Modeling and Implementation of Cognitive-Based Supervision and Assistance

Xingguang Fu[1], Dennis Gamrad[1], Henning Mosebach[2], Karsten Lemmer[2], Dirk Söffker[1]
[1]University of Duisburg-Essen, Germany; [2]German Aerospace Center (DLR), Braunschweig, Germany

Corresponding Author: Dirk Söffker, University of Duisburg-Essen, Chair of Dynamics and Control
Lotharstraße 1, 47057 Duisburg, Germany; soeffker@uni-due.de

Abstract. This contribution deals with the implementation of an automated cognitive-based supervision concept applied for the first time on a real vehicle. Here, the Situation-Operator-Modeling (SOM) approach is used as a representational concept to model and formalize the logic of interaction between driver, vehicle, and environment based on sensor and video measurements. The programmed implementation of Driver-Vehicle-Interaction (DVI) model is realized by a Java-Application, which can be connected with the ViewCar©, an experimental vehicle developed by DLR which is equipped with specialized sensors and cameras. Prefilters are designed and integrated to interpret and transform the scenes into situations and operators. As an output, the interpreted situation and all allowed actions performed by the driver are displayed on a user interface. Furthermore, the application has been tested with a replay mode as well as the ViewCar© itself.

1 Introduction

The safety of passing maneuver was already a focus in automotive history [6], but the aspect of trajectory planning for the lane changing maneuver [7], is not considered here. In this work, an approach of automated supervision and driver assistance [1] is introduced, which differs from known approaches in the literature [4, 8, 10]. The main advantage is, that beside the supervision, the proposed underlying concept can also be used to fully autonomous driving.

2 The underlying structure of action logic: the Situation-Operator-Modeling approach

In [12], a systemtheoretic modeling approach was introduced, dealing with a special situation-operator modeling kernel (calculus) called SOM. This modeling approach combines classical ideas of the situation and event calculus introduced by McCarthy [9], and leads to a uniform and homogenous modeling approach allowing to describe human learning, planning, acting, and also the formal description of human errors. The introduced SOM approach gives the modeling framework, which means the structure of changeable scenes, and therefore maps the structured 'reality' of the real outside world of a system into a formalizable representation. This is useful to describe problems, where the structure of the considered system is complex and cannot be modeled with known or classical approaches. As an example for such a system, the interaction between human operators or other intelligent systems and their environment is mentioned.

The core of the approach is the assumption that changes in the parts of the real world to be considered are understood as a sequence of effects. The items scenes and actions of the real world are used to model these changes. The item scene denotes a problem-fixed moment in time but independent from time, and the item operator denotes the action changing the scene in Figure 1. Both are connected by the underlying logic of action, the operator follows the scene, the following scene follows the operator, etc. The definition of the items scene and action are coordinated in a double win. They are related to each other and therefore can also be used to relate the assumed structure of the real world to the structure of the database – called the mental model – of an intelligent system. Humans (as human operators) and intelligent systems are included in the real world. Depending on their principal sensory inputs, their natural or technical perceptions, and the related knowledge base, intelligent systems adapt and learn only parts or aspects of the real word. These parts can be modeled using the developed situation and operator calculus. The describable part defines the system to be considered.

3 Concept of cognitive supervision and assistance

Using the Situation-Operator-Modeling, a related concept for automated supervision is developed and proposed in [1, 2, 13]. For the representational level as part of the cognitive approach, the SOM is used to model and structure the complex scene of the driver-vehicle-environment interaction. First, the scene is modeled as situation. Therefore, its characteristics have to be defined. Then, the set of actions which can be performed by the driver is specified and modeled as operators. The passing maneuver is chosen as an example because it provides enough complexity to demonstrate the approach, but on the other hand it is not too complex to get lost in details.

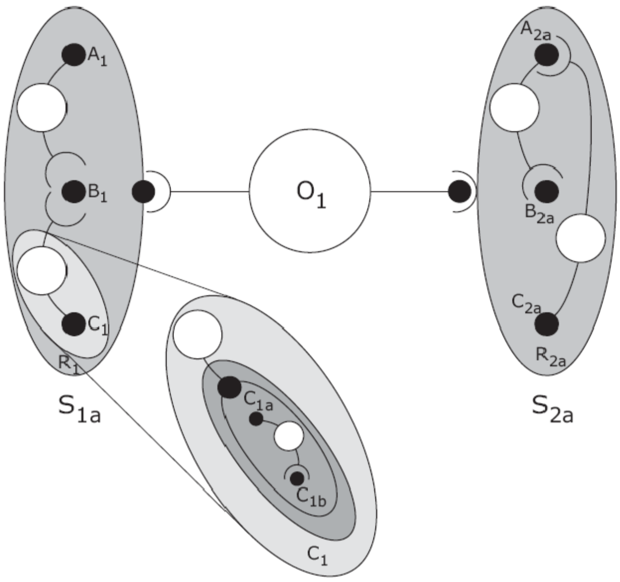

Figure 1. Graphical notation of SOM [12]

In Figure 2, the whole concept of automated supervision of the passing maneuver as an example is illustrated. On the sensory level, the relevant facts of the real world are perceived to set up a characteristic vector as situation description on the processing level. The operator representing the current action of the driver has to be identified from the existing basis operator library. With the situation description and the actual operator, a meta-operator representing the goal-oriented action of the driver can be chosen from the metaoperator library. On the analysis level, the assumptions of the operator are checked, whether the operator is – related to the actual situation - is applicable or not. This can be realized by checking the consistency between operators and situations as well as the interaction logic. Furthermore, models of typical human errors [3, 11], which are translated into SOM description [12], can be summarized in an error classification library. The operator and error libraries can be replaced or extended for automated supervision of other systems, while the overall structure remains unaltered. With this formalization possibilities the action sequence of the driver can be checked for consistency (checking assumptions), typical human errors, and goal conflicts so that the driver can be informed, if an error or conflict is detected.

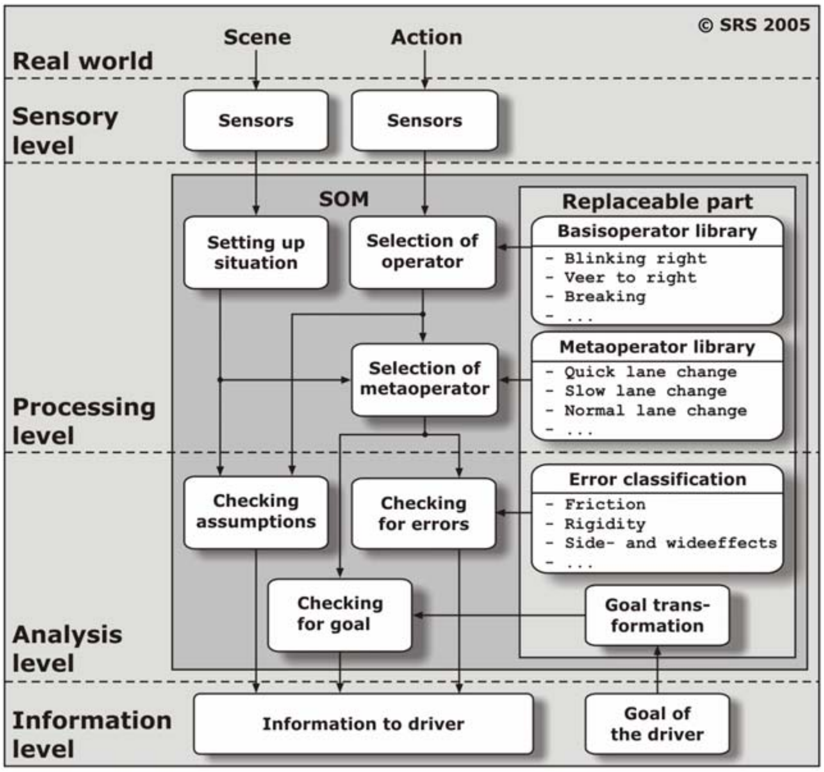

Figure 2. Concept of automated supervision [1]

4 Implementation within an experimental environment

The implementation of the concept of automated supervision to a real vehicle was firstly realized within a cooperation between the Chair of Dynamics and Control (University of Duisburg-Essen) and the Institute of Transportation Systems (German Aerospace Cente, DLR) [5]. Here, sensor and video data of the ViewCar©, a test vehicle equipped with several sensors and cameras, is used to build a general model of the Driver-Vehicle-Interaction (DVI). This is based on previous work [1, 2], as well as on new experiments with a visualization program displaying sensor measurements and video data together, which is shown in Figure 3.

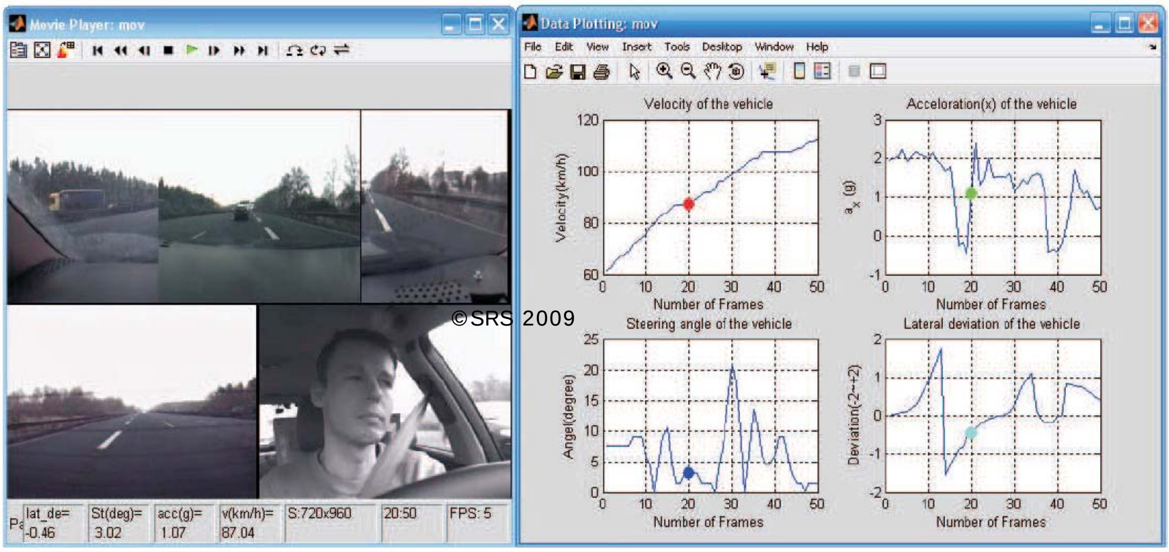

Figure 3. MATLAB-based visualization program displaying video and sensor measurements (cf. [5])

4.1 Establishment of Situations based on characteristics

Within the DVI model, the logic of interaction is represented using the Situation-Operator-Modeling approach. The sensor-based measurements of the vehicle are used as input of the model and are structured as a hybrid situation vector of the SOM-approach consisting of seven basic interpreted characteristics, as illustrated in Figure 4. Beside the ***actual velocity*** described by numeric values, the other characteristics are of different data types stated by a set of parameters, which are determined by actual scene presented by the measured sensor data.

Situation	Characteristic	Abrreviation	Parameter
•	Actual lane	al	[1/2/3/lane transition]
•	Actual velocity	v_a	[real]
•	Indicator signal set	iss	[off/left/right/warning lights]
•	Passing lane exists	ple	[left/right/left and right]
•	Actual lane free	alf	[yes/no]
•	Lane-change possible	lcp	[no/left/right/left and right]
•	Acceleration	ap	[yes/no]

Figure 4. Seven basic characteristics with parameters (cf. [5])

Most of the characteristics are not directly supported by the sensor data (such as *alf* and *lcp*), but obtained by a number of prefilters, which includes numerical calculations, data compressions, and other mathematical algorithms like fuzzy logics and neural networks etc.

Lane marks left (lm_l)	Lane marks right (lm_r)	***Passing lane exists***
0	0	Left and right
0	1	Left
1	0	Right
1	1	no

Table 1. Model 'Pendulum with Constraints', state event detection

For the characteristic ***passing lane exists*** (*ple*) as example, this characteristic gives information to the driver, whether there exists another lane for passing or changing. Associated with other characteristics like ***actual lane free*** and ***lane change possible***, the start situation of the lane-changing maneuver can be recognized. The parameter of *ple* varies with the change of the road structures, i.e., for a single-lane road the parameter is always denoted as linguistic value "no". Therefore, *ple* can be determined by detecting the lane marks on the highway, as listed in Table 1. Correspondingly, the prefilter of *ple* is graphically presented in Figure 5.

Prefilter of *passing lane exists*

Input sensor data: lm_l `[0/1]` lane marks on left hand side
lm_r `[0/1]` lane marks on right hand side

Algorithm:

If $lm_l = 0$ **and** $lm_r = 0$
then return `left and right`
else if $lm_l = 0$ **and** $lm_r = 1$
then return `left`
else if $is_l = 1$ **and** $is_r = 0$
then return `right`
else return `no`

Output: *passing lane exists ple* `[left/right/no/left and right]`

Figure 5. Prefilter of the characteristic ***passing lane exists*** (cf. [5])

4.2 Selection of Operators according to actions

As mentioned in Section 3, information about driver actions is processed with sensor data and then evaluated on processing level by a selection module on the sensory level. This module compares the information with the basic operator library to select the proper operators including a set of assumptions, which describe the relation of the operator to the initial situation. The assumptions are not fulfilled if the actual operator is not suitable to the current situation.

A driver controls the vehicle by operating different actuators such as pedals and the steering wheel. In the current DVI model, a number of actions of the driver are defined, including

- Drive,
- Accelerate,
- Decelerate,
- Brake,
- Set turning indicator left on/ right on/ off, and
- Veer to left/ right.

These actions are defined as basic operators, which can be identified by sensor data from the signals of indicator set, steering wheel, throttle pedal and brake pedal. By recognizing these sensor data, the operator selection module searches and compares the signals with the basic operators in the basic operator library, as illustrated in Figure 6. If one of the basic operators matches the signals from the sensor, it will be chosen and output as the actual operator.

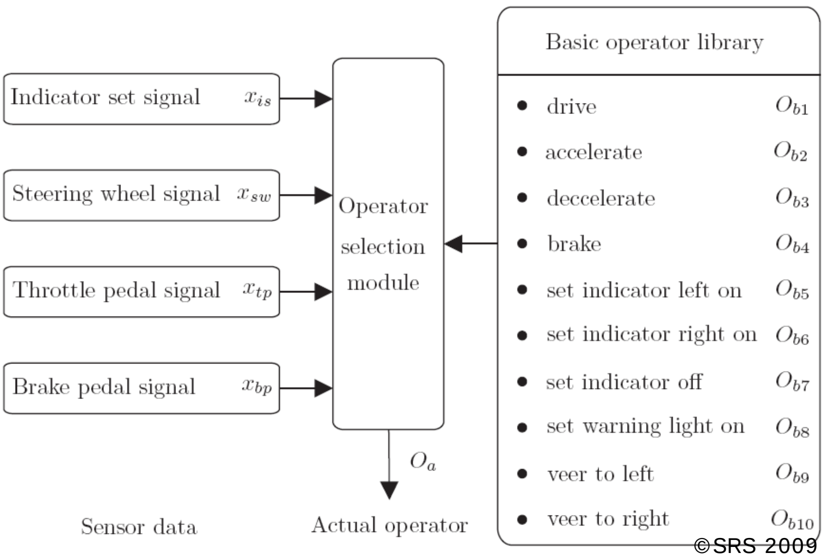

Figure 6. Setting up the actual operator by comparing the signals with the basic operator library (cf. [5])

4.3 Programming implementation

The implementation of this model is realized by a Java-Application, which enables the UDP-based communication between sensor system and a user interface. With the integrated prefilters and operator library, the application can provide online processing with raw sensor measurements from the database as input. The results of the analysis are fed back directly to the user interface, which displays the interpreted characteristics, operators matching the actual situation with respect to the assumptions, and the actual operator of the driver. Additionally, a warning message is shown if the actual operator does not match the actual situation. The structure of the modeling program is illustrated in Figure 7 and the user interface is shown in Figure 8, in which the driver takes an improper action "veer to left" that is not allowed according to the assumptions.

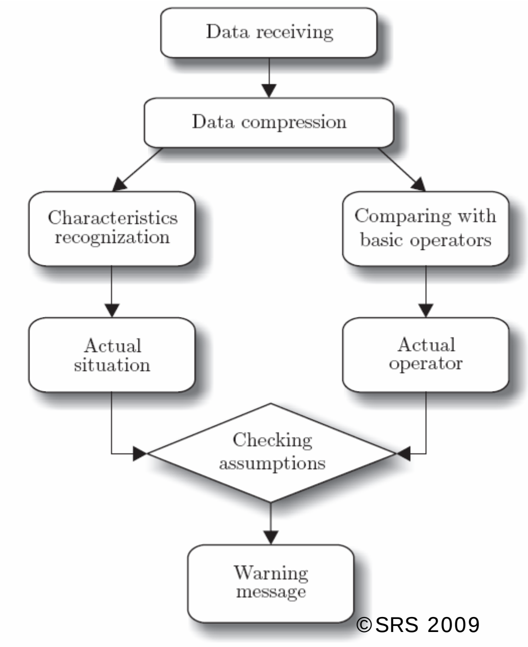

Figure 7. Structure of the modeling program (cf. [5])

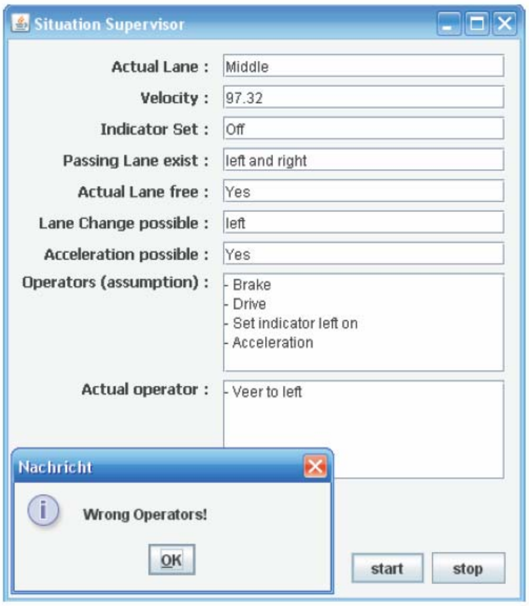

Figure 8. User interface (cf. [5])

The development and optimization of the DVI model is realized within an experimental environment. The equipment consists of a Can-analyzer, a Target-PC and the DVI software. The Can-analyzer is used so simulate the real ViewCar©. It sends the sensor measurements, which are stored in a data file, to the Target-PC over a CAN-Interface. The Target-PC displays the sensor measurements and forwards them to the DVI software per UDP communication.

4.4 Tests and results

Tests of the introduced DVI model was executed in both off-line and on-line mode, and finally in the real experimental ViewCar©. During the off-line tests, selected scenes of lane-changing maneuvers are experimented with the application in order to optmize and evaluate its performance. Due to the fact that the current sensor system does not support the measurements of distances between the ViewCar© to other vehicles in the environment, a set of assumed distance data is additionally added for the tests.

	Off-line mode	On-line mode
Characteristics	95%	85%
Current situation	90%	75%
Possible operators	95%	87%
Actual operator	98%	90%

Table 2. Identification rate of the characteristics, current situation, possible operators and actual operator in Off-line and On-line tests

Results from the both experiments are roughly listed in Table 2. Here, the data was measured from 5 hours highway driving of a certain driver. It can be seen that the off-line test has a better identification rate than the on-line mode because of the more complex driving environment and lack of the measurements support. As for the tests on the ViewCar©, the results turned out to be similar with on-line mode due to the fact that a Target-PC is also used in ViewCar© and therefore the DVI software works in the same way for both applications.

Besides, the warning functionality is also tested in different modes. In order to activate this function, a number of conflicts are created intentionally in the tests. For example, the signal of indicator set is manually deleted during a scene of changing lane to left. When this set of data is analyzed by the program, a warning message is given because the operator "indicator left on" is not detected when the driver veers to left.

5 Summary and future work

The contribution describes the implementation process of an automated cognitive-based supervision concept on a real vehicle. The introduced DVI model is used to formalize the interactions between human driver and vehicle by applying the Situation-Operator-Modeling approach. According to the approach, a set of basic characteristics are selected to describe the situations in the DVI model by means of programmable prefilters. Finally, the DVI model is implemented by Java-based programs which has been tested and validated on the ViewCar©.

The work in the future is focused on the analysis of more complex driving environments and the driving behaviours of a certain driver in order to personalize the program. Furthermore, the integration of cognitive-based supervision program and the driving assistance system should be improved.

6 References

[1] Ahle, E. and Söffker, D.: *Interaction of Intelligent and Autonomous Systems - Part II: Realization of Cognitive Technical Systems.* MCMDS-Mathematical and Computer Modelling of Dynamical Systems. Vol. 14, No. 4, 2008, 319 - 339.

[2] Blötz, N.: *Konzeption eines Überwachungsautomaten für Überholvorgänge im Straßenverkehr*. University of Duisburg-Essen, Chair of Dynamics and Control, Diploma Thesis, February 2005.

[3] Dörner, D.: Die Logik des Mißlingens, Strategisches Denken in komplexen Situationen. Rowohlt Verlag, 2002.

[4] Fiala, E.: *Lenken von Kraftfahrzeugen als kybernetische Aufgabe*. Automobiltechnische Zeitschrift 68, 1966, 156 - 162.

[5] Fu, X.: *Analysis and Modeling of a Driver-Vehicle-Interaction*. University of Duisburg-Essen, Chair of Dynamics and Control, Master Thesis, July 2008.

[6] Gibson, J. J. and Crooks, L. E.: *A theoretical field analysis of automobile driving*. Am. J. Psychol. 3, 1938, 453 - 471.

[7] Jula, H., Kosmatopoulos, E. B. and Ioannou, P. A.: *Collision avoidance analysis for lane changing and merging*. IEEE Trans. Vehicular Technol. 49, 2000, 2295 - 2308.

[8] Maurer, M. and Stiller, C. (Eds.): *Fahrerassistenzsysteme mit maschineller Wahrnehmung*. Springer Verlag, Heidelberg, 2005.

[9] McCarthy, J.: *Situations, actions, and causal laws*. Memo 2, Stanford University Artificial Intelligence Project, 1963.

[10] Naab, K.: *Automatisierung bei der Fahrzeugführung im Straßenverkehr*. In: at – Automatisierungstechnik 48, 2000. 211 - 223.

[11] Schaub, H.: *Modellierung der Handlungsorganisation*. Verlag Hans Huber, Bern, 1993.

[12] Söffker, D.: *Systemtheoretische Modellbildung der wissensgeleiteten Mensch-Maschine-Interaktion*. Logos Wissenschaftsverlag, Berlin, 2003. – also: Habilitation Thesis, Bergische Universität Wuppertal, Germany, 2001.

[13] Söffker, D.: *Understanding MMI from a system-theoretic view - Part II: Concepts for supervision of Human and Machine*. Proc. 9th IFAC, IFIP, IFORS,IEA Symposium Analysis, Design, and Evaluation of Human-Machine Systems, Atlanta, Georgia, USA, 2004.