

A framework for ambient traffic in the IZVW driving simulator

Martin Grein, Armin Kaussner, Hans-Peter Krüger, Hartmut Noltemeier

IZVW (Center for Traffic Sciences at the University of Würzburg)
Roentgenring 11
97070 Würzburg, Germany
e-mail: grein@psychologie.uni-wuerzburg.de

Abstract

We present a framework for ambient traffic in driving simulators. Each vehicle of the ambient traffic includes a model of a driver with a free configurable behaviour. The behaviour of the driver can be changed online during simulation. These behavioural models are loaded at runtime, so the framework can easily be updated to integrate special behaviour. The supervisor of an experiment can also configure the driver model of each ambient vehicle in dependance of the test drivers behaviour. To integrate computational expensive driver models, the traffic framework can be distributed on several computers in the network. Each computer calculates a subset of the ambient vehicles and their driver models. Therefore, the ambient traffic has to be synchronized via the network. The behaviour models control a low level vehicle simulation by commands like target accelerations and velocities. In this layer for each vehicle the framework computes simple dynamics simulations. The supervisor of an experiment defines situations using a simple scripting language. During simulation the framework ensures that a situation takes place at a certain point in the roadnetwork, even if the situation depends on the behaviour of the test driver.

Introduction

If simulation is used as a research tool, a flexible ambient traffic environment is needed which fulfills the following requirements:

- **realism**
from the view of the testdriver, the traffic has to behave as realistically as possible.
- **reproduceable behaviour**
within different runs of the same situation, the ambient traffic has to have the same behaviour as before
- **controllability**
it has to be guaranteed that predefined traffic situations will occur despite difficult driving conditions of the situation
- **measurement**
the researcher must be able to define which data from the ambient traffic are relevant for recording

The basic framework of the ambient traffic – as it was developed at the IZVW driving simulator – is structured in modules mainly

- **behaviour of a vehicle**
- **physics of a vehicle**
- **flow control for generation of situations**
- **interface to scenery database**

These modules are organized in separate libraries. These libraries can be loaded at run-time. Therefore only the changed modules have to be compiled.

Layer model

The driving simulation needs different information from the traffic framework:

- **VISUALIZATION:**
the visualization module generates an image of the vehicles and the road network
- **TRAFFICCONTROLLER:**
to generate a special situation for the testdriver, the trafficcontroller has to activate different vehicles at their positions and give them their special tasks
- **BEHAVIOUR:**
to generate realistic traffic flows, the behaviour of every traffic vehicle has to react to the behaviour of every vehicle in the neighbourhood.
- **MESUREMENT:**
in order to evaluate the traffic situation and the behaviour of the testdriver information is needed about the position of every relevant vehicle

The calculation for the different components like visualization, ambient traffic and road network database has to be distributed between different computers. For each visualization channel, in the IZVW driving simulator, one additional computer has to be integrated into the

simulation network. The main computer yields data like the position of the simulation car to every visualization computer in the network. Additionally the visualization computers have to know the position of every vehicle in the traffic environment.

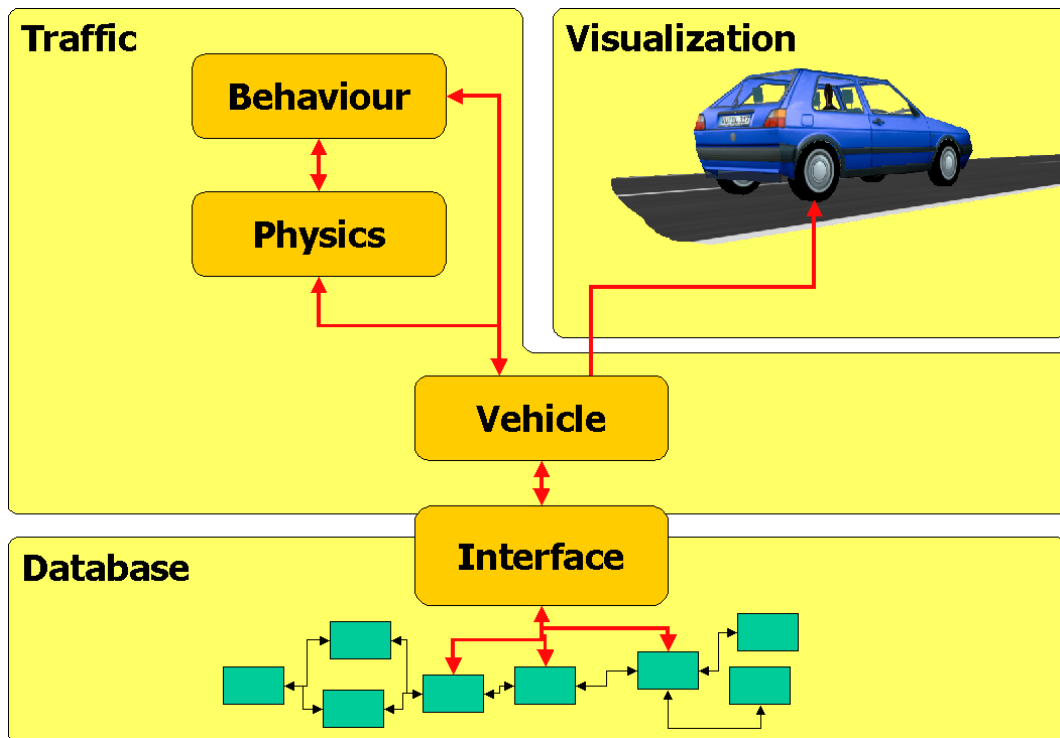


Figure 1 views of traffic and visualization on the vehicle

In order to be able to represent more complex behaviour of any traffic vehicle, the computation of traffic behaviour can be distributed between several computers which are integrated in the simulation network. These computers need also complete data of every position of the traffic vehicles (incl. vehicle speed, acceleration and position of the simulation car).

Since the ambient traffic is calculated on different computers, a meaningful compromise between data communication and computing time must be found. If all vehicle-relevant data like position and angle, road network section, condition of the lights, speeds and accelerations would be sent via the network, the capacity of a 100Mbit network would be far exceeded. For this reason only the commands for the vehicle control are sent and the positions of all vehicles are computed on each computer involved separately.

The lowest layer of traffic is represented by the TrafficManager. It is computed on each computer, each of them having access to all vehicle positions. The TrafficController is computed by all computers which deal with the behaviour models.

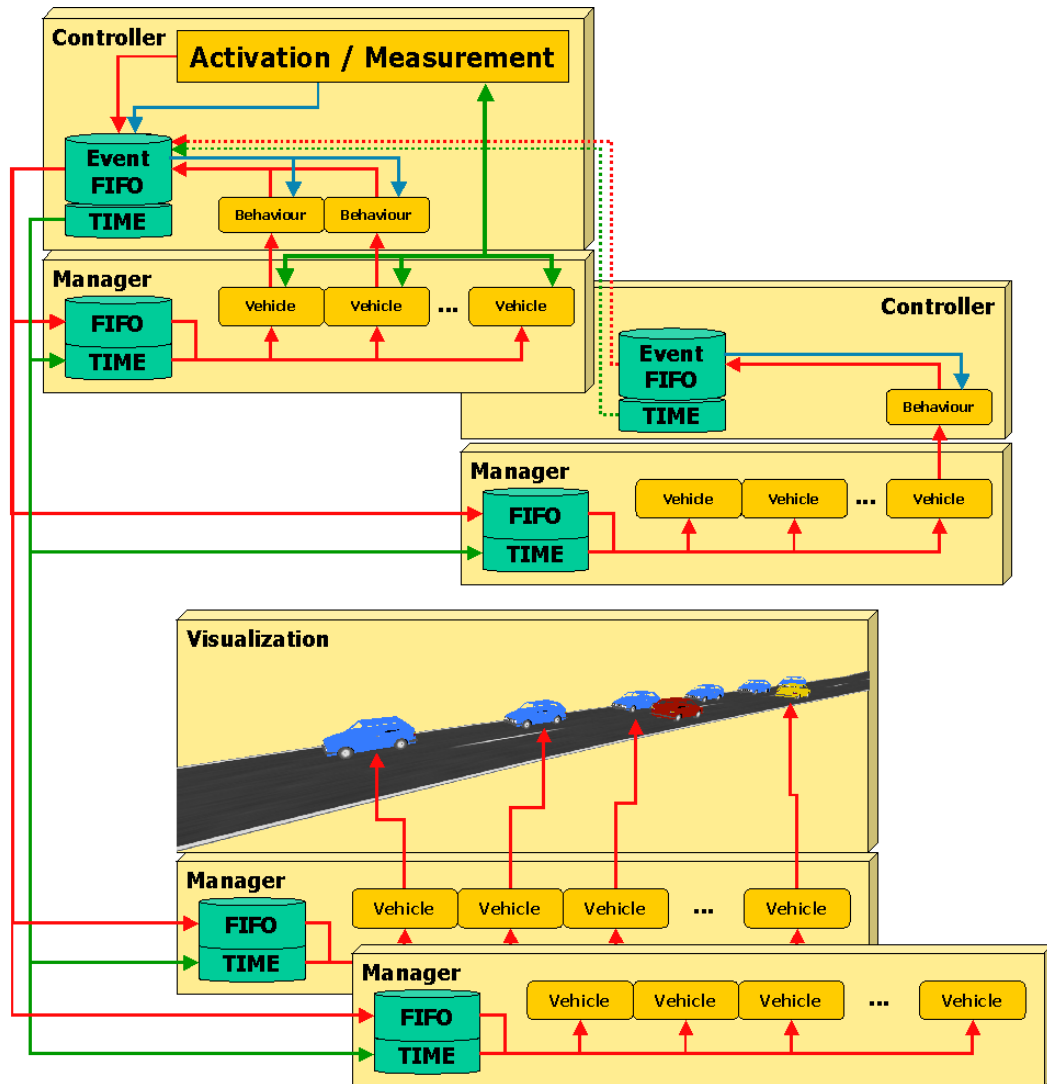


Figure 2 Layers of the ambient traffic framework

Vehicle

The Vehicle is controlled by a set of commands (events). It is activated by an instruction in the database at a defined point. Furthermore either the speed or the acceleration of the vehicle can be set by events. Based on these data and with regard to the database the current position of the vehicle on the lane is calculated.

The command set also contains parameters which are necessary for the vehicle to perform manoeuvres like lane changing. Additionally instructions for turn signals, brake lights or headlights can be given. Even the type of the vehicle can be changed during driving.

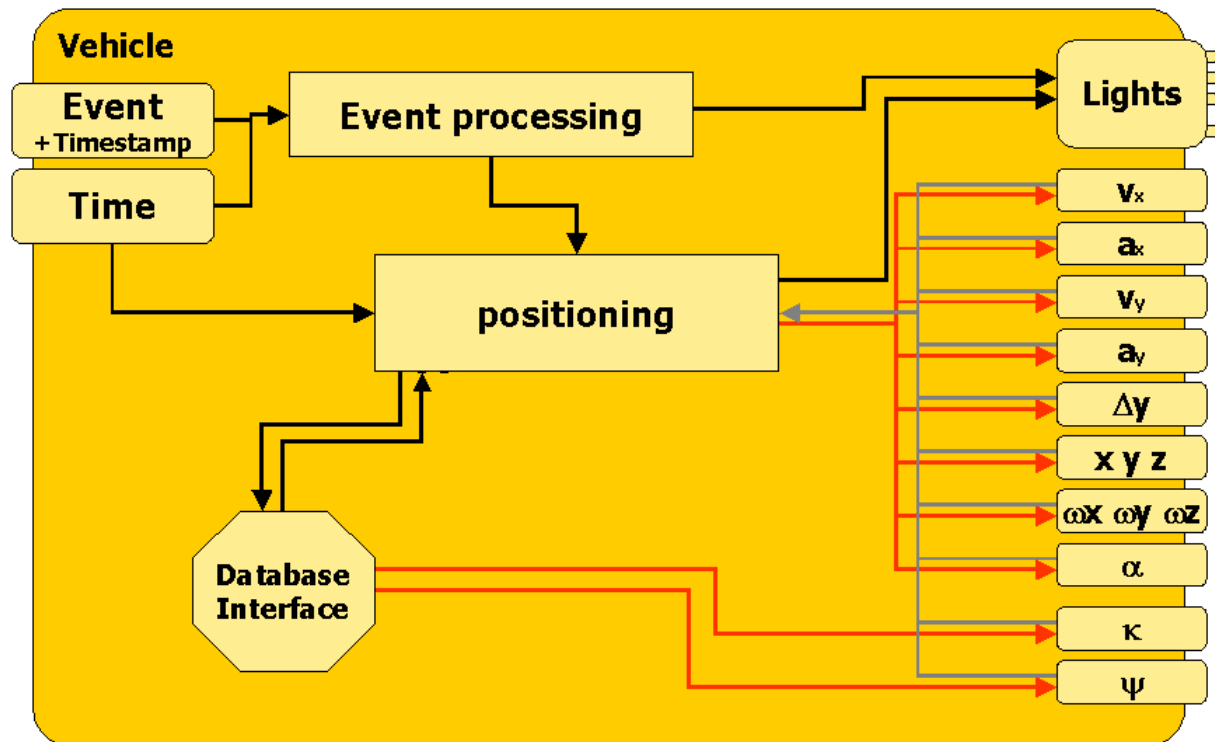


Figure 3 in- and outputs of the vehicle class

TrafficManager

The TrafficManager performs the calculations for all vehicles. Therefore, each computer which needs information about the vehicles of the ambient traffic, has to run the TrafficManager by itself. The TrafficManager receives the commands for all vehicles from the main TrafficController.

TrafficController

The TrafficController calculates the behaviour models of the individual vehicles. A behaviour model has to analyze the positions of the vehicles in the TrafficManager in order to get an overview of the traffic situation. The computation of complex behaviour models can be distributed within a computer network. Each computer in the network obtains information about the current traffic situation from its TrafficManager.

Physics of an vehicle

The simulation of vehicle physics have to be discretised to minimize the number of vehicle events. For example, a continuous speed process is replaced by an acceleration and the final speed.

Behaviour of a vehicle

A set of behaviour machines is defined for each vehicle. The priority of a machine is defined by their hierarchical order. The behaviour machines are implemented as finite state machines (like HCSM in [3]). The set of behaviour machines for a vehicle can be configured freely.

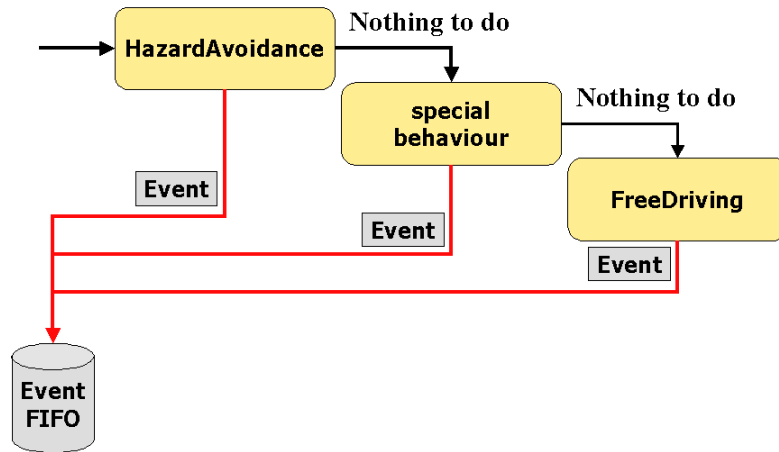


Figure 4 hierarchical set of behaviour machines

The behaviour machines send events to their vehicle. In figure 4 the set of behaviour machines consists of three machines. The HazardAvoidance prevents the vehicle from driving against the vehicle in front, by sending a negative acceleration. If the HazardAvoidance is not engaged, the control passes to the next machine and so on. In this way each desired behaviour can be produced by the composition of the behaviour machines. The priorities and the machines can be exchanged during simulation. Since each machine is implemented in a separate shared library, it can be loaded at runtime.

Source and drain flows

In the database, vehicle sources can be positioned. They create vehicles with a certain behaviour at the position of the source. These vehicles drive along a given course. A source activates the vehicles in a given time sequence. If the vehicle gets into the reach of the drain, it is deactivated. It can be reactivated by the source again. A drain is always associated with a specific source.

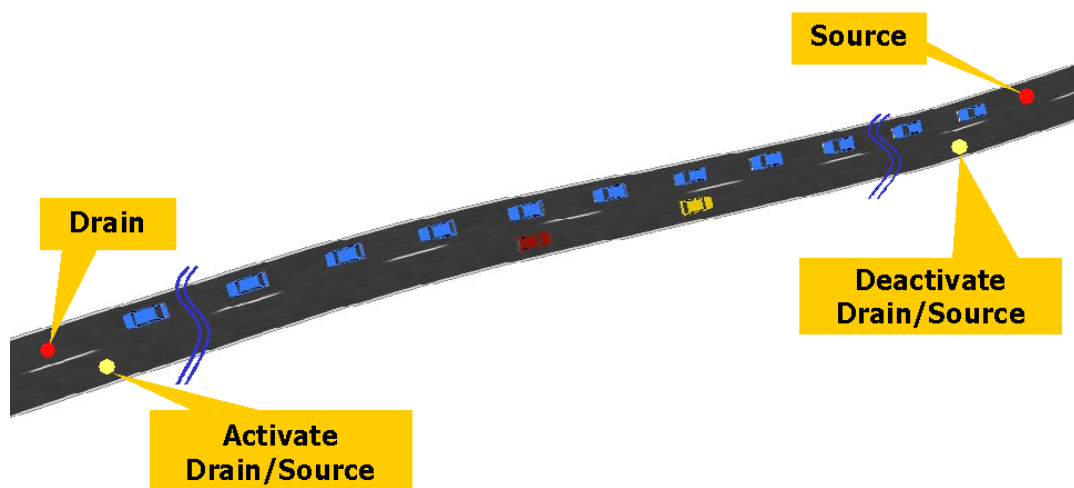


Figure 5 driving source and drain

In addition to the fixed positioned sources, sources can be attached to the simulator vehicle itself. For example, this can be used to produce oncoming traffic: A source moves with the

simulation vehicle on an oncoming lane. Behind the simulation car, a moving drain deactivates vehicles coming from the source. This method of generating oncoming traffic needs a small number of vehicles: Only a small region around the simulation car is filled with ambient traffic. The sources and drains are switched on and off via waypoints that are passed by the simulation car. Likewise, the time sequence of a source can be changed by these waypoints.

Scripting language

The researcher defines traffic flows in terms of sources and drains. For special tasks, he can define the behaviour of each single vehicle.

```
01 Trafficflow TF1
02 {
03     Source MS1
04     {
05         Position=(SimCar, 600,0);
06         Vehicles={(15, golf.winered),(12, golf.blue)};
07         SequenceMachineTimeRandom MQR
08         {
09             BehaviourMachine=((HazardAvoidance, 1.5),
10                               (FreeDriving, 27.78, 27.78, 1.3,-2.5));
11             Parameters = (Gauss, 4,1);
12         };
13         SequenceMachine= (MQR);
14         Flowpoints=
15         {
16             (S1, 570, 0, Traffic, Speedlimit, 13.89, 300),
17             (S2, 499, 0, Traffic, EndSpeedlimit)
18         };
19     };
20     Drain MD1
21     {
22         Position = (SimCar,610);
23     };
24     Flowpoints=
25     {
26         (S1,10,1, SimCar, ActivateSource, (MS1), ActivateDrain, (MD1)),
27         (S2, 2950, 1, SimCar, DeactivateSource, (MS1))
28     };
29 };
```

Script 1 Definition of a traffic flow

Script 1 describes the oncoming traffic (traffic flow TF1) on the lane 0 (numbered from left to the right). It defines a source (line 3), which position lies 600 m in front of the simulator car. The source will be positioned 600m in front of the simulator car on lane 0 (line 5). In line 6, the researcher determines that the source can use 15 red and 12 blue cars to generate the traffic flow.

The source activates vehicles with a given set of behaviour machines (line 9-10). Two machines are assigned to the vehicles: In line 9, a HazardAvoidance machine with 1,5

seconds distance to the front vehicle is assigned to the vehicles. The FreeDriving machine in line 10 tries to reach a target speed of 100km/h (27.78m/s). The HazardAvoidance machine has a higher priority than the FreeDriving machine. Time sequence is realized by a random series (line 11), with an average value of 4 seconds and a standard deviation of 1 second. A drain (MD1) moves behind the position of the simulation car (line 20). This drain deactivates every vehicle, which has a distance greater than 610m to the simulation car. MD1 deactivates only vehicles, which have been activated by source MS1.

The source and drain are controlled by the simulation car via the flowpoints specified in the lines 24-28. MD1 and MS1 are activated on course S1 at 10m. They are switched off, if the simulation car passes the point on course S2 at 2950 meters.

Measurement

With a simulation frequency of 100 Hz and 40 vehicles driving in the scene, the ambient traffic generates more than 120 kByte of data per second. In order to reduce the recorded data, a selection has to be made.

The vehicles are classified into 24 categories which might be relevant in a given situation. They are classified in relation to the simulator car. The vehicle driving in front of the simulation car on the same lane is labeled as `SameAheadNext`. The vehicle on the lane left behind the simulation car is labeled `SameBehindNextLeft`. If the vehicle on the left is on an oncoming lane, it is labeled `TowardAheadNextLeft`.

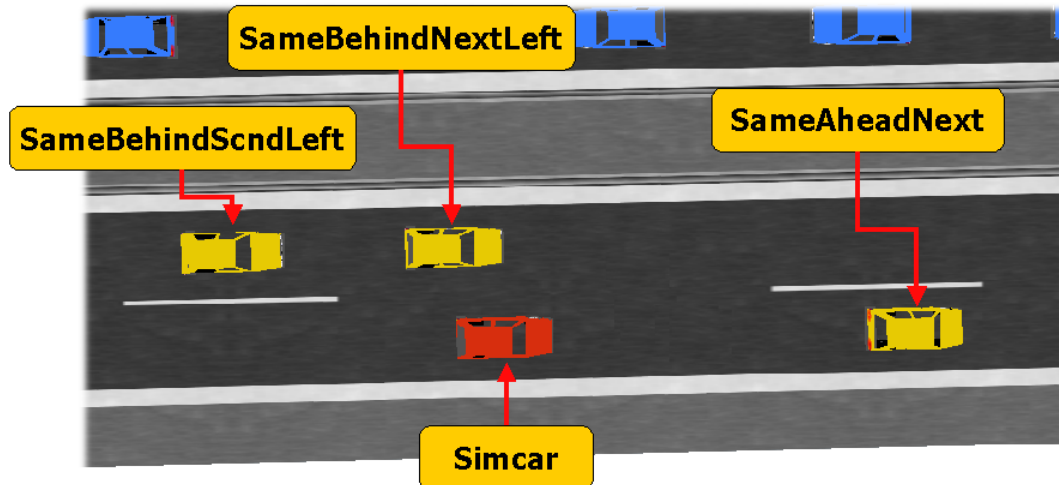


figure 6 measurement of ambient traffic

The values lateral distance, longitudinal distance and speed were recorded. The longitudinale distance describes the distance on the lane between the simulator vehicle and the classified vehicle. The lateral distance describes the distance between the lanes of these two vehicles. Therefore, 24 classifications of the next surrounding vehicles can be specified. However, this classification is not sufficient to describe all relevant vehicles when driving through an intersection. The researcher can define a mesurement along a lane, starting at a defined point

on this lane. This measurement describes the distance of the first vehicle on this lane to the defined point.

```
01 Measurement M1
02 {
03     Out1= SameAheadNext;
04     Out2= SameBehindNext;
05     Out3= TowardsAheadNextLeft;
06     Out4= TowardsBehindNextLeft;
07 };
08 Flowpoints =
09 {
10     (SK1,10,1,SimCar,ActivateMeasurement,M1),
11     (AUS,5,1,SimCar,DeactivateMeasurement,M1)
12 };
```

Script 2 Definition of a measurement

The researcher defines the pattern M1 as a measurement in line 1. Output 1 is connected to the vehicle in front of the simulation car (line 3). The vehicle in front of this is connected to output 2 (line 4) and so on. In Line 10, the measurement is activated, if the simulation car passes the point at 10m on the course SK1 on lane 1 (the lane on the right side of this road). Three columns in the recorded data file are reserved for eight (Out1 to Out8) vehicles which might be relevant in a given situation.

Conclusion

The presented ambient traffic framework permits a flexible definition of traffic conditions in the driving simulation. Individual behaviour of ambient vehicles can be adapted for special tasks. Traffic flows can be realized with vehicle sources and drains.

The researcher can define the ambient traffic with a scripting language. He is also able to select a set of recorded data which is relevant for the defined situation.

References

- [1] M. Grein, A. Kaussner, H.-P. Krüger, H. Noltemeier: A flexible application framework for distributed real time systems with applications in PC based driving simulators, Proc. DSC 2001, Sophia-Antipolis, 2001
 - [2] A. Kaussner, M. Grein, H.-P. Krüger, H. Noltemeier: An architecture for driving simulator database with generic and dynamically changing road networks, Proc. DSC 2001, Sophia-Antipolis, 2001
 - [3] J. Cremer, J. Kearney, Y. Papelis, HCSM: A Framework for Behavior and Scenario Control in Virtual Environments, ACM Transactions on Modelling and Computer Simulation, 1995
 - [4] M. Bartelme, M.Booth, J.Cremer, Experiment authoring for virtual driving environments, 1st Eurographics Workshop on Virtual Environments, 1993
 - [5] J. Cremer, J. Kearney, Scenario Authoring for virtual Environments, IMAGE VII Conference, Tucson, Arizona 1994
 - [6] J. Cremer, J.Kearney, Driving Simulation: Challenges for VR Techonology, IEEE Computer Graphics and Applications, September 1996
-