Less Driving While Driving? An Approach for the Estimation of Effects of Future Vehicle Automation Systems on Driver Behavior

Bertram Wortelen and Andreas Lüdtke

OFFIS – Institute for Information Technology, R&D Division Transportation,
Escherweg 2,
26122 Oldenburg, Germany
{Bertram.Wortelen,Andreas.Luedtke}@offis.de

Abstract. We present a concept for the automatic estimation of the effect of newly introduced in-vehicle systems on driver's gaze behavior by means of executable cognitive driver models. Structure and properties of the complete system are shown. Particular attention is paid to the underlying model of visual attention which is integrated in the cognitive architecture CASCaS. Different alternatives for the operationalization of human expectancy as one influencing factor of visual attention are presented and discussed. Principal applicability of the model is demonstrated on a laboratory four-instrument visual scanning task. First results are briefly discussed. Future steps are outlined.

Keywords: Cognitive Driver Model, CASCaS, Visual Scanning, In-Vehicle System Evaluation.

1 Introduction

Since the time of the first automobiles, there has been a constant process of technical improvement. Especially in recent years a lot of new systems have been integrated in new cars, which aim to automate parts of the driver's task. The level of automation varies between different systems. For the longitudinal control task there exist informatory systems like collision warning systems as well as supporting, partially autonomous driver assistance systems (PADAS) like ACC (Adaptive Cruise Control).

Similar assistance can be found for other subtasks. There are Lane Departure Warning Systems and Lane Keeping Assistants for the lateral control task, Navigation Systems for Route planning and a lot of informatory and warning systems like Blind Spot Detection or Traffic Jam Warning. Current technical progress, e.g. in advanced sensor networks and C2C/C2I-communication, bear high potential for further automation in the automobile domain. The DARPA urban challenge is aiming at an even more ambitious goal. It strives to pave the future of driving by motivating international teams to develop vehicles able to drive fully autonomous in urban areas. Some of these teams, like the German team of Jörn Wille (TU Braunschweig) or the Google team of Sebastian Thrun, even tested their vehicles in real traffic.

V.G. Duffy (Ed.): Digital Human Modeling, HCII 2011, LNCS 6777, pp. 523–532, 2011.
© Springer-Verlag Berlin Heidelberg 2011

Besides automation another trend can be foreseen in the automotive area. The number and quality of infotainment systems is likely to rise in near future [1]. Introducing internet access, freely available development kits and operating systems will introduce a wide range of new applications. An example of a near to market system is the AutoLinQ system under development at Continental. A similar trend in the smartphones market resulted in a rapidly growing App-market. These Apps might easily allocate the free attentional resources drivers gain from using automation.

The introduction of more and more automation in cars will change driver's role step by step from manual control to supervisory control. A lot can be learned from the aviation domain, where the automation process was introduced earlier. Potential problems this change entails are loss of situation awareness, new attentional demands or complacency induced by too much confidence in automation [2]. Important to consider are behavioral changes induced by the automation which do not lead directly to accidents, but may be one contributing factor. Dekker [3] pointed out, that the difficult question is not, which function to automate, but how to ensure a good working cooperation between human and machine. On the one hand the assistance system should support the driver; on the other hand it should always keep the driver in the loop, because the driver is held responsible for the consequences induced by the driver-car system.

One very important aspect in the context of supervisory control systems is the human visual scanning behavior, as it directly influences how well the system operator is aware of the current system state. When introducing a new assistance system in a car, the system designer has to evaluate possible side-effects on drivers' behavior, including unwanted changes in their visual scanning behavior. Estimating the effect of a novel system on drivers' behavior can be very valuable information for the designer. Different complementary approaches exist for this purpose [4]. Testing the completed system or prototypes with human test participants is an established way of evaluating the system. It provides the advantage of directly observing reactions of the participants to the system. Participants are able to articulate their impressions and feelings of the system, while data recording allows also for objective measures of influences on driving behavior. Unfortunately this has to be done in a late design stage, where a prototype of the system is available. In addition the evaluation of long-term effects is expensive and normally only applicable after market introduction.

Another alternative is to consult a human factor expert to give estimations about the system's effect on driver behavior, which is a good opportunity in early design stages, as the expert can already work on informal system descriptions.

Our approach aims at giving software tool support for the evaluation process and partially automating it. The basic concept is to do closed loop simulations with all involved components of the driver-vehicle-environment (DVE) system, including an executable model of the driver. In an optimum case the integration of a system prototype and an interaction description on how to use the assistance system into the DVE simulation system leads to an automatic prediction of drivers' behavior in a way it would have been observed by using test participants in a driving simulator.

Predicting human interaction with novel technical systems in a simulative way is a known approach in scientific literature. A famous general purpose example is CogTool [5] which automatically generates a predictive human performance model

from a user interface prototype. Salvucci adopted this approach for his promising Distract-R [6] system, which predicts changes in driving behavior when interacting with an in-vehicle interface.

In the spirit of Distract-R we are investigating whether the effect of newly introduced systems on visual scanning behavior can be evaluated in an automated way. Our approach goes beyond Distract-R as it does not only take the direct interaction with the user interface into account, but also the demands that are placed on the driver's visual scanning strategy. This allows evaluating a wider range of systems. We will show how our approach might even be used to evaluate partially autonomous driver assistance systems, which require only few direct interactions with the user interface. Instead these systems are critical for their possibility to provoke out of the loop effects. We aim to predict these effects by simulating the adaption of drivers' visual scanning behavior to the newly introduced system.

2 System Structure

Our long-term objective is to make a software-supported methodology available for systems designers, which allows predicting effects of a system design on driver's behavior. One important requirement for the practical application of this methodology is that it should add as little additional costs to the design process as possible, in terms of time and effort. In this section we present the software system structure and describe how the individual components are configured to enable an efficient interaction with the system designer.

2.2 Joint Driver-Vehicle-Environment Platform

Our research focuses on modeling and simulating driver behavior within a joint driver-vehicle-environment (JDVE) platform. The specific platform we are using has been developed by Schindler et al. [7] to especially fit the needs of a model driven system evaluation process.

The platform consists of all components typical for a realistic commercial driving simulator (see Fig. 1). The environment component contains a model of the physical environment, the surrounding traffic and the scenario with its related events, while the vehicle component simulates the dynamics of a specific vehicle model.

The platform has been designed in a flexible way to allow an easy exchange and integration of new components. This allows integrating new driver assistance systems into the simulation. The driver component is also exchangeable. It can be connected to a physical driving simulator cabin to do test with human drivers, or it can be instantiated by an executable driver model. This allows direct comparison of model behavior and human driver behavior.

In use with driver models the JDVE platform reveals further advantages. As no interaction with human driver is needed, there is no need to do real-time simulations. The JDVE platform is able to run in an as-fast-as-possible mode to significantly speed up processing time. By disconnecting all visualization components from the simulation the processing time can be further improved. A built in batch execution support allows to repeat the same scenario multiple times to produce driver behavior traces in a Monte-Carlo like way, showing a reasonable realistic variance in behavior.

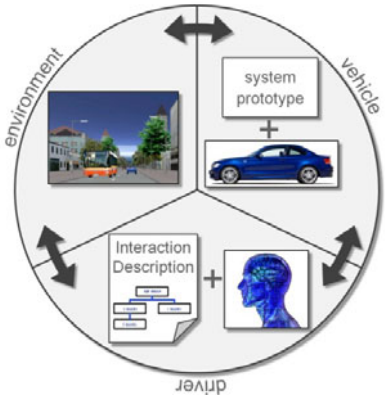

Fig. 1. JDVE setup for automatic system evaluation. The human driver is replaced by a driver model combined with an interaction description for the system under investigation.

2.2 Driver Model

The driver model we are using basically consists of two parts: a task independent and a task dependent model of human behavior. For the task independent model we use the cognitive architecture CASCaS (Cognitive Architecture for Safety Critical Task Simulation) developed at the German institute OFFIS. CASCaS was originally developed within the Aeronautics domain to simulate pilot-cockpit interaction [8]. In recent years it has been extended with several competencies to account for the special needs of driver models [9, 10].

It again contains a set of sub models (see Fig. 2) able to simulate different cognitive processes which are involved in everyday tasks. This allows CASCaS to produce reasonable realistic human-like behavior. But CASCaS alone cannot generate behavior. It additionally needs a description of the knowledge humans possess for handling a specific task like flying a plane or driving a car.

The representation of task dependent knowledge can widely differ. This depends on the layer chosen for processing of knowledge and the modeling technique chosen for the process implementation on that layer. Based on Fitts' concept of the three learning stages [11] CASCaS has three processing layers. The core layer of CASCaS which is used in every CASCaS model is the associative layer. It interprets GOMS like [12], rule-based knowledge to generate behavior. It represents conscious but familiar processing steps. In contrast the cognitive layer represents information processing in unfamiliar situations and the autonomous layer represents fast and automated behavior, which requires no conscious processing.

While knowledge processing on the associative layer is fixed within the architecture, the processing on the cognitive and autonomous layer can be adapted for the special needs of a specific model. For these layers an interface is provided to dynamically connect software libraries, which do the actual knowledge processing, regardless of the utilized modeling technique.

For the autonomous layer several modeling techniques have been evaluated for use within a driver model. The control theoretic model of Salvucci and Gray [13] for

lateral vehicle control and Boer's[14] model for longitudinal control, as well as Möbus' and Eilers' [15] BAD-MoB models for both aspects have been adapted for the sub tasks of lane-keeping, car following and driving with no leading vehicle. They have been used in different scenarios, including highway, rural and urban areas.

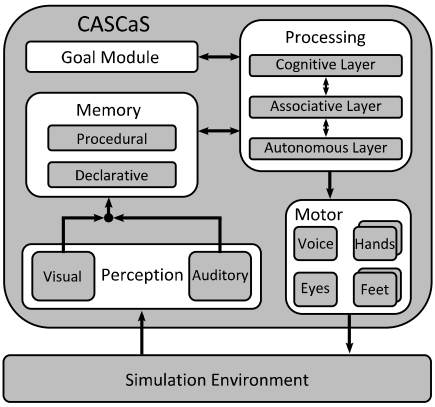

Fig. 2. High-Level view on main components of CASCaS

SVMs, stepwise linear regression models and BAD-MoB models have been evaluated for modeling parts of the behavior at traffic light approaches.

2.3 Scanning Model in CASCaS

The process that mainly drives the gaze behavior of models executed in CASCaS is the selection of goals. CASCaS maintains a set of active goals (G_{act}) which the model tries to achieve at the same time. This set can change dynamically during the simulation. For a driver model such a set can be:

- Keep appropriate distance to leading vehicle
- Keep car within driving lane
- Find a route to target location
- Listen to radio / talk to passengers / answer mobile phone …

Typically goals are hierarchically decomposed into sub goals up to single actions.

The goals listed above are all active at the same time. On the associative layer only one of the active goals is processed at any time. We refer to this as the selected goal. To achieve all goals the model has to do multitasking by switching the selected goal from time to time. Which one is selected is determined by the goal module.

The autonomous layer in contrast can really process multiple goals in parallel as the behavior on this layer is very skilled and needs little conscious processing. But these goals might fight for the limited resource of visual attention. In such a case the selected goal will get the resource. Therefore goal selection is the process that mainly drives the gaze behavior of models executed in CASCaS.

In order to enable a realistic gaze behavior, we integrated a visual scanning model into the goal selection process of CASCaS. We choose the SEEV-model of Wickens

[16], which predicts the probability of visually attending an information source based on four general influencing factors:

$$P(IS) = \underbrace{S - Ef}_{Bottom-up} + \underbrace{Ex \cdot V}_{Top-down} . \tag{1}$$

The influencing factors are:

(S) Saliency of the information source
(Ef) Effort required for obtaining information from an information source
(Ex) Expectancy, that an information source offers new information
(V) Value of the information from an information source for solving a particular task

We focus on the top-down factors Expectancy and Value as these are highly relevant for very familiar and skilled task executions. Like Wickens [16] we also distinguish between different sources of expectancy:

$$Ex = Cx + Er . \tag{2}$$

(Cx) Contextual Cues, like an alarm signal, which probably results in moving attention to the source of the alarm
(Er) Event rate (information bandwidth). Denotes how often an information source provides new information.

As the goal selection mainly drives visual attention in CASCaS, we enable the annotation of goals with the top-down factors of the SEEV model. The probability of selecting a goal ($P(g)$) in CASCaS is given by the relative value of these factors:

$$P(g) = \frac{Ex(g) \cdot V(g)}{\sum_{g_a \in G_{act}} (Ex(g_a) \cdot V(g_a))} . \tag{3}$$

Care has to be taken with this implementation as goals and information sources cannot be used synonymously. A goal in CASCaS can use more than one information source. As a consequence the described way of goal selection would result in different percent dwell times (PDTs) for the information sources, than the original SEEV model would predict. There are easy ways to work around this in CASCaS which we will not discuss in this paper.

2.4 Operationalization of Expectancy

Assuming that a system designer has to do the work of modeling the system interaction, then she/he has to define the parameters for the SEEV model. System designers are normally engineers, who are not very knowledgeable in human factor issues. From their point of view defining these parameters is much effort for little gain. To reduce the effort and increase the acceptability of the method for the designer, we provide the possibility to automatically estimate the expectancy factor Ex. In this section we first present how the event rate Er as part of Ex can be operationalized. Then we show how it can be generalized to also include expectancy stemming from contextual cues (Cx).

We define an event in this context as a reaction of the model to some visual information. The number of reactions per time defines the event rate. The idea behind this definition is that a reaction to information is a good indicator for the relevance of that specific piece of information for the current task. For sure there are exceptional cases where this assumption is not true. People might perceive and notice relevant information without showing a visible response. Nevertheless we think that in most cases this gives a useful approximation.

Within CASCaS a reaction to visual information occurs when a procedure rule is fired, whose left hand side matches the visually perceived information and whose right hand side defines the actions to be taken. Thus an event is derived dynamically from the simulation of the model's interaction description.

This has some consequences. This definition of event rate deviates slightly from the one of Wickens [16]. Wickens does not explicitly operationalize an event, but in his model the event rate, respectively information bandwidth, is an environmental property of the information source, which in principal can be measured in an objective way. By using our definition the interaction description explicitly defines all possible events. But the model might miss some events and only the ones it perceives determine the event rate. Thus it is a subjective measurement of the model, which can deviate from the real environmental event rate.

Another consequence is, that the model is now able to adapt to changes in the event rate, because the event rate is no longer a static parameter, but determined dynamically during simulation.

Assuming that humans are good at adapting to properties of the environment, they might also adapt to more aspects of the event sequence than just the average event rate. An information source that periodically produces events with a fixed distance between two events might result in a different visual scanning behavior than an information source with the same average event rate, but a very random distribution of event distances.

So a better strategy would be to base the decision where to look on the probability that new relevant information is available at an information source, given the time since the last event. Therefore we refine the operationalization of event rate by:

$$Er(g) = P(e_g \mid \Delta e_l) \ . \tag{4}$$

Where $P(e_g \mid \Delta e_l)$ is the probability that since the last event e_l a new event e_g has been produced for goal g under the condition that the time since the last event for that goal is Δe_l. This probability is obtained by tracing the cumulative event distance distribution during simulation.

Again assuming, that people are good at adapting to environmental properties, they might not only use the time since the last event Δe_l as a predictor for new events. Other contextual information can also be used as predictors. Therefore we generalize equation 4 to also include contextual predictors:

$$Ex(g) = P(e_g \mid \Delta e_l, cx_1, \cdots, cx_n) \ . \tag{5}$$

Thus event rate Er and contextual cues Cx as expectancy producing factors are treated in a similar way within CASCaS. Which factors are used as predictors has to be defined by the modeler.

3 Application to Laboratory Task

In order to show the general application of the visual scanning model, we created an interaction description for a well known laboratory task developed to investigate scanning behavior. It is the Four Instrument Scanning Task of Senders [17].

In section 2.3 and 2.4 we showed that CASCaS offers different ways to use the scanning model. One way is to define fixed values for the top-down SEEV parameters. In the CASCaS procedure language this is done by annotating goals, when they are instantiated in the model. For observing the gauges we create one goal for each gauge and annotate it with the pointer speeds Senders used for each instrument measured in rad/s:

```
Goal, name=observe_gauge_1, value=1.0, event_rate=0.5
Goal, name=observe_gauge_2, value=1.0, event_rate=1.0
Goal, name=observe_gauge_3, value=1.0, event_rate=2.0
Goal, name=observe_gauge_4, value=1.0, event_rate=4.0
```

Due to the fixed parameters we are able to calculate the expected fixation probability for each gauge before the simulation. Gaze fixation and selected goal should correlate perfectly for this model. Therefore we determine the fixation probability for a gauge by its respective goal: $P_{fix}(g)$. It is given by the probability that a transition is done from any other goal to that specific one. The probability of doing a transition from goal a to goal b is given by the probability of having selected a times the probability of selecting b as next goal.

$$P_{a \to b} = P_{fix}(a) \cdot \frac{P(b)}{\sum_{g \in G \setminus a} P(g)} \tag{6}$$

The sum term does not include $P(a)$ because reselecting a would not result in a transition neither of selected goal nor of fixated information source. With equation 6 we get:

$$P_{fix}(g) = \sum_{\tilde{g} \in G \setminus g} P_{\tilde{g} \to g} \quad . \tag{7}$$

Solving the resulting set of linear equations gives the fixation probabilities in column 3 of Table 1. We connected the executable CASCaS model to an implementation of Senders' task, so that it receives the current state of all gauges. Running the model for one hour resulted in the simulated fixation probabilities listed in the fourth column. Simulated and calculated probabilities are almost identical, which just indicates, that the implementation is working correct. Correlation between model and experimental data is high with a Pearson coefficient of 0.881, but a very small sample size (n=4).

4 Application to In-vehicle System

Infotainment systems tend to draw drivers attention to their display, while PADAS might reduce the attentional demand of the primary driving task. So both kinds of

systems bear the potential danger of inappropriately reducing the amount of time drivers spend looking at the road scenery, especially when used in combination.

The proposed modeling approach aims at predicting the changes in visual attention. The example of Senders task served to investigate the principal applicability of the visual scanning model. For realistic systems a lot of additional problems are expected.

The definition of an event might not be clear in every situation. People might especially weight events differently.

The information value might be hard to define in relation to other task.

Gaze behavior in real world tasks is normally not a pure scanning behavior. In some situations people show fixed gaze patterns. When changing lanes on a highway people develop gaze patterns like looking-to-side-mirror → looking-over-shoulder → looking-to-leading-vehicle. In other situations people show a visual search behavior, e.g. when searching for specific information on a display. Visual search behaviors and fixed gaze patterns are mixed with visual scanning behavior in real world tasks.

Our next step is to apply the visual scanning model to a cognitive driver model. We will investigate whether the visual scanning model is able to make useful predictions despite all the anticipated problems. We will investigate a scenario where two kinds of systems are involved. One system is a variant of an ACC as an example of a partially autonomous driver assistance system, which is expected to reduce attentional demands of the primary driving task. The other system is an artificial visual search task which is used as placeholder for in-vehicle information systems.

Table 1. Calculated, simulated and observed fixation probabilities for each instrument

	Signal bandwidth	Calculated P_{fix}	Simulated P_{fix}	Observed P_{fix} (Senders [13])
Gauge 1	0.5 rad/s	0.1	0.095	0.077
Gauge 2	1 rad/s	0.186	0.185	0.172
Gauge 3	2 rad/s	0.314	0.319	0.214
Gauge 4	4 rad/s	0.4	0.402	0.537

5 Discussion

We successfully integrated parts of the SEEV model into the cognitive architecture CASCaS and demonstrated it on a laboratory visual scanning task.

With equal value parameters and fixed event rate parameters our implementation is basically identical with Senders' Random Constraint Sampler [17]. Even though the results seem to reasonably fit Senders' data, in later experiments with a six-instrument setup [18] Senders argued that the Random Constrained Sampler fails in explaining experiment results. The model especially underestimated fixation probabilities for instruments with low bandwidths and overestimates high-bandwidth instruments.

Acknowledgments. The research leading to these results has received funding from the European Commission Seventh Framework Program (FP7/2007-2013) under grant agreement n° 218552, Project ISi-PADAS.

References

1. Löwer, C., Bolduan, G.: Smartphones auf Rädern, vol. (1), pp. 48–51. Heise, Hamburg (2011)
2. Sarter, N.B., Woods, D.D., Billings, C.E.: Automation Surprises. In: Salvendy, G. (ed.) Handbook of Human Factors & Ergonomics. John Wiley & Sons, Chichester (1997)
3. Dekker, S.W.A.: Ten Questions about Human Error: A New View of Human Factors and System Safety. Lawrence Erlbaum Associates, Mahwah (2005)
4. Servel, A., Knapp, A., Jung, C., Donner, E., Dilger, E., Tango, F., Mihm, J., Schwarz, J., Wood, K., Ojeda, L., Neumann, M., Brockmann, M., Meyer, M., Kiss, M., Flament, M., Kompfner, P., Walz, R., Cotter, S., Winkle, T., Janssen, W.: Code of Practice for the Design and Evaluation of ADAS. Tech. Report, V3.0 (2010), http://www.prevent-ip.org/download/deliverables/RESPONSE3/D11.2/Response3_CoP_v3.0.pdf
5. John, B.E., Blackmon, M.H., Polson, P.G., Fennel, K., Teo, L.: Rapid Theory Prototyping: An Example of an Aviation Task. In: Human Factors and Ergonomics Society Annual Meeting Proceedings, vol. 53(12), pp. 794–798. San Antonio, Texas (2009)
6. Salvucci, D.D.: Rapid Prototyping and Evaluation of In-Vehicle Interfaces. In: ACM Trans. of Computer-Human Interaction, vol. 16(2), pp. 9:1–9:33. ACM, New York (2009)
7. Schindler, J., Harms, C., Noyer, U., Richter, A., Flemisch, F., Köster, F., Bellet, T., Mayenobe, P., Gruyer, D.: JDVE: A Joint Driver-Vehicle-Environment Simulation Platform for the Development and Accelerated Testing of Autonomous Assistance and Automation Systems. In: Cacciabue, P.C., Hjälmdahl, M., Lüdtke, A., Riccioli, C. (eds.) Human Modeling in Assisted Transportation. Springer, Italia (2011)
8. Lüdtke, A., Pfeifer, L.: Human Error Analysis Based on a Semantically Defined Cognitive Pilot Model. In: Saglietti, F., Oster, N. (eds.) SAFECOMP 2007. LNCS, vol. 4680, pp. 134–147. Springer, Heidelberg (2007)
9. Wortelen, B., Zilinski, M., Baumann, M., Muhrer, E., Vollrath, M., Eilers, M., Lüdtke, L., Möbus, C.: Modelling Aspects of Longitudinal Control in an Integrated Driver Model: Detection and Prediction of Forced Decisions and Visual Attention Allocation at Varying Event Frequencies. In: Cacciabue, P.C., Hjälmdahl, M., Lüdtke, A., Riccioli, C. (eds.) Human Modeling in Assisted Transportation. Springer, Italia (2011)
10. Lüdtke, A., Osterloh, J.-P., Weber, L., Wortelen, B.: Modeling Pilot and Driver Behavior for Human Error Simulation. In: Duffy, V.G. (ed.) ICDHM 2009. LNCS, vol. 5620, pp. 403–412. Springer, Heidelberg (2009)
11. Fitts, P.M., Posner, M.I.: Human Performance. Brooks/Cole, Belmont CA (1967)
12. Card, S.K., Moran, T.P., Newell, A.: The Psychology of Human-Computer Interaction. Erlbaum, Hillsdale (1983)
13. Salvucci, D., Gray, R.: A Two-Point Visual Control Model of Steering. Perception 33(10), 1233–1248 (2004)
14. Boer, E.R., Ward, N.J., Manser, M.P., Yamamura, T., Kuge, N.: Driver Performance Assessment with a Car Following Model. In: Proceeding of the Third International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design, pp. 433–440 (2005)
15. Möbus, C., Eilers, M.: Mixture of Behaviors and Levels-of-Expertise in a Bayesian Autonomous Driver Model. In: Duffy, V.G. (ed.) Advances in Applied Digital Human Modeling, pp. 425–435. CRC Press, Boca Raton (2010)
16. Wickens, C.D., McCarley, J.S.: Applied Attention Theory. CRC Press, Boca Raton (2008)
17. Senders, J.W.: The Human Operator as a Monitor and Controller of Multidegree of Freedom Systems. IEEE Trans. Hum. Factors Electron HFE-5(1), 2–5 (1964)
18. Senders, J.W.: Visual Scanning Processes. University of Tilburg, Netherlands (1983)