



Version 1.1.2/22-09-2005  
Copyright © 2005 by ST Software B.V.

All rights reserved. No part of the contents of this document may be reproduced or transmitted in any form without the written permission of ST Software



# software modules for developers of driving simulators



## CONTENTS

<b>1</b>	<b>GENERAL</b> .....	<b>2</b>
<b>2</b>	<b>STCONTROL: USERINTERFACE (GUI) MODULE</b> .....	<b>4</b>
<b>3</b>	<b>STTRAFFIC: RUNTIME TRAFFIC- AND SCENARIO GENERATION MODULE</b> .....	<b>6</b>
<b>4</b>	<b>STRENDER : GRAPHICAL RENDERING MODULE</b> .....	<b>9</b>
<b>5</b>	<b>STROADDESIGN: ROAD DESIGN TOOL</b> .....	<b>10</b>
<b>6</b>	<b>STSCENARIO SCRIPTTOOL</b> .....	<b>13</b>



## 1 General

ST software provides a number of software packages which, together, form a complete driving simulator software suite. The functionality included enables the configuration of a range of different types of driving simulators. Together with the tools to create graphical 3D databases and simulations, developers are now able to configure and create a driving simulator that matches their specific needs. With these tools, courses for driver training can be developed at low costs and high learning value. Scientific experiments on driving behaviour can be designed and run fast and efficiently. Also, special courses and demonstrations can be created with minimal time and costs. All software packages are integrated products that closely work together. Two packages are used in the design (off-line) phase, while the others are used in the runtime simulations. In a modular and scalable approach each runtime package is applied on a separate PC.

Since a number of databases and basic lessons are included with the package, developers don't start from scratch. The databases can be changed and extended according to their own needs and the lessons can be modified because all lessons are created in a script language.

### 1.1 Development desk system

This is where the design and evaluation of 3D road databases and traffic scenario scripts takes place. The system constitutes one single Microsoft Windows™ based PC running three software modules for the development and evaluation of simulation tasks. The complete development of simulation tasks may be performed from this desk. When finished the script and road network files are copied to their proper runtime PC's for the actual driving simulation

#### StRoadDesign

Graphical road databases can be created with the *StRoadDesign* software package. This designer allows developers of simulators to build their own databases or change existing databases. *StRoadDesign* generates the geometrically correlated

graphical and 'logical' databases required for respectively the intelligent traffic module in *StTraffic* and the graphical rendering module *StRender*.

#### StScenario

The simulations are generated with the scripting language *StScenario* for which a handy script tool is offered. This scripting language is very versatile, and easy to learn for people with traffic-analytical insights and a little (software-) technical background. The best way is to start with the lessons that are included with the package or take a training course from ST software in simulation development. After that, one can add functionality to the lessons, change them or create new lessons. At runtime scripts are executed in *StTraffic* in a binary format resulting in fast simulations with high frame rates, even when a lot of intelligent agents (vehicles or bicyclists) are simulated simultaneously. Scripts basically control the simulation and are the prime and rich interface to the functional properties of the core software. Script setup and control the simulation and can e.g. even be used to communicate with external computers for special tasks.

### 1.2 Runtime simulation system

The *runtime simulation* system constitutes the main driving simulator executive environment. Each individual driving simulator consists of a number of interconnected PC systems. There is one GUI-based operator console running *StControl*, one traffic simulation module running *StTraffic* and one or more graphical rendering units running *StRender*, the number depending on the required graphical environment. Generally three rendering systems may cover 180-215 degrees horizontal visual field of view. Sound is generated by the *StControl*-PC on-board sound system.

#### StControl

The graphical user interface *StControl* allows control over the simulation module *StTraffic* and can be configured to govern as many renderer modules as needed. It allows



the start-up of the simulator, controls its state (pause, run, shutdown) and selection of scenarios. At the heart of the controller function is the network based communication protocol *stCommlib* which connects the simulation modules.

### StTraffic

This module is the heart of the simulator and computes the dynamic traffic system based on intelligent agents based technology. Each simulator has one *StTraffic* PC which reads and interprets the scenario scripts, reads the driver cabin status, computes the internal engine model and the traffic system and transmits the resulting status of the simulated vehicles to the renderers. The internal vehicle model contains an engine model, clutch and a 3DOF cabin model based on Newton forces and a 'magic formulae' based road contact computation. When required it is also possible to use a different external vehicle model instead.

Users may add their own vehicle cabin to the *StTraffic* module or drive the simulator with a game set connected to *StControl*. This allows for full scalability: the simulators that are build with the software can range from low end to high end..

### StRender

The rendering package, *StRender*, is a configurable real-time 3D graphical rendering application which shows the graphical environment including simulated traffic in real-time generally at a smooth 60 cps frame rate. Configuration parameters like viewing angles, viewing offsets, viewports and mirrors can be defined by the user. External viewing monitors can be added. It is easy to add as many renderers in one simulator as required. Systems may then range from simple 1-channel simulators with only the road view ahead on a single

monitor, to a full 360° field of view with multiple channels and projection screens.

*StControl* runs on the MS windows operating system (windows 2000 and XP). *StTraffic* and *StRender* run on both the MS Windows 2000, Windows XP and linux operating systems. At present, real-time rendering performance is best on the linux platform. So, for high performance rendering, linux is recommended at present for *StRender* as well as *StTraffic*.

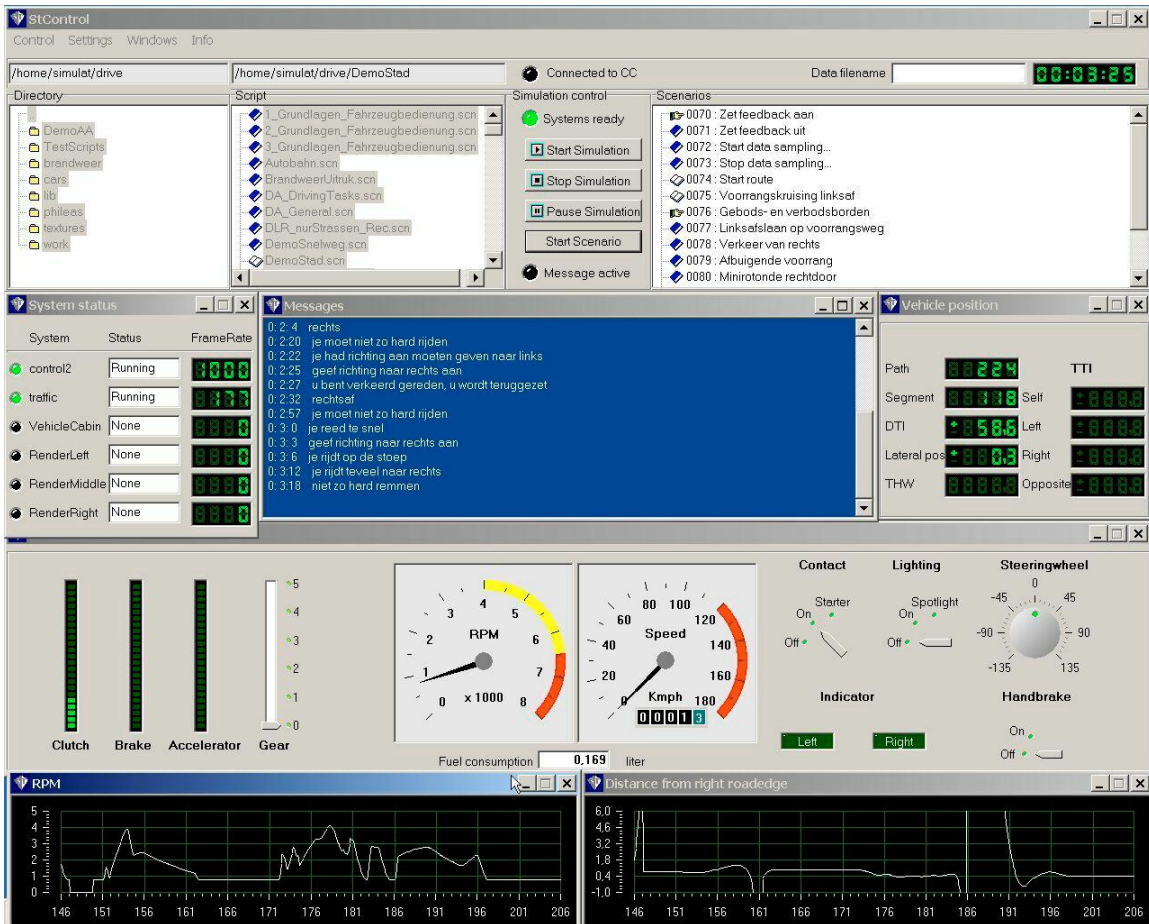
With these software packages, developers are then able to create both fully functional driving simulators as well as all relevant content, like databases and simulations, courses and experiments, for their driving simulators. In summary the following modules are provided by ST Software:

Package	Phase	Description
<i>StRoad-Design:</i>	design	Road network and environment database design tool
<i>StScenario + scripttool:</i>	design	a text editor tool for creating scenario scripts and for syntax checking
<i>StControl:</i>	runtime	user interface (GUI) module to control and configure the simulator system
<i>StTraffic:</i>	runtime	Runtime core traffic- and scenario processing module
<i>StRender:</i>	runtime	3D graphical rendering module

In the following chapters these system components will be described in more detail.



## 2 StControl: Userinterface (GUI) module



*StControl* is a MS Windows™ based graphical user interface (GUI) for controlling a single driving simulator system. *StControl* overviews a single simulator system with all computers that are part of the driving simulator. It serves two primary functions:

- 1) Configuration of the driving simulator system and control of all communication between the components.
- 2) Control of the simulator, the lessons (simulations) and scenarios, together with providing feedback to the operator of the system.

The GUI gives a broad overview of the functionality of the simulator software. *StControl* includes the following functions:

- It controls all Ethernet communication between the different computers in the local network of a single driving

simulator system. The subsystems may be remotely started, stopped, paused or shut down from the GUI.

- It provides an interface to the operator to manage all supervisor functions of the simulator.
- The directory panel enables selection of the folders on the traffic PC, via an FTP protocol, containing the scenario script files (driving lessons or experiments). The scenario script files contain the full description of the experiment or simulation which may consist of many discrete scenarios. After the user has selected a script, the simulation can be started by pushing the “Start Simulation” button. A list of all selectable scenarios is displayed in the script panel which also shows their status.
- It shows all speech messages that are provided to the trainee or subject who



participates in an experiment in a verbal text format, together with information on the time the message was delivered. The operator then knows which messages have been delivered.

- It shows all kind of information on the vehicle cabin, such as vehicle speed, gear position, clutch position and so on. Also, the distance driven and fuel consumption in the present simulation is shown. In addition, the user can open up to 6 separate panels that show analogue time histories of one minute for a whole range of data that can be selected by the operator. Data that can be selected includes lane position, vehicle speed, engine rpm etc.
- It offers the user the possibility to drive the simulator with a gaming steering set. The interface is standard configured for

a range of force-feedback and non-forcefeedback game sets such as the Logitech Wingman. This facilitates testing and debugging of simulations.

- It contains all facilities for generation of engine sound, wind- and tire/road contact sounds, starter engine, indicator sounds and pre-recorded human voice messages.

The user configures the system with a resource file. This describes the computers and functions that are used in the simulator system. For example, the user can add as many renderers as desired, or add a computer to the system that controls a moving base system.



### 3 StTraffic: Runtime traffic- and scenario generation module



This module drives the real-time simulation. It communicates via TCP/IP and UDP protocols with 1..n rendering computers (*StRender*), a vehicle cabin, and *StControl*. The program starts up in the required mode with a number of command line parameters and when running it can be controlled by an operator using *StControl*. It loads a scenario script file, generally by a GUI command, which plays a central role in the simulation process. Because this script language has a very rich functionality, it is documented in a separate document.

#### Script processing

After a scenario script has been selected a number of things are initialized, for example:

- The script file contains the name of a road database to be used for this simulation. The road database was created by *StRoadDesign* (see later). Both the logical road database and graphical road database are identified now:
  - the **logical road database** is used by the traffic simulation system and gives a mathematical description of the road network geometry by which all traffic participant are able to navigate through the network of roads

- The **graphical road database** is a 3D scenegraph to be used for real time visualization by the renderer PC's.

- All scenario statements are parsed and converted to a binary representation that results in fast real-time performance at runtime. Within a scenario script file, other scenario script files may be included. By this, existing functionality may be re-used in other simulations.
- All kinds of other initializations are done, for example the simulator vehicle, data storage, data communication and so on. All connected systems report 'ready' to *StControl* when done.

During program execution, all simulation functions will be executed in a continuous loop at a high rate. In the loop, all scenarios are evaluated sequentially. Then all traffic participants are instructed to perceive their surroundings, to decide what to do and to make a movement. Then the movements are converted to a position in space, a heading, roll and pitch of the vehicle. And finally the positions of all traffic participants are forwarded to the rendering PC's via an Ethernet protocol. In the meantime all required communication to the other computers in the local network is handled.

#### Multiple scenarios

In the scenario script file, the user is free to specify any number of scenarios. Each scenario is indicated by a unique number. A



scenario starts if a specified condition becomes true, and it stops when another condition becomes true. Also, scenarios can be assigned to specific agents (vehicles or bicyclists) who then perform their own specific tasks.

By this mechanism, literally hundreds of scenarios can be active at the same time. Each scenario then controls its own activation.

### **Traffic behaviour**

Anything within in the simulation task can be controlled by scenarios, the creation of traffic participants included. Traffic is created on the fly by the scenarios. But once the traffic participants exist, they govern their own behaviour, unless the scenario scripts override their default properties. This important principle creates a natural interaction between traffic participants, which can nevertheless be controlled by changing the parameters via the scenario script. All traffic participants are responsible for their own termination. A number of different types of traffic participants are available. There are a number of different types of vehicles (cars) and bicyclists. Pedestrians will be available in the near future.



All traffic participants move in the logical network of roads, and they can be perceived and acted upon by all other traffic participants. They are autonomous agents with their own set of behavioural rules which they activate depending on and as required by the momentary situation. This depends on the type of road, the presence of road

signs and markings, the presence of other traffic participants, and so on. Behavioural parameters and sensors can be changed in real time via scenario scripts. Each vehicle is able to:

- follow other vehicles safely
- overtake other traffic participants
- give right of way at intersections
- choose the correct lane of highways and other multiple lane situations
- choose the correct speed depending on type of road and road signs
- evaluate and stop for traffic lights, stop signs etc.
- let other vehicles merge into the traffic stream at highways
- follow their route autonomously
- etc.

### **Left and right-hand traffic**

The simulator user can choose between a left-hand driving system, as in England, and a right-hand continental driving system.

### **MultiCabin simulation**

Each system handles only one human driver. However, A very advanced feature of the software is that simulator systems can be connected so that all traffic is shared between different simulators. This mechanism allows that a number of human drivers can drive on different systems, yet are integrated in 'one world' in which they see and respond to each other and to all other traffic that is generated on these systems.

### **Script features**

Traffic can be modeled depending on the actions of each human driver. All human drivers are 'visible' by the autonomous agents and are fully interactive. A large number of other things can be handled in script in real time. For example, sending a speech message string to *StControl*. The script language ensures that the simulation is an open system in which all variables are accessible to the user: driver performance can be monitored online and performance feedback and task- and error analyses can be made during driving. The results can then be used to give real-time feedback to the driver and to send data to *StControl* for printing a report. A number of scenario scripts containing all this functionality are



included in the package for use in driver training.

### **Data sampling**

As another example, driving data can be sampled in a continuous stream of data samples or by storing event related signals in a database. The user may want to start data sampling to a binary file or to an ASCII text file. Data stored in binary files contains the 'raw' data that can be visualized and processed with the offline data analysis and visualization software. The user can choose from a wide range of variables to store. In addition, the user can do his own computations in script by means of a set of user defined variables, which can be sampled and stored as well. Alternatively, the user may write scripts that process data during the simulation (for example compute minimum time-headways, average speed, standard deviation of lateral position and so on) and store these to text files.

### **External devices**

Another powerful feature of the script language is communication and interfacing with external computers. You can open as many UDP data communication ports as you want, and read from or write to these ports from within the script code. The user can then write an application on another computer and send data, according to a protocol that is defined in script, to that other computer. In that way the user can, for example, store or visualize data on another PC, or present messages via a stand-alone support system on another PC or control an external device such as an AICC. Or program a computer to register and store button presses that are associated in time with events that are detected and transmitted by the simulation software. Also the user can receive data from another computer according to a protocol that is defined in script.





## 4 StRender : Graphical rendering module



*StRender* is the ST Software runtime rendering module that enables smooth real-time visualization of the simulated traffic environment in a driving simulator. It has a scalable design that allows rendering to one or more connected graphical displays by using several rendering units. In a configuration file the viewing and projection properties are defined for each window and its associated viewports. Each defined window in the configuration file will open a window on its associated PC system and allow definition of one or more out-window and mirror viewports per window. *StRender* is designed for integral use with the *StTraffic* simulation software and *StControl* simulation control unit. It receives real-time simulated traffic data from *StTraffic* and displays the cars and other dynamic data like traffic light states on the screen(s). Time extrapolation techniques result in a smooth movement of traffic participants despite asynchronicity between the traffic and rendering modules. State-of-the art rendering technology, based on the OpenSceneGraph rendering toolkit, enables real-time frame rates on standard PC hardware, e.g. demonstrated 60 fps for average complex city databases including 20 or more simulated cars. Of course frame rate performances depend strongly on scene complexity and effective database organization.

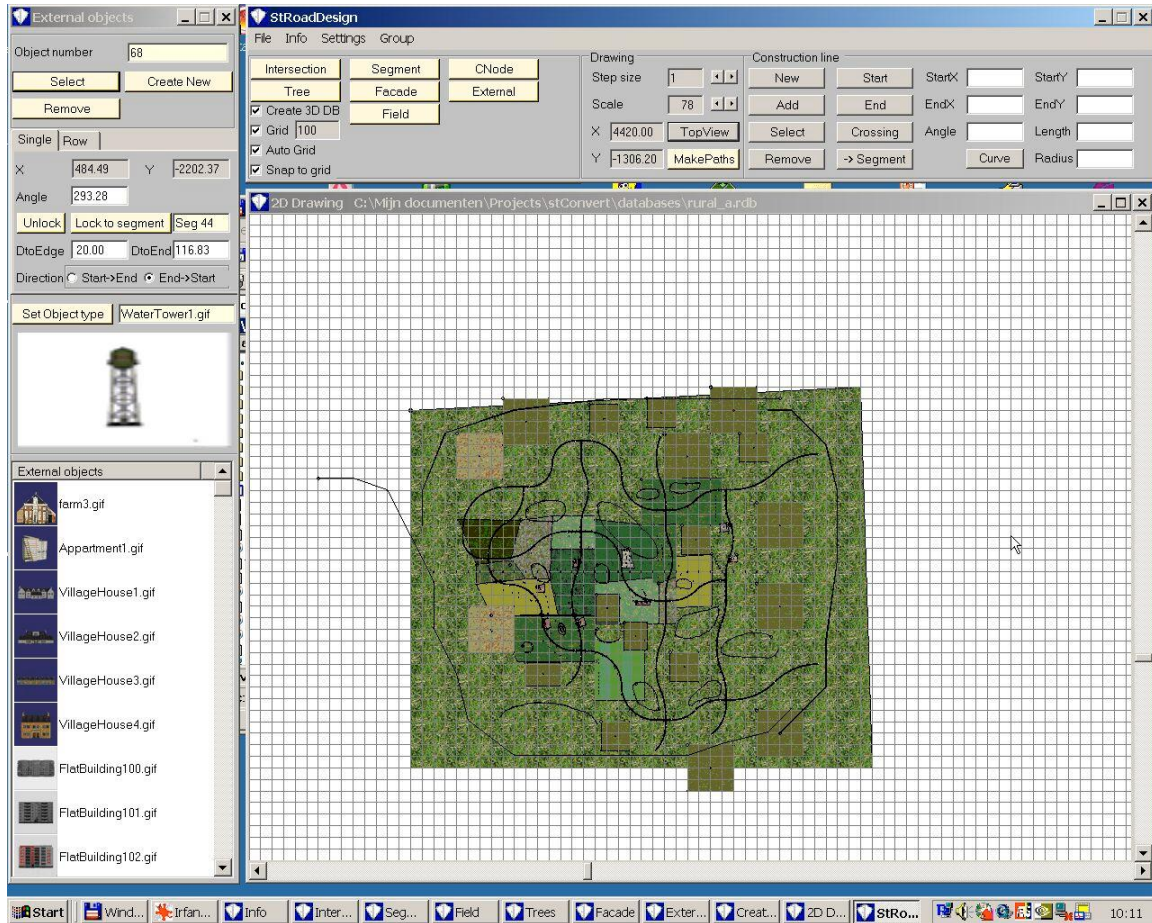
Some key features of ST-Render:

- Runs on MS Windows™ and Linux (e.g. Suse 9.3) environments.

- Runs on a standard off-the-shelf PC system, e.g. Pentium4 3.2 Ghz CPU and Nvidia™ FX6800 graphics card.
- Allows a scalable display configuration by integrating one or more display screens side-by-side, each driven by its proprietary rendering PC. This modular approach enables building any driving simulator that ranges from single screen low-end to a multi-view high-end system.
- Allows definition of separate additional viewports on the display screens, e.g. for mirror views or car-following monitor top-views.
- Accurate positioning and smooth movements of traffic participants by time-extrapolating methods.
- Use of open source OpenScene Graph 3D graphics libraries for core rendering tasks. These enable the highest performance rendering available today and the use of multi-format databases.
- UDP and TCP Ethernet protocols for data communication with the *StTraffic* simulation module and system control by the ST-Control module.
- Rendering of graphical databases generated by *StRoadDesign* and-or 3<sup>rd</sup> party database modelers, e.g. Multigen™.
- Includes standard a number of different real time multi-LOD car models each in different colors.
- Rendering of animated bicyclists and pedestrians (autumn 2005).
- Renderers are configured by means of a resource file, which specifies geometry of windows, view ports, mirrors, view angle, etc. to setup the renderers.
- Functionality is included to present bitmap images (instruction or dual tasks) via the renderers, controlled, like any other simulation feature, by scenario scripts.



## 5 StRoadDesign: Road design tool



*StRoadDesign* is a designer for creating graphical databases and logical databases for the intelligent interactive traffic of *St-Traffic* to drive in. A logical database describes all relevant road characteristics in mathematical terms for the intelligent traffic, and the human simulator driver, to drive in. Because of the facilities the designer offers, developers are able to build their own 3D databases, with all road geometry, road signs and markings that are required, including fully textured environments populated with trees, houses, road furniture and so on. This gives the user a high level of flexibility. To mention a few features of this package:

- All graphical databases that are exported are in the industry standard OpenFlight™ format. Completely textured databases with many 3D models can easily be created with *StRoadDesign*.
- The flight databases are automatically converted into the efficient '.ive'-format. This is a binary OpenSceneGraph format that results in easy management and very fast database loading time.
- All graphical databases are in multi Level-Of-Detail (LOD), resulting in very fast real-time performance. LOD for road segments is one of the features that can be configured by a resource file.
- All external models (like buildings, street furniture etc.) can be optimized individually by adding a distance at which they are rendered. This mechanism allows including a large number of external models to the database while still keeping a high drawing frame-rate in *StRender*.
- An important part of the design philosophy is that external models are either created by the user or are



purchased via third parties (like [www.turbosquid.com](http://www.turbosquid.com) that offers a large number of cheap low polygon models). The best way to operate is to buy 3D studio models and convert them to OpenFlight with a conversion program like PolyTrans. To modify the external models 3D studio is recommended.

- The design of road-geometry is highly intuitive and user-friendly. You make a 2D design of a network of roads and intersections. The 2D design is then converted into a 3D database automatically. To our knowledge there is no other designer or modeler on the market that allows creating complex road geometry databases so easy and fast.
- Because graphical databases and logical databases are created from the same source, a geometrical match is guaranteed. The intelligent interactive traffic agents in *StTraffic* are able to drive autonomously in these databases, and this is displayed graphically with *StRender* which uses the correlated graphical database. This approach guarantees seamless operation of all the databases created with *StRoadDesign* in *StTraffic* and *StRender*.
- All kinds of roads and intersections can be designed, ranging from simple one-lane roads to complex multiple-lane roads. Different kinds of lanes are supported, such as exit lanes, bicycle lanes, pavements for pedestrians, complex lanes for weaving on highways etc. Complex multiple intersections can be build easily. Also, roads may split in more roads, and multiple roads may merge together into a single road. Because of these facilities all kinds of road-structures can be designed, such as mini-roundabouts, large roundabouts, multi-lane highways with complicated exits and entrances and so on.
- All kinds of road markings and road signs can be added. Since road signs are defined in a resource file, it is easy to convert all road signs in a database to signs that are used in another country. The user may add specific road signs to the designer via a resource file or change the characteristics of existing road signs. Internationalization of databases is made easy by this.
- Different kinds of traffic lights can be added to intersections. The control strategy of these traffic lights can be completely adjusted. Also, setting the control strategy of traffic lights on complex multiple intersections is easy. Traffic lights are controlled at runtime either automatically or by scenario scripts.
- Complex road structures can be selected as a 'group' and saved to file for re-use, or copy and pasted to another position in the road network. Also, these groups may be rotated as a whole and moved to another position, and a group may be connected to another part of the road network. Because of these features it is easy to design and save a specific road structure once and then import the group as a whole into a road network as often as you wish. A group may consist of any number of roads, intersections, structures where lanes split or merge, external models, trees, landscape polygons or whatever, including all specific traffic lights, signs and markings. *StRoadDesign* comes with a whole range of pre-defined road structures like roundabouts of different types and complex intersections that you can import immediately in your own databases.
- A map or hand-drawn sketch may be scanned and imported and scaled to the appropriate dimensions in the designer as a background on which to model a network of roads.
- The database can be dressed up with any number of landscape elements (textured planes), trees and bushes, vertical panels with tree lines and other items to cover the horizon, and any number of houses and other buildings, road furniture like street lamps and so on. All these objects and their textures are defined by the user in resource files, to which the user can add specific textures and external object. Because of this, it is very easy to configure *StRoadDesign* according to the users' specific requirements, and design a typically British database, or a Chinese database, or a Dutch database.
- If the user wants to add their own (or third party) textures (for trees, facades,



grounds etc), an adequate photo processing program is recommended to modify the textures. *StRoadDesign* requires texture bitmaps in the '.rgb' format. Also, there are some requirements on the textures' pixel dimensions, and for some purposes, a transparency alpha channel must be added. For '.jpg'-to-'.rgb' conversion and other texture modifications Adobe's Photoshop is the recommend tool.

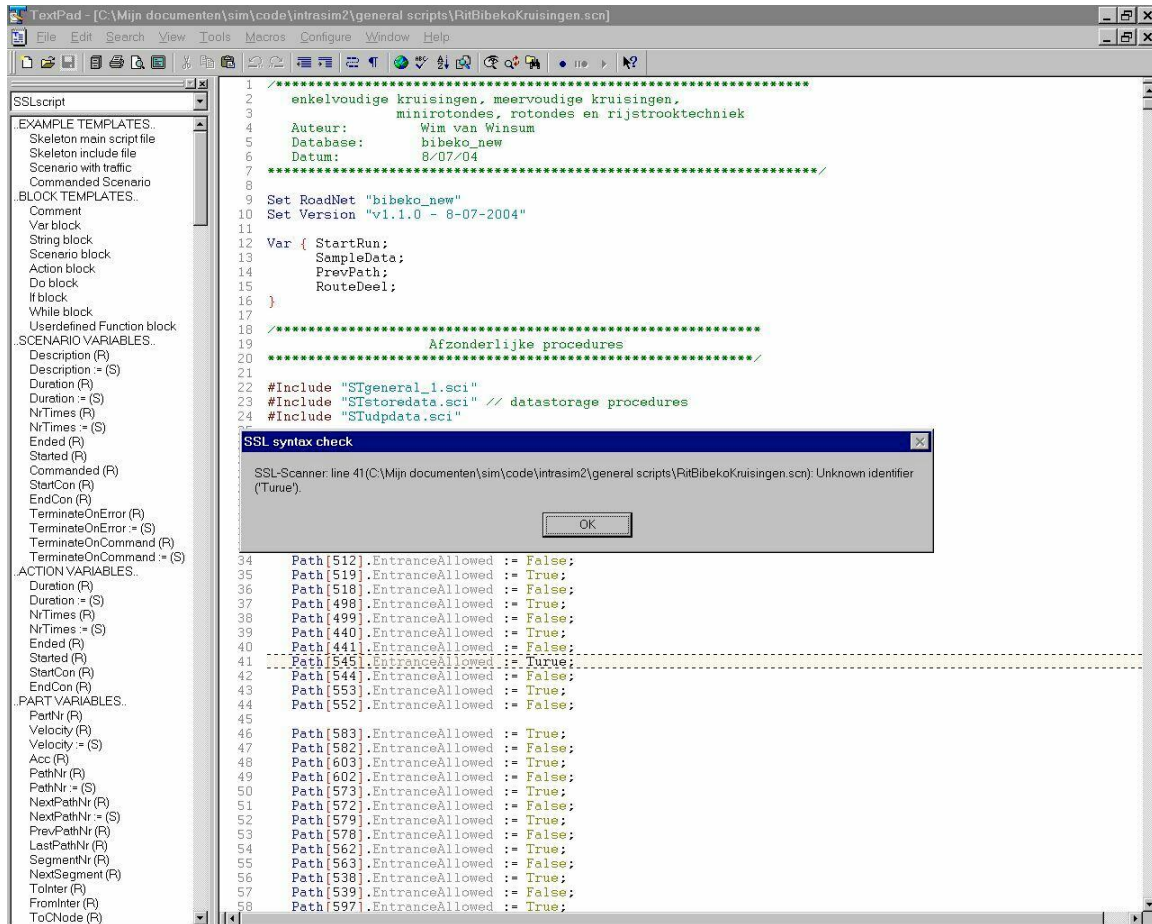
- *StRoadDesign* comes with a set of databases and resources like textures, external objects and groups of road structures to give the user a quick start.

In the present version of *StRoadDesign* only flat roads can be constructed. The height dimension will be added as a future development. As already indicated in the text, for professional development of databases the user requires other software packages in addition to *StRoadDesign* of which the following are recommended:

- 3D studio <http://www.autodesk.com>
- Polytrans <http://www.okino.com>
- Adobe Photoshop <http://www.adobe.com>



## 6 StScenario Scripttool



*StScenario* is the scripting language that was developed for real-time simulation by ST Software. The scripting language is very versatile but requires some skills of programming and abstract thinking to learn. The scripts are written in plain ASCII and parsed by *StTraffic* into efficient binary trees that are interpreted in real-time. This results in very high simulation frame rates in which a high number of scenarios can be simultaneously operating in real time.

ST Software provides a tool to help in writing scenario scripts. The script tool uses the commercially available third-party text editor 'TextPad' to create scenario script files. TextPad can be ordered from [www.textpad.com](http://www.textpad.com). From this editor a program is called to perform syntax checks on the scenario scripts. The tool includes all script language functionality that can be

used by the developer of simulations. The user may click on complete templates by which code for new scenarios of specific types is generated, or the user may search for specific system functions or keywords and generate code that uses these by a simple mouse click. Another important feature is that all reserved keywords of specific categories are highlighted in different colors. This helps to reduce the number of errors in scenario creation. The syntactical correctness can be checked by a simple mouse click. If there are any errors, the user is guided to the script file that contains the error and the cursor is directed at the line that contains the error. The type of error is indicated too, and the user may fix it and try again. The script language offers the user all functionality needed. Here are some important features:



- The user defines a series of scenarios. A scenario is a predefined list of situations with a start and an end condition. The scenario becomes active if a logical start condition evaluates to true. The scenario terminates itself when a logical end condition becomes true. In the meantime, the scenario is active and all kinds of things can be done.
- The user can define any number of scenarios. Each scenario activates automatically when required and each scenario terminates automatically when required. Any number of scenarios can be active at the same time, which may result in very complex simulations in which all complexity is handled by the specific scenarios.
- Scenarios may be re-used by including them into the main script file. Because of that, the same functionality can be used again and again in different simulations.
- All traffic is created by scenarios. Traffic participants are autonomous agents that are vehicles of various types, bicyclists and in the near future also pedestrians. The initial positioning and removal of these agents is controlled by the scenarios. Also special properties, like the routes of these agents, are controlled by scenarios. The main principle is that as long as you don't define anything for these agents, they behave natural, respect all rules and navigate through the database by themselves. However, their behaviour can be controlled completely if this is required by the user by changing the variables of individual agents, or assigning the agents to specific scenarios that control their behaviour. All intelligent agents are able to evaluate the situation they are in, and choose the appropriate behavioural rules. Because each agent handles its own task, complexity is greatly reduced for the user.
- The user can choose between a left-hand driving system, as in England, and a right-hand continental driving system by simply calling a system procedure once.
- The script language gives access to all relevant driving simulator functionality via system functions and system procedures. This gives the user full real-time control in which the present state can always be checked dynamically.
- In contrast to various other driving simulators, simulations are never static, because all conditions are tested dynamically. This principle allows for example, that something happens if the simulator driver is three seconds removed from a specific intersection, or any intersection of a specific type. This is very important because it ensures that simulations are reproducible and comparable between different trainees, whether the trainee drives slow or fast.
- Feedback to the simulator driver can be done easily from script. Also, a number of script files are supplied with *StScenario* that monitor all driver behaviour, apply driver error categorization and send feedback to *StControl*. *StControl* handles speech messages and other sounds by playing the appropriate .wav files.
- Binary data sampling and creating both binary and ASCII files can be done from script. All kinds of scientific data like time headway, time to collision and lateral position can be sampled and stored in files for off-line data analysis and visualization. In addition, data processing can also be done in real time during the simulations.
- The user can define UDP Ethernet ports from within scripts and send data to other computers via these ports. Also, data from other computers can be read via user-defined UDP ports. By this the user can program applications that communicate with the simulator scripts and even control the simulator by applications that are developed by the user. This principle makes sure that the system is open to the user.
- *StScenario* is supplied with a number of script files for driving through databases and for generating traffic, and examples of a number of applications. The scripting language is fully documented, and the user documentation is included.



Modules driving simulator software for developers