Concept for a Task–Specific Reconfigurable Driving Simulator

Bassem Hassan
Heinz Nixdorf Institute
University of Paderborn
33102 Paderborn, Germany
Bassem.Hassan@hni.upb.de

Jürgen Gausemeier
Heinz Nixdorf Institute
University of Paderborn
33102 Paderborn, Germany
Juergsen.Gausemeier@hni.uni-paderborn.de

***Abstract* - Driving simulators support the development process of new vehicle systems, such as Advanced Driver Assistance Systems (ADAS). Driving simulators are varying in their structural complexity, fidelity and cost. Furthermore, they consist of numerous simulation models that cooperate at runtime. These partial models represent dedicated aspects of the vehicle under test, other traffic participants as well as the environment. Since the development of a driving simulator is costly and complex task, testing and training of ADAS often requires more than one driving simulator. There is a need for a reconfigurable driving simulator that allows the operator to reconfigure the simulator and its structure or exchange the simulation models in a simple way without in depth know how of the system structure or interface topology. This paper describes a concept for a task-specific reconfigurable driving simulator for testing ADAS. A systematic for the development process is presented, the key software and hardware components of the driving simulator are identified, and a configuration mechanism and its software are briefly presented.**

Keywords - Advanced Driver Assistance Aystems (ADAS); reconfigurable driving simulator; confiuration mechanis; solution elements

I. INTRODUCTION

The influence of modern Advanced Driver Assistance Systems (ADAS) on the vehicle gets more and more complex making it increasingly difficult to understand and hence analyze the interplay between ADAS, vehicle, vehicle environment and driver. Driving simulators have played a vivid part in developing new automobiles and their subsystems, providing reproducible testing conditions and a safe testing environment, as well as a means to reduce development time and cost.

Nevertheless, driving simulators are typically designed and built for a special purpose in order to support a specific analysis task in a predefined environment. Adapting such systems to support new functions or applications is very complex, time consuming, and often not feasible. Thus, with the increasing role of ADAS, there is a need for highly adaptable and reconfigurable systems that can be conveniently tailored to new specific functions.

Such a reconfigurable driving simulator closely resembles a building block concept. The various simulation models, software and hardware components that constitute the driving simulator need to be widely combinable. Moreover, there are also different levels of details simulation models, ranging from simple low-fidelity models to their respective complex high-end models, which provide a detailed simulation. Also the hardware components range from simple to complex components, which constitute different simulator setups that are capable of simulating specific aspects of the ADAS under test [1].

A reconfigurable driving simulator requires compatible software and hardware interfaces and a reliable checking mechanism to achieve consistent configurations of the system [1].

The second section of this paper, will describe the state of the art and the related work. The third section will describe the reconfigurable driving simulator concept. The fourth section will be a conclusion and the future work.

This work is part of the project TRAFFIS (German acronym for Test and Training Environment for Advanced Driver Assistance Systems).

II. RELATED WORK

Modern driving simulators show a broad range of applications, e.g., driver and safety training, vehicle evaluation, road design, and vehicle dynamics simulation. They can be used for research purposes to study the behavior of the driver, develop and evaluate new vehicle subsystems such as ADAS, driver training and many more. Furthermore, driving simulators range from expensive from low-cost desktop systems, which are even used in small companies, to high-end systems, which can realistically affect an entire vehicle mounted on a motion platform. Due to the big variation of the driving simulators complexity, fidelity, cost and the purpose of use; the conception, specification and selection process of a driving simulator is a hard process [2-4]. Typically the driving simulator user is not an expert in driving simulator techniques and can't decide easily which driving simulator match with his requirements. Methods for selecting a driving simulator have been developed, for example, Negele [5] proposed a method that allows the developer of new vehicle technologies to formulate and customize a driving simulator concept to match his requirements, in form of advice which subsystem has to be used in which application scenario.

Here are also three examples of previous attempts towards building a reconfigurable driving simulator:

 ISBN: 978-1-61208-308-7

A. *UCF (University of Central Florida) Driving Simulator*

The UCF driving simulator housed in the Center for Advanced Transportation Systems Simulation; is a driving simulator with a high driving fidelity and immense virtual environments. The UCF driving simulator provides a research platform for multi-disciplinary investigations in the traffic engineering area [6].

The UCF simulator was designed, that the vehicle simulation model and the vehicle mock-up are interchangeable [7].

B. *University of Valencia Training Simulator*

The Valencia training simulator is a cranes driving simulator for training of new workers and operators of port areas that are responsible for loading and unloading ships [8].

The Valencia driving simulator was designed, that the crane simulation model and the projection system are interchangeable [8].

C. *BVG Trams Driving Simulator*

The "Berliner Verkehrsbetriebe" (BVG) is the widest tram network in Germany [9], the BVG simulator providing a training platform for the trams driver.

The BVG simulator was designed so that the tram mock-up is interchangeable.

There were three examples of simulators, which are giving some configurability or flexibility by exchanging one or more components of driving simulators, these examples shown that till date there is no method or approach, which allow the user to reconfigure whole the system without in depth know how of the system structure and components.

III. RECONFIGURABLE DRIVING SIMULATOR CONCEPT

Towards a reconfigurable driving simulator, there is a need for a development systematic. It describes how to build up a reconfigurable simulation system. The main concept of the development systematic is described in form of phases/milestones diagram as shown in Fig. 1. It is divided into the following five phases/milestones:

A. System specification
In this phase the driving simulator will be specified by using specification technique. The specification technique itself will be shortly described and the important specification results e.g. the application scenarios will be presented.

B. Solution elements deployment
In this phase the existing solution elements e.g. different variants of the vehicle mock-up will be consolidate into the reconfigurable system. The concept of the solution elements deployment will be presented.

C. Configuration mechanism development
In this phase a configuration mechanism will be developed. This ensures the consistency and compatibility of the selected solution elements from the above step, to build a valid configuration setup of the driving simulator.

D. Configuration tool development
In this phase a user friendly interface will be designed and developed, to allow the user easily to select the suitable solution elements and to generate a driving simulator configuration.

E. Simulation initialization
In this phase the interface between the generated configuration and the simulation software will be developed.

On the following part, each phase/milestone, as well as the results of each phase/milestone, will be described.

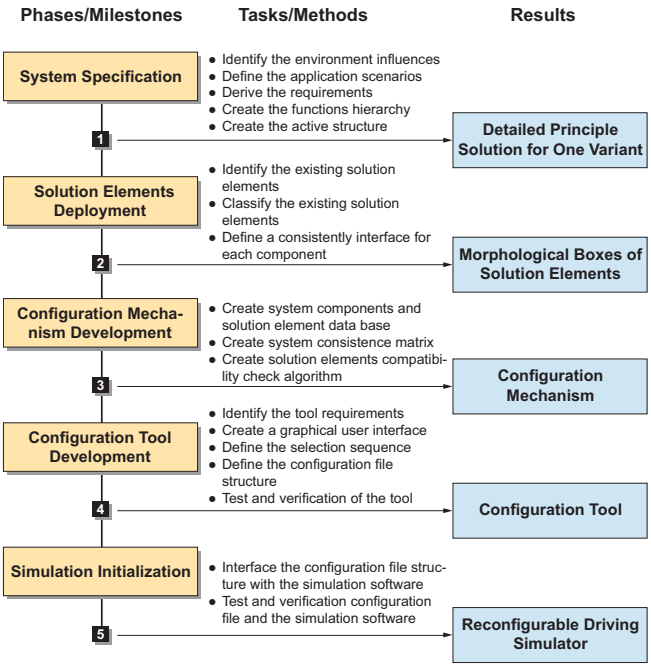

Figure 1. Reconfigurable driving simulator development systematic

A. *System Specification*

Driving simulators are complex multidisciplinary systems, which consist of mechanical components; such as the motion platform, electronics components e.g. microcontrollers, control components e.g. motion cueing algorithms, and Information technology components; such as various software and simulation models. Accordingly, a driving simulator with a motion platform is considered as a typical mechatronic system.

In order to describe a mechatronic system in early design phases, which is called "Principle Solution", the Conceptual Design Specification Technique for the Engineering of Complex Systems "CONSENS" is used [10], which was developed at the University of Paderborn.

As shown in Fig. 2, the developed specification technique divides the description of a principle solution into

different partial models according to specific aspects as following: requirements, environment, application scenarios, functions, active structure, shape and behavior [10].

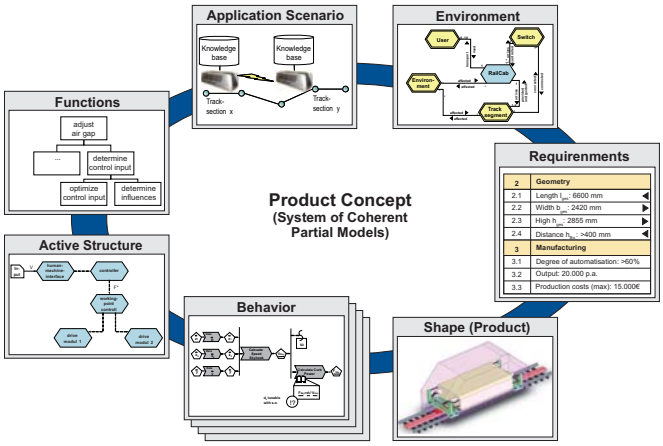

Figure 2. Specification of coherent partial models according to CONSENS [10]

We used CONSENS to develop the driving simulator principle solutions; this modeling process was done during many workshops with the driving simulator developers and users, who are involved in the TRAFFIS project. The important modeling results will described in the next part.

Environment: Investigation of the driving simulator environment shows that the external influences come from: driver, driving simulator operator, driving instructor/test manger, energy source and the ground.

Application scenarios: As described in the introduction; the TRAFFIS project focuses on testing and training for ADAS. The ADAS example used here is a Headlamp Control Module (HCM). HCM is a driver assistance system developed by Varroc [11]. It controls the headlamps to ensure an optimum road illumination by using high beam as often as possible. The project partners defined 7 application scenarios, which were classified and summarized into 3 different simulator setups as following:

- TRAFFIS Full Version: This application scenario has the most complex structure with a full scale motion platform. Its target is to test an ADAS control unit and the camera as Hardware in the Loop (HiL) simultaneously together with a Driver in the Loop (DiL).
- TRAFFIS Portable Version: This application scenario is a stripped version from the full version; its target is to train truck drivers on the new ADAS.The training should be by the logistics centers onsite, therefore a portable system with simple motion platform is needed.
- TRAFFIS Light Version: The aim of the application scenario is to test ADAS as Software in the Loop (SiL). Here, only a computer based system is utilized without a motion platform. This version is useful for the ADAS developers in early phases to test his control algorithms daily in a simple way.

Active structure: Since the full version is the most complete simulator configuration, it is modeled in this partial model in order to identify the driving simulator main components and the relationship between these components, with help of the defined application scenario. The result of the active structure modeling will be presented next and it is shown also in a simplified way in Fig. 3; they are categorized into the well-known model-simulation-analysis process.

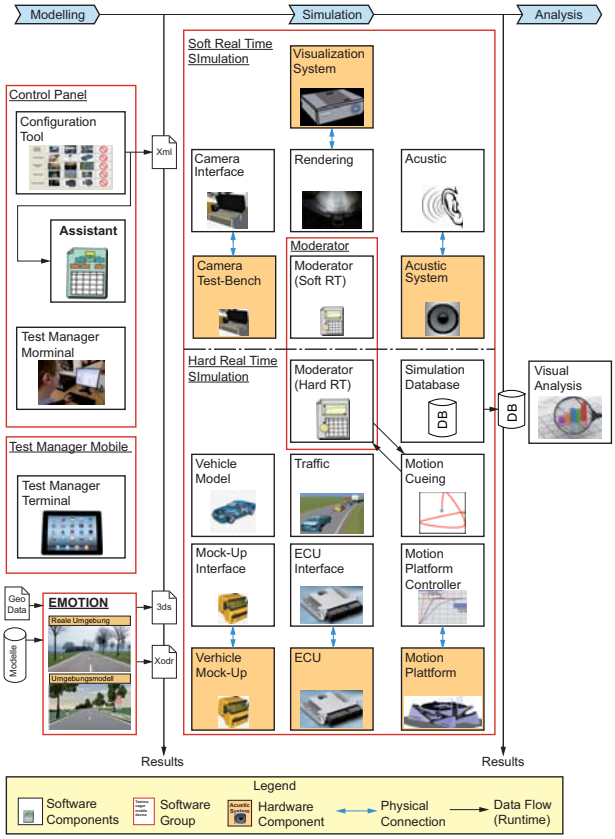

Figure 3. Identified driving simulator system components

The main result from the first phase is the identification of the driving simulator main components within in the principle solution; the important identified components are described as following:

Emotion (software): This is a software tool that allows the automatic generation of virtual roadways based on geographic information systems [12]. The software generates logic and graphic representation of the environment.

Configuration tool (software): Target of this tool is to create a driving simulator setup depending on the user needs by selecting the desired software and hardware components to generate a task specific driving simulator setup. This software concept will be described in detail in section C and D.

 ISBN: 978-1-61208-308-7

Assistant (software): This software operates in a preprocessing step, reads the configuration file, allocates selected software components, loading them, then distributes them over the available resources (computers, real-time hardware, etc.).

Operator (software): The operator software is a user-friendly graphical user interface, which enables the driving simulator operator to operate (load, start, stop, etc.) the whole system by using a standalone software.

Test manager mobile terminal (software): This software component runs on a mobile device, it's allow the test manager or the driving instructor to trigger some predefined events in run time.

Moderator (software): The moderator plays a very important rule during the simulation run-time. It reads the configuration file, which contains the communication and interface topologies between the different software components and establishes the communication between these components regarding their real-time specifications.

Vehicle mock-up (hardware): This is the real vehicle mock-up (driver's cabin), which represents the MMI (man-machine interface) between the driver and the driving simulator.

Vehicle mock-up (software): This software interface redirects the input from the dashboard in the driver cabin to the simulation software and vice verse the output from the simulation to the driver cabin.

Vehicle model (software): It is a software package developed to simulate the actual nonlinear physical characteristics of a typical vehicle in response to the operator inputs in real-time using a multi-body dynamics system.

Motion cueing algorithm (software): This describes the presentation of haptic information (cues) with the aim to resemble real movements in virtual environments [13].

Motion platform (hardware): Is an active mechanism, which allows haptic feelings of being in moving.

Motion platform controller (software): In order to synthesize a controller for actuators of motion system, different actuator controllers can be used to guarantee that the motion system of the simulator exactly follows a desired trajectory defined by the motion cueing algorithm [14].

Rendering (software): The rendering software visualizes the virtual driving scenario onto the projection system. It typically renders the driver view showing all detail of the current driving situation, as well as other traffic participants, the road, and the complete environmental scenery around the simulated vehicle.

Visualization system (hardware): The rendered scenery needs to be presented to the driver in the vehicle. Thus the rendering software feeds different types of visualizations devices such as screens or projectors.

Simulation results (database): During a driving session, some selected data are logged and stored in a database for later retrieval and analysis.

Visual Analysis (software): The logged simulation results are later used in the post processing phase for a visual analysis of the driving session.

The following components will be only listed but not described:

Traffic model (software), Control unit (hardware), Control unit (software), Camera test-bench (hardware) [15], Camera test-bench (software) [15], Acoustic (software) and Acoustic system (hardware).

B. Solution Elements Deployment

The result of the first phase is the identification of the main driving simulator components. Each component defines a function of hardware or software subsystem and its standard inputs outputs interfaces. Under each component is a group of possible solutions, which are called solution elements.

For example, the vehicle model is a software component defines the physical model of the vehicle under test, and the group of the solution elements could be (personal car model from dSPACE Company, personal car model from Paderborn University, truck model from dSPACE Company, etc.).

In order to consolidate all existing solution elements into the reconfigurable driving simulator, the existing solution elements have been identified and classified under its corresponding components; moreover a general interfacing concept had been developed.

To interface the entire solution elements together without in depth know how of each solution element, each solution element is considered as a black box. That means that only the inputs and outputs interfaces have to be taken in account. To keep the configuration process flexible and extendable any solution element could be added as soon as his inputs and outputs interfaces are defined. The only required task to integrate any solution element is to map its inputs and outputs to the predefined inputs and outputs of the parent component, which is called here signal multiplexing.

Fig. 4 shows an example of this concept: a vehicle model has to be integrated as a solution element, the model will be considered as a black box, but all the signals has to be mapped to the parent component signal description. The output signal which called "Otutput_ID563[m/s]" contains the vehicle under test velocity in m/s, but this signal unique name and unit predefined by the parent component is "Chasis_Velocity" in km/h.

In order to integrate this vehicle model; the user has to connect all the input and output signals with different names and units to the unique names and units of the parent component.

This concept allows the user to integrate new solution elements to the system with minimum effort, since this integration step will be done only once for each new solution element.

 ISBN: 978-1-61208-308-7

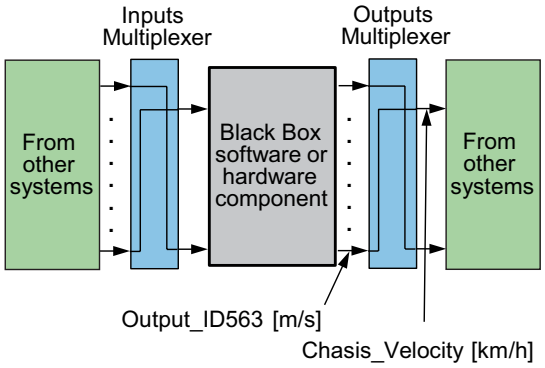

Figure 4. Integration of a Black Box model into Driving Simulator System

Since all solution elements are integrated within the configuration software, the result is a driving simulator configuration matrix shown in Fig. 5. The first column of the matrix lists all 23 components as identified in the first phase, while the rows contain different solution elements for each component. Fig. 5 shows the configuration matrix in form of morphological boxes.

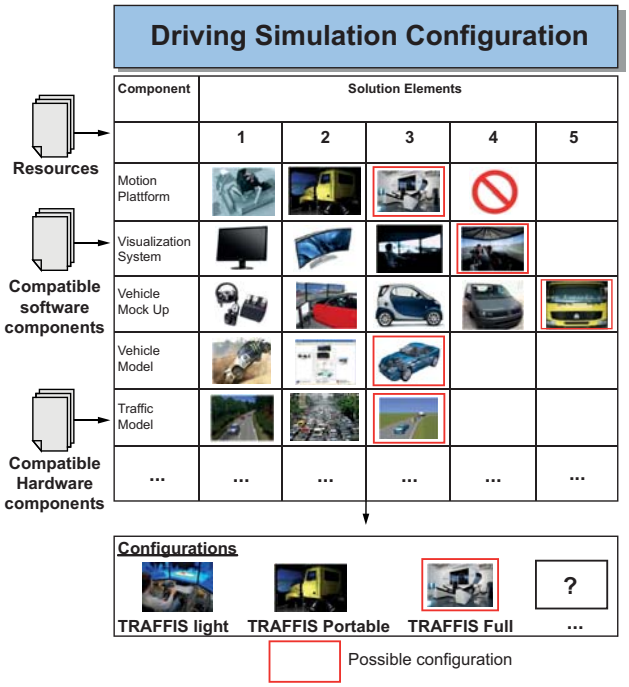

Figure 5. Driving simulator morphologische boxes

C. Configuration Mechanism Development

The result of the second phase is the configuration matrix in morphological boxes form. The morphological boxes allows the user to select one solution element under each row "component" and the combination of these selected solutions elements described a possible configuration. To ensure the consistency and compatibility of the selected solution elements, a configuration mechanism has to be developed to ensure the consistency and compatibility.

The configuration mechanism is ensuring the following function: preservation of the selected solution element consistency, checks the solution elements compatibility, add or modify components and solution elements and generating the configuration file, which contains the selected components, interface topology, selected resources, and signals for analysis. This mechanism concept and a prototypical implementation are done during master thesis supervised by the authors.

D. Configuration Tool Development

The result of the third phase is the configuration mechanism, which is a complex sequential process. In order to make a user friendly interface, in this phase, a concept and implementation of easy to use software has to be developed. The software guides the user during the selection process in a sequential order till the user finishing the configuration. The advantage of such software is to execute the configuration mechanism in the background without any annoying of the user. Fig. 6 shows a screen shot of the developed software.

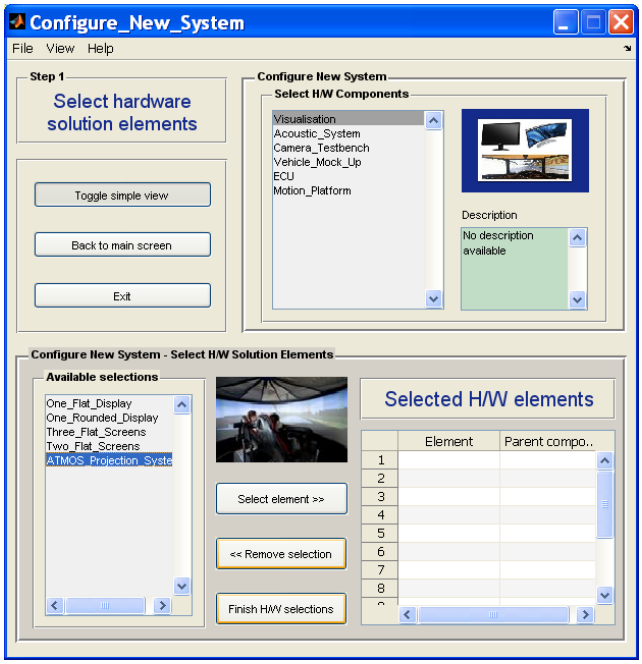

Figure 6. Configuration software

E. Simulation Initialization

The result of the fourth phase is the configuration file. The configuration file contains the selected components, interface topology, selected resources, and signals for analysis. In order to use the generated configuration file, there is a need to interface this configuration file with the simulation software. The interfacing between the configuration file and the simulation will done by two software components “the assistant” and “the moderator”, the interfacing will done through following steps:

Software distribution over resources: The assistant software reads the driving simulator configuration file; it retrieves the models, software components from the file storage system and distributes them over the available resources (computers).

Software initialization: The moderator software initializes the communication between all system components as described in the interface topology in the configuration file and prepares the system for start-up.

Simulation run-time: as soon as the user starting the simulation, the moderator software ensures the communication between all the system components and logs the selected signals for analysis in the configuration file during run time in the simulation database.

Post-processing: After the simulation session is finished, the moderator software logged signals from the simulation run-time for analysis.

Regarding the interface topology (moderator task on run-time), it is a challenge to connect all 23 components together. This work is often done manually by connecting input and output signals of the components directly. Moreover, manual connection has to be repeated or modified each time the driving simulator structure is changed to fit user specifications. Therefore, there is a real need for an automated method in order to moderate the interface between the selected solution elements, to allow the user to change the system structure or to replace one or more solution element without knowing details or the interface topology.

The concept of the communication topology is that all components will be connected by the moderator software via an input signal bus and an output signal bus instead of connecting the system components directly. Fig. 2 shows an example of the communication between the moderator software and the motion cueing component (two black arrows). The signal bus contains a detailed description of each signal it contains: unique signal name, signal unit, frequency, resolution, communication protocol, physical port, if this signal is mandatory or optional, and a description of the signal.

The moderator software ensures the communication between all the system components, as shown in Fig. 7. Moreover, the moderator software checks the selected signals for analysis in the configuration file, and logs them during run time in the simulation database.

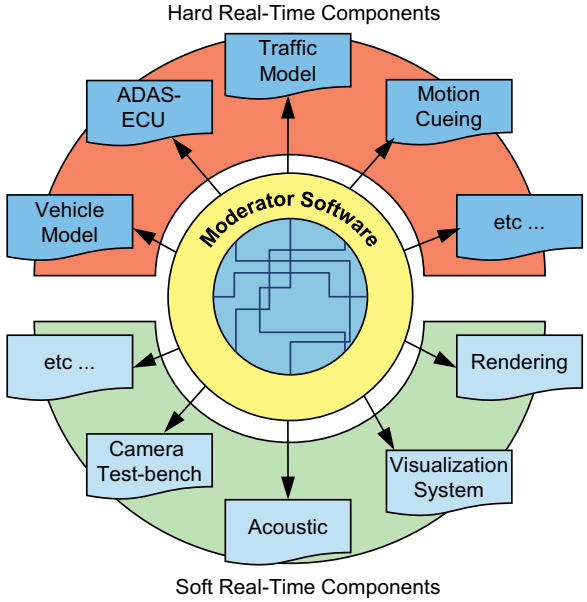

Figure 7. Moderator Software Task at run-time

IV. Conclusion and Future Work

This paper described a concept for a task-specific reconfigurable driving simulator for testing and training of ADAS. The concept demonstrates how individual subcomponents and their respective simulation models of varying fidelity and level of detail are configured into valid and working simulator setup based on the requirements of previously defined in the application scenarios. We applied the concept with the context of the TRAFFIS project and achieved promising results with our first reconfigurable simulator two prototypes (*TRAFFIS Full Version and TRAFFIS Light Version*). These prototypes vary distinctively from each other depending on the different application focuses for test and training of ADAS in their respective testing environments.

Within the TRAFFIS project context; we plan to further prove this concept by developing a third simulator prototype and elaborate more complex application scenarios for testing and training of further functions of ADAS in more complex environments.

Acknowledgment

This work, as part of the project TRAFFIS (German acronym for “Test and Training Environment for Advanced Driver Assistance Systems”), which is funded by European Union “ERDF: European Regional Development Fund” and the Ministry of Economy, Energy, Industry, Trade and Craft of North Rhine Westphalia – Germany, within the “Ziel2” program.

We thank our project partner dSPACE for providing detailed vehicle and traffic models, as well as specific HiL-simulation hardware. We thank our project partner Varroc for providing a head light control module for a adaptive bending lights.

REFERENCES

[1] B. Hassan, J. Berssenbrügge, I. Al Qaisi, and J. Stöcklein, "Reconfigurable Driving Simulator for Testing and Training of Advanced Driver Assistance Systems," Proc. International Symp. on Assembly and Manufacturing (ISAM 2013), July 30th – Aug 2th, 2013, Xian, China.

[2] K. Koscher, A. Czeskis, F. Roesner, S. Patel, S. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, and S. Savage, "Experimental Security Analysis of a Modern Automobile," Proc. IEEE Symp. on Security and Privacy, May 16-19 2010, Berleley/Oakland, California, USA.

[3] G. Weinberg and B. Harsham, "Developing a Low-Cost Driving Simulator for the Evaluation of In-Vehicle Technologies," Proc. of the First International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI 2009) Sep 21-22 2009, Essen, Germany.

[4] J. Berssenbrügge, "Virtual Night Drive – A Method for Displaying the Complex Light Distribution Characteristics of Modern Headlight Systems Within a Simulated Night Drive," Ph.D. thesis, Faculty of Mechanical Engineering, 2005, University of Paderborn, Germany.

[5] J. Negele, „Anwendungsgerechte Konzipierung von Fahrsimulatoren für die Fahrzeugentwicklung," Ph.D. thesis, Faculty of Mechanical Engineering, 2007, Technische Universität München, Germany.

[6] M. Abdel-Aty, X. Yan, and E. Radwan, " Using the UCF Driving Simulator as a Test Bed for High Risk Locations," Technical Report: Florida department of transportation, 2007, Florida, USA.

[7] D. Gue, H. Klee, and E. Radwan, "Comparison of Lateral Control in a Reconfigurable Driving Simulator," Proc. DSC North America, 2003, Dearborn, Michigan, USA.

[8] M. Lozano, M. Fernandez, and R. Martinez,"A Reconfigurable Projection System For Several Gantry Crane Simulators," Technical Report, 1999, University of Valencia, Spain.

[9] http://www.bvg.de/index.php/en/17106/name/Tram.html [retrieved: October, 2013]

[10] J. Gausemeier, T. Gaukstern, and C. Tschirner, "Systems Engineering Management Based on a Discipline-Spanning System Model," Proc. Conference on Systems Engineering Research (CSER'13), Eds.: C.J.J. Paredis, C. Bishop, D. Bodner, Georgia Institute of Technology, March 19-22, 2013, Atlanta, USA.

[11] C. Schmidt, "How to Make an AFS System Predictive: ADASIS Interface Implementation," Proc. 7th International Symposium on Automotive Lighting, September 25-26, 2007, Darmstadt, Germany.

[12] S. Kreft, J. Gausemeier, M. Grafe, and B. Hassan, "Automated Generation of Virtual Roadways based on Geographic Information Systems," Proc. International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (ASME 2011), August 29 - 31 2011, Washington DC, USA.

[13] J. J. Slob, "State-of-the-Art Driving Simulators," a Literature Survey, 2008, Eindhoven University of Technology, Eindhoven, Netherlands.

[14] I. Al Qaisi and A. Trächtler, "Constrained Linear Quadratic Optimal Controller for Motion Control of ATMOS Driving simulator," Proc. Driving Simulation Conference, 2012, Paris, France.

[15] Y. Tan and B. Hassan, "A Concept of Camera test-bench for testing Camera Based Advanced Driver Assistance Systems," Proc. International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (ASME 2013), August 4 - 7 2013, Portland, USA.

 ISBN: 978-1-61208-308-7