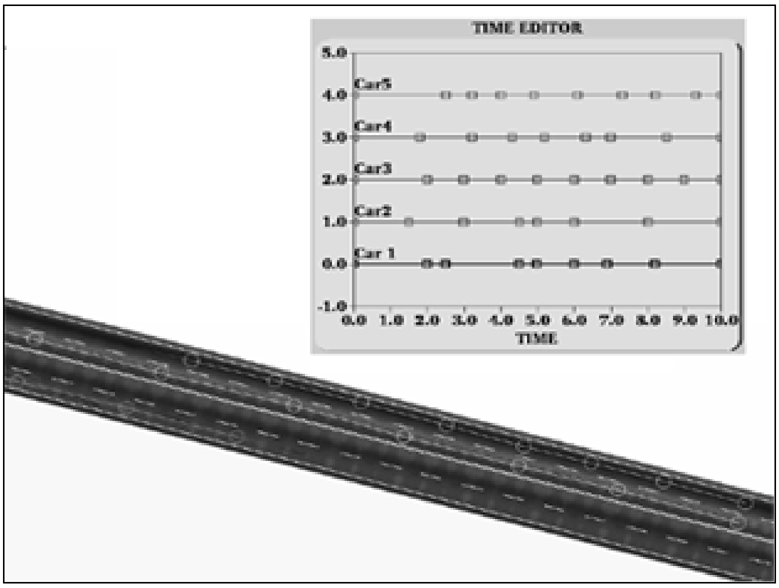

Figure 6:
Interface for pre-defined sequence space-time definition in the integrated tool

DYNAMIC SIMULATION ARCHITECTURE

The dynamic traffic control system is based on an object-oriented architecture that provides the scenario designers with a very flexible way to control the behavior of individual vehicles as well as to control and synchronize the global traffic situation.

The model represents each vehicle as an object containing certain attributes that describe its internal goals, and its own safety limits, as well as others concerning its visual appearance. To control the behavior of the vehicle at the operational level the system contains a set of behavior control objects (BCO) which perform the control of the vehicle for a particular task.

For instance, there is a BCO Car-Following server that takes care of the car-following task for the vehicles, a vehicle can subscribe to this object or cancel the subscription if it is no longer needed. After one vehicle has subscribed to one of these servers, the BCO will take care of controlling the vehicle evolution. This offers the flexibility of having several BCO Car-Following objects based on different simulation approaches (fuzzy, deterministic, etc.) and the vehicles can select one or another as a function of its internal state.

Upon these BCO that represent operational task controllers there is another set, the Tactical Control Objects (TCO) that take care of the tactical level of the simulation (i.e. to decide whether or not to perform a lane change or passing a vehicle). The TCO will select the right action at each time based on the internal state of the vehicle and taking into account the overall traffic configuration and the constraints imposed by the pre-defined events.

APPLICATIONS

This integrated design system has been already used to generate the scenarios for some applications of driving simulation, focused on different objectives. Within the ARTIST ESPRIT project, the tool has been used to design the scenarios of a racing game. It is currently been used in the design of an experiment that involves the modeling of hundreds of kilometers of the real toll freeway of AUMAR company going from Tarragona to Alicante in the Mediterranean coast of Spain. This simulation will include a huge visual complexity and ambient traffic, making use of the lanes generated by the tool. Its objective is to test different hypothesis concerning the activation level of drivers depending on the road environment. Real and simulated driving results will be compared.

A different approach has been followed in the SIVAS system, an instructional tool for driving schools based on real-time simulation of traffic situations, that makes use of predefined motion sequences to perform short simulations with educational contents. The driving school teacher can use a simple interface to present the traffic situations and change the point of view and some other options [*6*].

The scenario design system was first used jointly with a urban driving simulation environment by the authors at Nissan Cambridge Basic Research Center to analyze the behavior of drivers when performing certain maneuvers in order to compile data able to generate Hidden Markov Models of such tasks.

REFERENCES

[1] Bayarri S., Fernández M. and Sanmartín J. "Scenario Modelling Tools for Driving Simulation Experiments”. First International Conference on Traffic & Transport Psychology. 22-25 May 1996. Valencia. Spain.

[2] Bayarri S., Fernández M. and Pérez M. "Virtual Reality for Driving Simulation”. Communications of ACM. May 1996.

[3] Bayarri S., Pérez M., Coma I., Martínez F. and Lozano M.. “Arquitectura de Programación para un Modelador 3D”. Advanced Computing and Telematics Applications. N.4 July 1996. ISSN: 1133-8970. University of Valencia. Spain.

[4] Artz B. E. "An Analytical Toad Segment Terrain Database for Driving Simulation". Proceedings of *Driving Simulation Conference '95*. Sophia Antipolis. France, 12-13 Sept. 1995. ISBN-2-87717-051-9. Ed. Teknea, 1995.

[5] Fernández M., Liu A., Bayarri S. and Pentland A. "Tuning of Driver Behaviour Models Used in Driving Simulation”. First International Conference on Traffic & Transport Psychology. ICTTP’96. 22-25 May 1996. Valencia. Spain.

[6] Bayarri S., Fernández M. et al. “Sistema de Formación Vial Asistido por Simulación Visual”. Proceedings of Spanish Conference on Computer Graphics CEIG’97. Barcelona, 25-27 June 1997.