SILAB – A Task Oriented Driving Simulation

Hans-Peter Krueger, Martin Grein, Armin Kaussner, Christian Mark

Center for Traffic Sciences, University of Wuerzburg

Roentgenring 11

D-97070 Wuerzburg, Germany

Krueger@psychologie.uni-wuerzburg.de

Phone: +49 931 31 26 53

Fax: +49 931 31 26 16

Abstract

The core element of traditional simulation is a fixed database which is overlaid by a traffic model. Despite obvious efficiency of simulator use, shortcomings must be seen in (1) imperfect reality and control, (2) fixed sequencing of scenarios which cannot be altered during simulation, and (3) missing of objective evaluation of drivers' performance. Therefore, at the University of Wuerzburg a new concept for driving simulation was developed basing upon scenario elements which arbitrarily can be linked together during runtime. The scenario elements are entities consisting of road environment, traffic configuration and task criteria. Combining these elements during runtime enables the user to adapt the temporal sequence of scenarios to any other variable (e.g. learning status of a trainee, state of the driver). Together with a newly developed online assessment module, this architecture allows adaptive training as well as precisely tailored simulation rides for research.

Introduction

The use of simulators is steadily increasing both in training and research. Significant progress has been achieved in complexity and realism of scenarios, mostly triggered by technological development. On the other hand, the basic structure of simulation remains mostly unchanged. The core element of a simulator is a fixed database which is overlaid by a traffic model. While the efficiency of simulator use cannot be questioned any more, the list of shortcomings is impressively long. They can be classified into three areas:

1. imperfect reality and control,
2. fixed sequencing of scenarios which cannot be altered during simulation, and
3. no objective evaluation of drivers' performance, i.e. no possibilities to detect during simulation if
    - the driver has correctly accomplished the scneario
    - the driver has made a specific mistake
    - the planned scenario did not take place because of unforeseen behaviour of the driver

Due to the fact that a given scenario can only be generated at a few locations within the database, long distances between the scenarios are inevitable. Most simulation rides consist of rare events with high workload, followed by dull intermediate elements. This low density of scenarios has the effect that the driver soon discriminates between critical and non-critical episodes. The effect is amplified by the fact that – caused by the limited number of available elements (vehicles, objects, pedestrians) – relevant elements are concentrated within a critical scenario, leading to convictions like "whenever a pedestrian moves he will soon cross the road". Therefore, simulation rides are characterized by the fact that the drivers can anticipate the occurrence of critical scenarios very easily and also can identify relevant elements within the scenario with ease. The important learning goals of readiness for unexpected events and of discrimination between safety relevant and irrelevant elements in enriched scenarios can only partially be achieved in such environments. The obvious remedy are very dense and complex scenarios and intermediate elements. Less obvious, but critical are advanced methods to control ambient and deterministic traffic to avoid interference between both. Unfortunately, most authors only specify which kind of scenario they designed but fail to report how many of these planned situations really happened within the simulation. From our experience, in complex scenarios about one third of the intended situations do not occur due to unexpected behaviour of the driver.

Training and research applications of simulation not only require the design of relevant scenarios but also active structuring of the temporal sequence of scenarios. Take for example the question whether a driver who is nearly falling asleep will be able to react properly to an oncoming vehicle at a crossroad. The situation to be presented in the driving simulation depends on the activational state of the driver which cannot be predicted in advance, but must be waited for. Depending on the driver's position in the database it can take a lot of time until a usable crossroad will appear. Additionally, at the crossroad an oncoming vehicle must be available which drives at a velocity able to produce a dangerous time-to-collision. Quite a lot of research questions are of this nature where the occurrence of a scenario is triggered by an external or internal variable.

Furthermore, conditional sequencing of scenarios would be an important progress in simulator training. Up to now, simulation rides were designed independently from the actual performance of the trainee. It is known from learning theory that training success strongly depends on the degree to which the training is tailored to individual needs. A proper concept would be a data bank of scenario classes, covering the relevant driving tasks. For each scenario class there is a set of scenarios of similar difficulty for repetition purposes as well as a set of scenarios with increasing difficulty. Depending on his or her actual performance, the driver will experience new scenarios following a training schedule based on learning principles. A respective data bank would also offer an objective assessment of driving performance resulting in an individual profile with direct hints what has to be trained.

Prerequisite for adaptive learning and testing is to assess driving performance on-line. Up to now, traditional one-to-one interaction between instructor and trainee is found in real driving as well as in simulator training. Evaluation is mostly done by subjective methods, making only in part use of the fact that the relevant variables to describe driver behaviour are available in the simulation. Objective performance indicators require effort both from the user (What is the correct behaviour the trainee has to show?) and from the developer of the simulation (How can behaviour be described in terms of objective information?).

To overcome those shortcomings, a new concept for driving simulation was developed at the University of Würzburg. As can be seen from the description above, the main reason for restricted usability of traditional simulation is the use of a fixed data base. Therefore, our simulation is based on scenario elements which can be linked together arbitrarily during runtime. Secondly, these scenario elements consist of a specification of road environment, traffic configuration and task criteria.

Previous work

The basic architecture of our simulation was presented at the DSC2001 ([2]). In this concept the road network and its surrounding landscape is created during runtime of the simulation and, in consequence, can be changed during simulation by taking advantage of the driver’s limited area of visibility. To create such a dynamic database, the user encapsulates scenarios in so called scenario modules. The definition of a scenario module contains the road network, the surrounding landscape and the behaviour of vehicles in the surrounding traffic (including pedestrians).

Scenario modules resemble so called database tiles as described in [1] and [3]. The common principle is that the user builds a complex database by combining predefined tiles. For example, a tile may contain a city block or a section of a highway. The tiles have to be translated and rotated to form a continuous road network. Usually these transformations are made offline, i.e. before the simulation. This means, that the sequencing of the tiles during simulation is fixed. In our architecture, the scenario modules are transformed online, which allows flexible sequencing of the modules during simulation.

The user combines the scenario modules following a topological rationale. The topological combination of scenario modules allows three special types of connections:

- Topological loops: Two road networks are connected in an endless loop, without regard to their geometry. In Figure 1a two scenario modules, m1 and m2, form such a loop. The road network in m1 and m2 is a simple straight road.
- Topological junctions: More than one road can be connected to another road without a junction. In Figure 1b the driver reaches m2, independent of the route he chooses at the crossroad in m1.
- Conditional connections: The connection between modules depends on conditions. In Figure 1c Module m2 is only connected to m1 if the driver is faster than 100 km/h. Otherwise m3 follows m1.

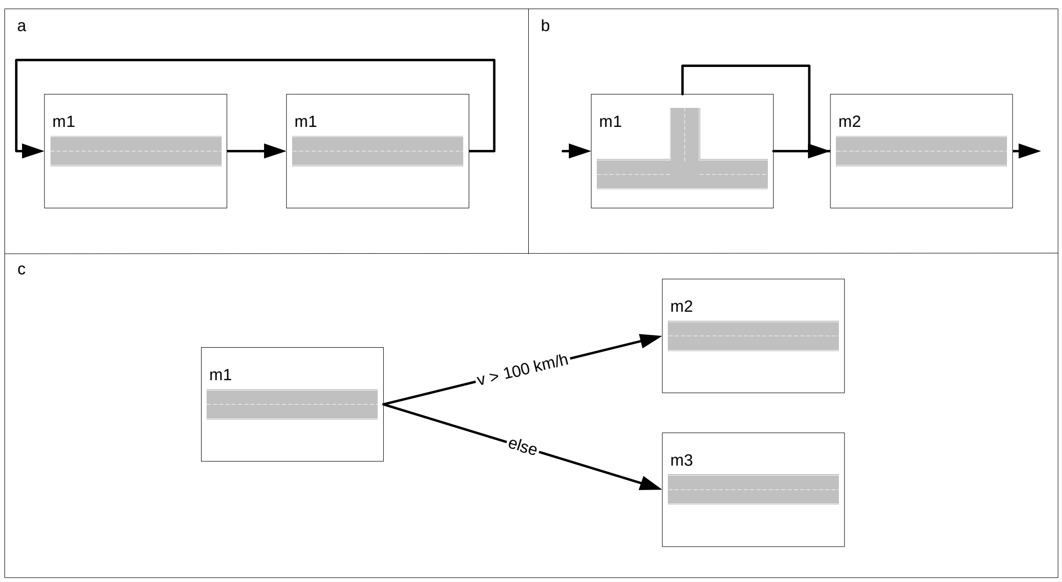

Figure 1: Types of topological connections

During runtime, the geometrical and graphical representation of the road network is computed on the base of the predefined topological representation. If the scenario element which follows the actual one is determined by the topological representation, an algorithm called 'geometric instantiation' generates the road network of this scenario module and links it to the previous one. All this must happen outside the visibility area of the driver. This procedure is appropriate for many applications but has shown some shortcomings:

1. The driver's area of visibility has a fixed radius of 2 km, which is suitable for scenarios that take place on rural roads and highways. In urban scenarios the driver has to travel long distances until the next scenario module can appear.

2. Apart from the vehicles that are relevant for a scenario usually only a few other vehicles drive around autonomously. Especially in urban scenarios this results in low traffic density.
3. The conditional connections are hard to use since they have to be defined using low-level parameters of the simulation (e.g. velocity, lateral deviation).

In most cases the user does not want to account for all possible routes the driver may take at junctions and crossways. Instead of completely defining the topological network, he is interested in defining only the sequence of scenario modules. This sequence has to appear irrespective of the route the driver has chosen.

Extensions of the initial concept

To cope with these problems the initial concept of dynamic databases was extended in several ways:

1. Scenario modules are completely defined using a graphical editor. Additionally, the areas of visibility are defined individually for each scenario module.
2. The simulation of surrounding traffic offers mechanisms to create high scene density by controlling vehicles that are not directly relevant for a scenario. These vehicles belong to the so called “ambient traffic”. Furthermore, pedestrians can walk in the scenario autonomously.
3. The behaviour of the driver in a scenario module is analysed automatically during simulation. The results of this analysis are used to control the conditional sequencing of scenario elements.
4. The user only has to define the sequence in which scenario modules must follow each other during simulation. The connections for handling ‘wrong’ turns at junctions (i.e. turns that are not intended by the scenario) are generated automatically.

Definition of scenario modules

A graphical editor was developed which allows to define all aspects a scenario module (see Figure 2).

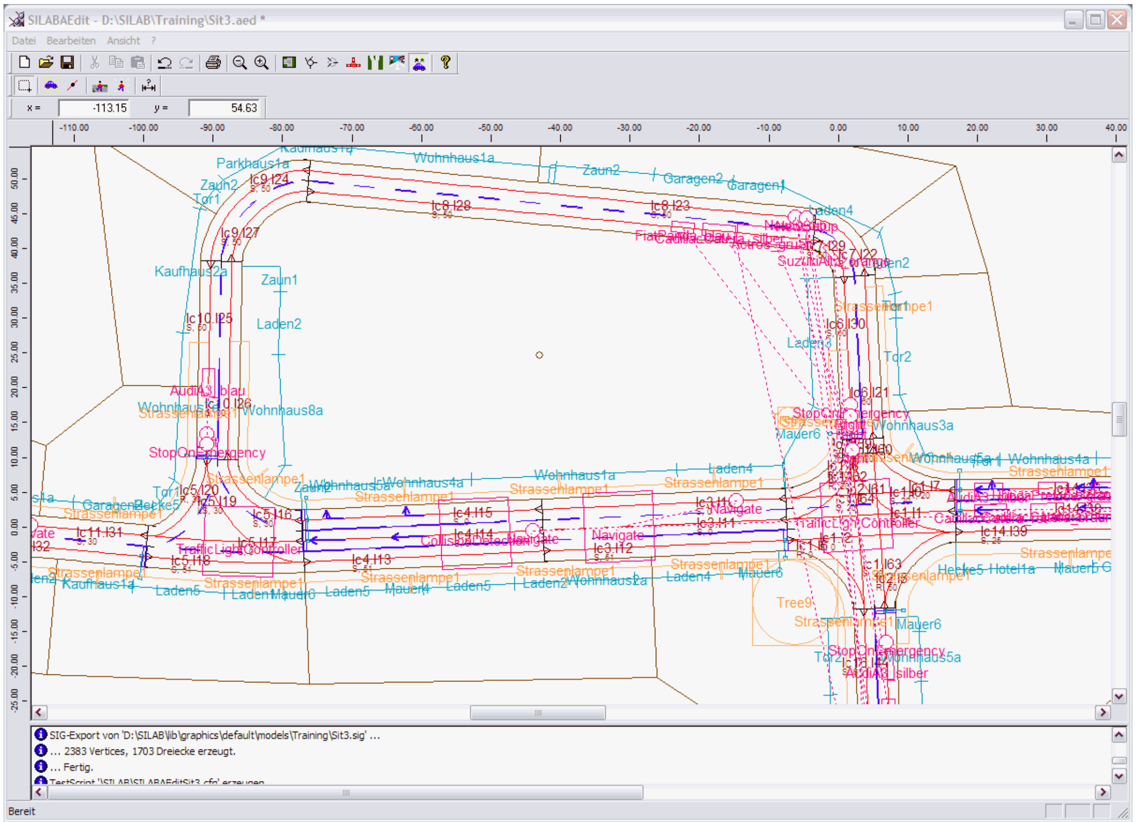

Figure 2: Graphical editor for the definition of scenario modules

The editor structures a scenario module in several layers. Each layer offers tools to edit different aspects of the module. The main layers are (see Figure 3):

- Layout: By drawing simple figures like lines and circles the user defines the overall layout of the road network.
- Cross section profiles: A cross section profile stores information about the number of lanes of a road, their width, relative height and pavement. The cross sections are placed on the layout figures).
- Lanes: The cross sections profiles are connected via lanes. Lanes are represented as parametric curves in 3D space.
- Lane markings, traffic islands, borders.
- Signs and traffic lights: These objects are placed in relation to the lanes. For example, each sign has a particular position along a lane and a certain distance to this lane.
- Graphical objects: The user can edit the graphical characteristics of a scenario by placing in several graphical objects (houses, trees, street lights etc.).
- Behaviour of traffic and pedestrians: The user defines where and under which conditions vehicles and pedestrians appear. He can control the behaviour ranging from autonomous driving/walking to precise manoeuvres.
- Areas of visibility: For each road that leaves the scenario module the user defines the area within which the end of the road can be seen by the driver. The size of these areas depends on road geometry and the graphical objects. For example, if a road ends in a sharp bend and has a row of houses at its side, the area of visibility can be kept very small (radius in urban scenarios typically < 200 m). During

simulation, the geometric instantiation connects the next scenario module as soon as the driver enters the respective area. The determination of the area of visibility is supported by the program.

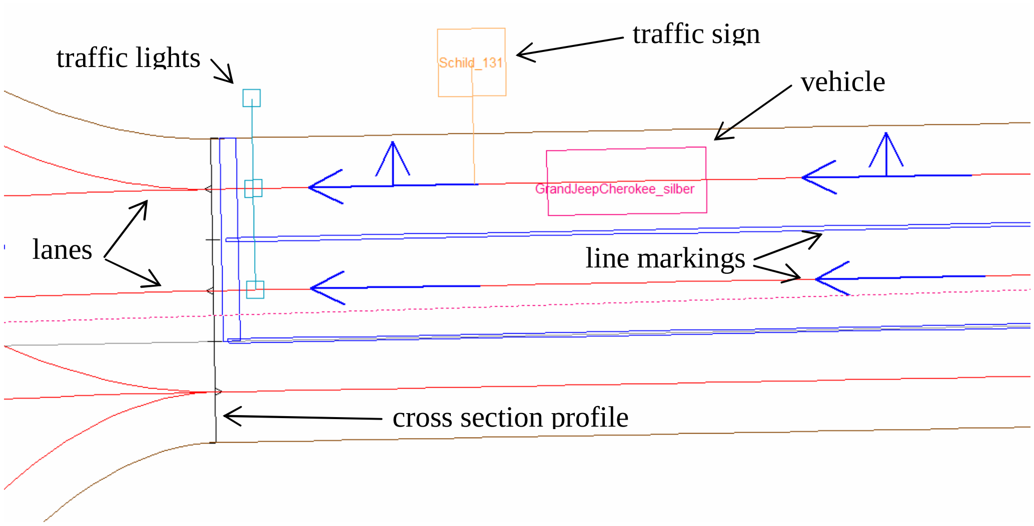

Figure 3: Elements of the editor

The road network is based on simple layout figures and their positions are defined in relation to lanes. Therefore, variants of the road network are easily created by simply adjusting the layout figures. During the development of scenarios the user can always test the scenario on his desktop computer by driving via joystick or keyboard.

Ambient traffic and pedestrians

The introduction of short visibility areas allows very dense sequencing of scenarios. In order to produce scenarios enriched with relevant and non-relevant elements, many objects – especially vehicles and pedestrians – must be included which are not directly relevant for the scenario itself. This is of particular importance in the intermediate pieces between scenarios. The difficulty is to prevent the ambient traffic from disturbing the relevant vehicles which create the actual content of the scenario.

To create vivid ambient traffic, the user defines a number of setup points for each vehicle of the ambient traffic. At each setup point a vehicle can be started. Additionally, each setup point has stored a route to a goal point. As soon as a vehicle of the ambient traffic reaches its designated goal point, it disappears and starts again at another, randomly chosen setup point. The setup points have to be defined at positions that are not visible for the driver provided that he takes the planned route through the scenario. However, if the driver gets too close to a setup or goal point, no vehicles will start or disappear. The same procedure was applied to pedestrians that are not relevant to a scenario. They use path planning algorithms to stay on the sidewalks/crosswalks in order to prevent collisions with other pedestrians, vehicles or obstacles. A screenshot of a scenario is given in Figure 4.

Figure 4: Urban scenario

Online analysis

The online analysis of a scenario is able to detect the following events:

- The driver accomplished the scenario without mistakes.
- The driver behaved in a way that the planned scenario did not occur (e.g. the driver has chosen an unplanned route at an intersection).
- The driver made one or more errors out of a set of possible mistakes.

To make online analysis of driver behaviour possible, the user has to define criteria for a successful solution of the driving task. To this end scenarios are divided into smaller units, so called "phases". For example, an overtaking manoeuvre might have the phases "approach preceding car", "wait for gap in oncoming traffic", "change to left lane", "overtake", "change to right lane" and, if oncoming traffic appears during overtaking, "cancel overtaking". The transitions between phases are controlled via parameterized rules. A rule detects basic events and manoeuvres by evaluating low level parameters of the simulation. The user chooses the relevant rules from a set of about 20 rules. Exemplary rules are:

- The driver follows a vehicle in front / stops behind a vehicle in front.
- The driver accelerates / brakes / turns the steering wheel / activates a turn signal.
- The driver has caused too high lateral/longitudinal/angular acceleration, indicating that his vehicle is out of control.
- The driver has a particular level of time to collision (TTC) to an obstacle / vehicle / pedestrian.
- The driver is close to a certain obstacle / vehicle / pedestrian.

By identifying the phases with states, the online analysis has the structure of a finite state diagram. In order to detect driving errors not depending on the phases of the scenario, the user can activate global rules. For instance, the driver should always maintain safe TTCs. In the last step the user assigns to each rule the respective output information. This can be text messages which are presented during simulation or protocols which can be used for debriefing.

Adaptive sequencing of scenario modules

Additionally, and most importantly, the result of the online analysis may be used as a trigger for the decision which scenario should follow as the next one (adaptive training). To achieve this goal, a learning structure must be implemented in the simulation which controls the selection of the next scenarios from the scenario data bank. The learning structure must be based on psychological and didactic considerations.

Conclusion

Starting from the shortcomings of traditional simulation with fixed databases, our concept of a dynamic generation of scenario sequences during runtime was substantially extended. New basic elements of the simulation are scenario modules consisting of definitions of road environment, traffic configuration and task criteria. These modules can be linked together arbitrarily. The introduction of variable areas of visibility allows for a very dense temporal sequencing of scenarios. Furthermore, new control mechanisms for the surrounding vehicles and pedestrians lead to more dense and realistic traffic in urban environments. By including performance criteria into the scenario definition, driver behaviour can be assessed online. With help of these new facilities, adaptive training tailored to the needs of the trainee is possible. Also, a potent tool for experimental research is now available.

References

[1] V. Govil, R. Mourant: A Tile/Scenario Algorithm for Real-Time 3D Environments (poster), Proc. of the 32nd International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH), Los Angeles, 2005

[2] A. Kaussner, M. Grein, H.-P. Krüger, H. Noltemeier: An architecture for driving simulator databases with generic and dynamically changing road networks, Proc. DSC 2001, Sophia-Antipolis, 2001

[3] Y. Papelis, O. Ahmad, G. Watson: Developing Scenarios To Determine Effects Of Driver Performance: Techniques For Authoring And Lessons Learned, Proc. DSC North America, Dearborn, Michigan, 2003