Context Processing for Automotive Human-Machine Interfaces

A Demonstrator for Situation-Aware Interaction in Road Vehicles

Ljubo Mercep, Alois Knoll
Robotics and Embedded Systems
Technische Universität München
Garching, Germany

Gernot Spiegelberg
Siemens Corporate Technology
Siemens AG / TUM Institute for Advanced Study
München, Germany

***Abstract*—In this work, we focus on the improvement of human-machine interaction inside the automobile by reducing the complexity of involved context management. A contribution in the form of a context processing engine named probabilistic application layer (PAL) is provided, which addresses following issues: Guarantees for time bounds in performing safety-critical exact inference, standardized application interface towards application space, dynamic changes on the structure of the belief networks by adding or removing nodes while retaining time guarantees. By addressing these issues, we take the context processing away from the application developer and ease the development of the situation-aware human-machine interaction. Additionally, we present a prototypical implementation of PAL, implemented natively on an embedded platform and connected to the human-machine interface of our project vehicle prototype.**

Keywords—situation awareness; knowledge modeling; human-machine interaction; human-machine interface; driver assistance

I. Introduction

The human-machine interaction in a current road vehicle can be divided into three categories: active control of the vehicle's dynamics, control of various safety-relevant vehicle functions (e.g. direction indicators) and, finally, control over the non-safety-relevant infotainment functions [5]. During the interaction with the vehicle throughout all three categories, the driver might inadvertently perform different kinds of errors affecting the vehicle control. Such human errors are party responsible for 95% of all accidents. Human behavior is the sole responsible cause in 75% of all the cases, showing a clear mismatch between driver skills and situation and task complexity [4].

The work by [11] and [15] suggests further division of driver errors into cognitive, judgment and operation errors, which are mapped onto the corresponding human processes of observation, assessment and action. So-called *assisting assessment applications* are called upon, which analyze all the in-vehicle information sources, detect hazardous situations and inform the driver. Such applications should be uniformly present throughout the three human-machine interaction categories, in order to provide a consistent user experience.

Work by Müller in [3] presents multiple possibilities of restricting the control over infotainment functions in order to avoid lowered driver performance in vehicle control and the associated safety risks. Rule-based priorization of feedback data is found to be relevant for avoiding driver distraction, but it can only take place after the current driving situation has been correctly assessed. Thus, situation awareness precedes the rule-based approach of preparing the driver feedback.

Situation awareness is defined as the perception of environmental elements with respect to time and/or space, the comprehension of their meaning, and the projection of their status after some variable has changed. It can be recognized as an important enabler of decision making in the field of artificial intelligence. In the case of the man-machine interaction, and drawing a comparison to human-human interaction, situation awareness additionally has the potential to improve the interaction by following means:

- The amount of data which has to be exchanged between the human and the machine to perform a specific task or come to a common understanding is reduced
- The feedback which is provided by the machine can be affected by user workload and information relevancy
- Users can be modeled based on the previous and current interactions and an assessment of the mental or physical state or overall proficiency for a given task can be performed

In this work, we present an enabling technology for development of assisting assessment applications in the area of human-machine interaction and driver assistance, as defined by [11], and demonstrate the first implementation thereof.

This work is organized as follows. Section II gives an overview of the project Diesel Reloaded, which set the framework for the presented research. Section III focuses on current methods for knowledge modeling and related work in the area of driver assistance. Section IV presents the approaches for inference on the chosen knowledge model. Section V presents our contribution, the Probabilistic Application Layer (PAL). Section VI explains the PAL prototype design, focusing on the current software implementation. Finally, we conclude and provide an overview of future work in the section VII.

II. Project Diesel Reloaded

In the scope of the interdisciplinary project "Diesel Reloaded", research is performed in the fields of system architecture, energy management, human-machine interfaces and driver assistance. The project is hosted at the International Graduate School for Science and Engineering (IGSSE) of the Technische Universität München. The project leader is Prof. Dr.-Ing. Gernot Spiegelberg, a Rudolf-Diesel-Industry Senior Fellow at the Institute for Advanced Study (IAS) of the Technische Universität München and the leader of the E-mobility initiative in Siemens Corporate Technology. During the first two project years, a serial plug-in diesel-electric truck has been constructed, the Innotruck. It has been successfully test-driven with a fully functional drive-by-wire system. The human-machine interface consists of a touchscreen-based central console and two redundant sidesticks integrated into the driver's seat.

Fig. 1. Innotruck on the left, a photo of vehicle's cockpit on the right.

A driving simulator developed by the company VIRES, the Virtual Test Drive (VTD), has been used for development purposes, testing and data collection. The simulator has been equipped with a sidestick and connected to the PAL over a standard VTD software interface.

III. Knowledge Modeling

In order to implement a reasoning method which achieves situation awareness, one has to represent all the known data about the user, the vehicle and the outside world. When considering related work regarding descriptions of dynamic systems with varying degrees of knowledge uncertainty, with the final purpose of inferring a small and changeable subset of questions regarding to the current situation, several probability frameworks come into consideration.

The Bayesian networks (BN) represent joint probabilities of a set of random variables and their conditional independence relation [8]. Dynamic BNs (DBN) work with multiple time slices, which are interconnected into a larger single network. The connections can be understood as Hidden Markov Models. DBNs therefore present a generalization of a system for modeling dynamic events [13].

Bayesian and Markov networks (MN) are a very popular method of choice for uncertainty management in artificial intelligence. A BN, represented by an acyclic directed graph, can be converted into the MN through moralization and triangulation. A MN is represented by an acyclic hypergraph, which is a chordal undirected graph in which each maximal clique corresponds to a hyperedge [12]. DBNs have already been used for intelligent user-assistance systems, to model multimodal sensory observations, changing state of the user and various constraints regarding available resources. The DBN manages sensor measurement ambiguity, evolvement of user's affective state over time and decisions about the user's needs.

Situation-aware driver assistance systems have already been implemented with DBNs in the, for this work very relevant, publication [11]. The importance of fusion of human-machine interfaces, machine-machine interfaces and driver assistance systems has been thoroughly analyzed and a model for generating probability of hazards has been presented. Main components of the required architecture are a knowledge broker, a utility-based knowledge exchange and a reasoner. The authors recognize that the interaction with the driver can benefit from situational awareness through an intelligent information feedback management. The main contribution is the identification of major architectural components of a situation-aware driver assistance system, with a focus on sensor-based knowledge derivation and vehicle-to-vehicle and vehicle-to-infrastructure connectivity. In comparison to this work, we place focus on the management of the knowledge stored in the probability network and efficient and intuitive querying of this knowledge by the application space.

Identified challenges in machine learning for user modeling, according to [6], are the need for large data sets, the need for labeled data, concept drift and computational complexity. The need for large data sets amplifies the issue of computational complexity of inference, and is addressed by query optimization in PAL. The issue of data labeling is delegated to the applications on the one side and the data sources inside the vehicle on the other and is not addressed by this work. Same can be said for the concept drift, since this issue affects knowledge models which feature non-mutable user attributes. Adding and removing of new knowledge models i.e. probability networks is a task handled by PAL, but the network training and machine learning is out of its scope.

Proactive driver assistance with Dynamic Bayesian Networks has been analyzed by [14], pointing out the problems with erroneous assistance or the one which annoys the user by false positives. Setting up correct thresholds for proactive assistance would be a task for the higher-level applications.

BN is used to model and predict driver's behavior by analyzing the entire driving context in the work of [1]. A suggestion is given on the high-level model containing vehicle, environment and driver information, with the knowledge on the system history.

The work by Cou [2] has shown how BN can perform multi-sensor data fusion and emphasized the ability to explicitly model key problem features, like sensors' performance.

The work done in [10] describes, in another very relevant work on standard platform for sensor fusion, how conditional probability density functions inside a BN can be a standardized output format for all sensors and recognition algorithms. The addition of new sensors and recognition algorithms is done by manipulating a limited set of nodes, not affecting the entire system. Calculation performance has been identified as the main disadvantage, partially alleviated by modular network design. As the result of this overview of state-of-the-art, we have chosen the Dynamic Bayesian Networks as the underlying knowledge representation for PAL.

IV. Inference Method

Inference on the knowledge stored inside the Bayesian probability network can be performed by exact methods, such as the Junction Tree algorithm, or approximate methods such as Markov Chain Monte Carlo (MCMC). An overview of MCMC inference is given in [9]. The chosen method for context processing uses exact inference and avoids heuristics and approximation taking into account the safety-critical nature of tasks in a road vehicle. Knowledge uncertainty is handled with a mathematically consistent toolset, which provides repeatability of inference provided that the knowledge stays the same. Since exact inference in BNs is NP-hard, we construct the so-called Junction Tree (JT) structure around the BN. The construction of the JT is NP hard, but it is theoretically done only in two cases – when the system is initializing for the first time and when a node gets added or removed. A Junction Tree T is a tree on G constrained by three properties:

A. Family Property

For each node V in graph G there is a cluster C of T which contains the family of V.

B. Tree Property

There exists only one path between any pair of clusters in T.

C. Junction Tree Property

For any two clusters A and B of T any for every cluster C on the path in between A and B, the following property holds:

$$A \cap B \subseteq C \tag{1}$$

A Junction Tree can be constructed from the Bayesian Network in five steps, as explained in [7]:

1) Moralization

The original edges in BN are made to be undirected and additional edges are added between the parents of each node

2) Triangulation

Edges are added to the graph until it becomes chordal, meaning that every cycle with four and more nodes has a chord – an edge joining two non-adjacent nodes in a cycle.

3) Construction of the Junction Graph

Maximal cliques are found and marked as clusters for the next step.

4) Forming the Junction Tree

A tree is created by removing the redundant links in the cluster graph from the previous step. Every link between two clusters is assigned a separator.

5) Creating clique potentials

The original potential tables are used to calculate the initial clique potentials, forming the inconsistent JT. After message passing, the JT is made consistent.

V. Probabilistic Application Layer

Our contribution presented in this section builds upon the DBNs for knowledge representation and extends the Junction Tree algorithm in order to support dynamic removal and addition of nodes together with strict time guarantees on exact inference. Furthermore, we provide an application interface to the PAL which enables the HMI and ADAS applications to place complex queries while being context-processing-agnostic. The name probabilistic application layer therefore denotes the created separation layer between probabilistic reasoning and high-level application space.

A need for a separation layer between our vehicle ICT architecture on the one side, and the higher-level HMI and DAS applications on the other, arose from the project start. The HMI and ADAS applications ought to be able to place context-related queries to the vehicle with specific quality-of-service (QoS) requirements. The QoS requirements in this sense define the allowed worst-case latencies and/or data update frequencies for the query answering. This might be relevant to the ADAS applications operating with safety-critical queries, such as the ones related to pre-conditioning of the passive safety systems in a case of a known time-to-crash.

An example of such continuous and safety-relevant ADAS application query might be "Is the frontal crash imminent?", which has to be incorporated as a hypothesis into the knowledge model. Such hypothesis has to be continuously checked for truth and the result of inference has to be delivered to the application in a timely manner. In the rest of this work, we will often switch between the wording "query" and "hypothesis" when we describe the questions placed by the applications to PAL. The reason for this interchanging of terms is that an initial query, placed by an application, transforms into a node called hypothesis node once it becomes a part of the probability network. The hypothesis node can take a value between 0 (the answer to the question is 100% false) and 1 (the answer to the question is 100% true). To recap, when a query gets added to PAL, it becomes a hypothesis which is assigned a certain probability of being true.

An HMI application, for example, could use PAL to adapt the head unit or infotainment display to the user's proficiency (power user / novice) or his assessed level of tiredness. The suggested data flow for such customization of feedback, which can be directly implemented in our vehicle prototype, is given in Figure 2.

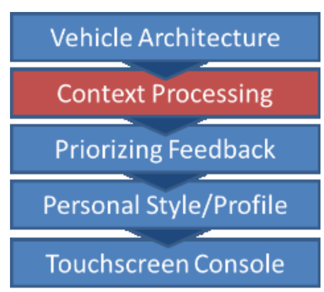

Fig. 2. Context processing can be an integral part of vehicle feedback.

Taking both examples into account, the operation of PAL and its level of abstraction inside the vehicle are shown in Figure 3.

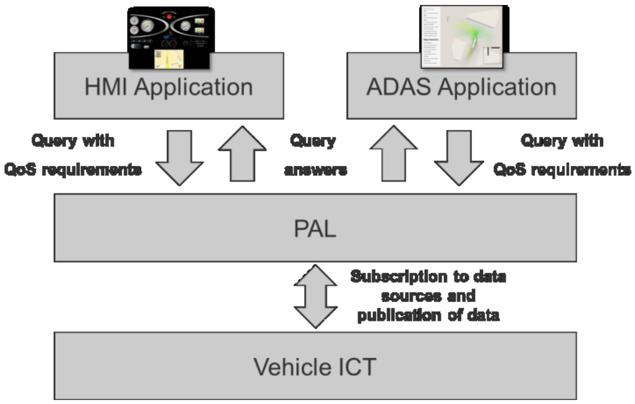

Fig. 3. PAL builds upon the vehicle ICT architecture and provides an interface to the application space.

Usage of Junction Tree for exact inference indirectly defines the JT message passing algorithm as the basis for global conditional probability tables' (CPT) normalization. We implement an extended message passing protocol (EMPP), which contains semantic description of a node, the description of the computing complexity of associated inference, as well as classical CPT data from the JT message passing protocol. The semantic description is important for the application interface, reconfiguration manager and so-called modal separation.

The nodes are internally organized into PAL modality groups, based upon the a priori knowledge of their conditional independence. As an example, a pair of feature extraction networks based on camera data might be affected by sunlight or another shared hidden variable. Even though this dependence is not captured as a dependency inside the individual feature extraction network (as it is not be important for its operation), a higher-level network can use the semantic description of nodes to take common hidden variables into account. PAL provides the description of such dependencies.

Every new data source has to be ordered into higher level clusters based on the low level ontology description. In addition, a subjective specification sheet is included. Each sensor has a specific probability of detecting an object in its field of view, depending on a set of known parameters, like the angle and distance of the object. On the higher level, however, more data is known in the system, such as weather conditions and time of day, and this data can be used to further modify the plausibility of a sensor measurement. In essence, this type of reasoning is similar to the previously described modal grouping, but it focuses on the plausibility of inferred knowledge and not on the cross-dependence.

Inner workings of PAL are shown in Figure 4 and are explained in the following.

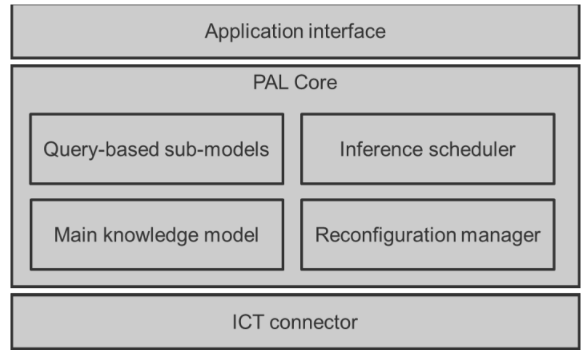

Fig. 4. Inner workings of PAL.

A. *Application Interface*

The interface towards the application space is based on XML and heavily leaned on a standard used for semantic sensor web – Open Geospatial Consortium (OGC) Sensor Web Enablement (SWE). An application can request the currently active set of hypotheses (what the PAL is already answering), the potential set of hypotheses (what the PAL knows how to answer but isn't doing it yet) and suggest new methods of combining data (new feature extraction networks). As mentione before, it can also request an answer from PAL with a fixed quality-of-service requirement and/or continuous updates.

B. *Main Knowledge Model*

The main knowledge model contains all the probability networks inside the vehicle, whose nodes can be divided into three layers:

1) *Hypothesis Layer*
2) *Feature Extraction Layer*
3) *Modality Groups*

The hypothesis layer (HL), as mentioned before, contains all the to-be-answered queries placed by HMI and ADAS applications. Inference on these top-level nodes propagates throughout the lower layers with certain degree of computational overlap, which is harnessed via JT-based dynamic programming. Answering the queries is the sole purpose of PAL, so the entire underlying structure is chosen in order to facilitate the inference.

Feature extraction layer (FEL) layer contains all the classical probability networks inside the vehicle. An example is a network for driver assessment based on camera data, or a network which stores driver's personal profile for infotainment system. This layer can be interpreted as a set of different object classification networks which are fed by underlying data sources.

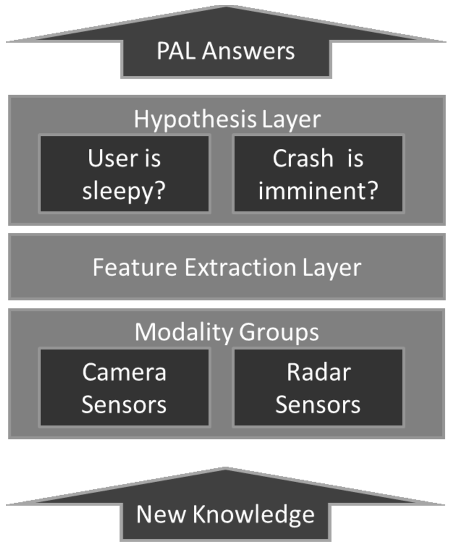

Fig. 5. Data flow through the main knowledge model

Newly acquired knowledge from the vehicle ICT architecture enters the PAL in the modality groups (MG) layer and propagates towards the hypotheses. This layer contains all the data source nodes, grouped into clusters according to their a priori assumed conditional dependence. For example, all the data sources which are stemming from camera systems would be grouped into the visual sensor group. The term modality is used here exclusively to denote a common sensor modality. Since the inference is performed on the JT query-specific sub-models, this grouping inside the DBN only becomes relevant if the group gets included into a JT sub-model.

C. *Query-Based Sub-Models*

One of the main tasks of PAL is to optimize the application queries i.e. to reduce the involved computational complexity of exact inference on large probability networks. If there are, however, no applications placing queries on the inference engine, DBNs in the main knowledge model remain the sole knowledge representation inside PAL. Only after the first query is placed (and the hypothesis layer contains at least one node) does the PAL construct a query-based sub-model in order to optimize the inference. Apart from the JT algorithm and lazy inference, PAL also references the overlapping node clusters from different JTs to reuse already inferred knowledge and accelerate propagation of new knowledge.

D. *Inference Scheduler*

The hypothesis layer can contain application queries with different criticalities. The driver assistance applications might perform time-critical inference, while the human-machine interface applications might be interested in periodical updates about the driver's assessed fatigue level. Multiple factors determine the time required to perform the exact inference, assuming equal processing power:

- Data update rate bounded by the vehicle ICT architecture
- Time to perform the knowledge normalization in the JT
- Time to perform variable elimination in the node cluster of interest

Time for knowledge normalization is mostly determined by the time necessary for upwards knowledge propagation, since the HL is located near the query-based JT root. Inference scheduler is responsible for checking the time bounds of inference on the hypothesis layer and guaranteeing that the safety-critical hypotheses are within bounds. It is priority based, with earliest deadline first within the same priority level.

E. *Reconfiguration Manager*

Apart from the continuous updating of the knowledge inside the PAL, it is possible to dynamically add or remove a certain data source, hypothesis or the in-between located feature extraction networks. The adding process has two steps: Semantic identification (what exactly is being added and how can it be used?) and performing structural changes on the DBN. Adding of a new element can be rejected, if it conflicts with the QoS requirements of existing HL, by increasing the inference time beyond the allowed margin.

F. *ICT Connector*

This component is responsible for fetching the data from the vehicle ICT architecture and pushing it into the PAL. It contains the correlations between the lowest layer of nodes in modality groups and real data inside the vehicle. In its current form, this component produces simulated data or connects to the Vires VTD simulator. In its fully developed form it should subscribe to the relevant data over the Innotrucks data-centric architecture.

VI. IMPLEMENTATION

The first version of PAL presented in this work has been implemented with the Digia C++ QT Software Development Kit and deployed on the Intel Atom based embedded computer.

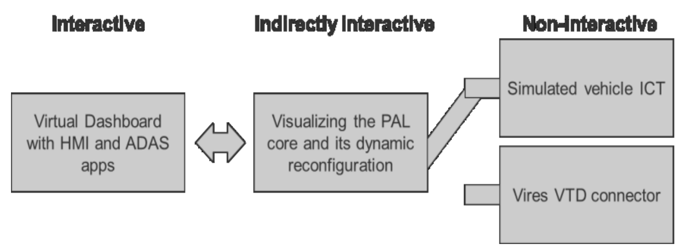

Fig. 6. Demonstrator components.

For our demonstration, we use the PAL to evaluate two hypotheses:

1) The driver is tired
2) Crash is imminent

The first hypothesis tries to determine a trend based on a longer amount of measurements; the second relies on rapid data acquisition. In addition, the first hypothesis is not bounded in time, i.e. it can be answered with *best effort*. The second hypothesis has a time bound *t* which is determined by the time necessary to react to the upcoming crash and reduce the consequences (e.g. initiate emergency braking).

The demonstrator has two modes of operation, as shown in Figure 6.

In the first mode, the user drives a vehicle in the VIRES Virtual Test Drive simulation, which also provides simulated vehicle data to PAL. A virtual dashboard is with two apps is displayed to the user. The first one analyzes driver tiredness and displays an assessment thereof. The other issues a warning if a crash is imminent. Another display provides an insight into the PAL core, in particularly visualizing the dynamic reconfiguration.

In the second mode, the user can only operate the virtual dashboard and the driving itself is fully automated and non-controllable. This is a "minimal package" of the demonstrator, used to focus only on the inner workings of PAL.

VII. Conclusion and Future Work

We provide an approach for separating the context processing from the application space and focus on inference optimization and inference real-time guarantees for dynamically changing probability networks. The approach is based on Bayesian networks which are optimized for queries over sets of hypotheses through the modified Junction Tree algorithm. We keep the structure of the tree and the time guarantees intact during addition and removal of nodes. The message passing algorithm is extended to accommodate for node description.

Future work will focus on further development of PAL as well as on knowledge exchange with other vehicles and with higher-order intelligent transport systems.

Acknowledgment

We would like to thank the International Graduate School of Science and Engineering (IGSSE) as well the Institute of Advanced Study (IAS) of the Technische Universität München for all the support in the organization of our project. Furthermore, we would like to thank fortiss, an Institut der Technischen Universität München, for all the constructive advice and practical support.

References

[1] A. Rakotonirainy, A. Maire, "Context-aware driving behavioural model", 2005.

[2] C. Cou, Th. Fraichard, P. Bessiere and E. Mazer, "Multi-sensor data fusion using Bayesian programming: An automotive application." Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on. Vol. 1. IEEE, 2002.

[3] C. Müller, G. Weinberg, "Multimodal Input in the Car, Today and Tomorrow," MultiMedia, IEEE , vol.18, no.1, pp.98-103, Jan. 2011

[4] EUROPEAN COMMISSION, "Final report of the eSafety Working Group on road safety", Information Society Technologies, 2002.

[5] G. Geiser, "Man Machine Interaction in Vehicles", ATZ 87, 1985. pages 74 - 77.

[6] G. I. Webb, M. J. Pazzani, D. Billsus, „Machine Learning for User Modeling", 2001, Volume 11, Issue 1-2, pp 19-29

[7] H. Pan, D. McMichael and M. Lendjel, "Information Fusion, Causal Probabilistic Network And Probanet II: Inference Algorithms and Probanet System.", 1997.

[8] J. Pearl, "Probabilistic reasoning in intelligent systems: networks of plausible inference", 1988.

[9] K.M. Hanson and G. S. Cnnningham. "Operation of the Bayes inference engine." Fundamental Theories of Physics", 105 (1998): 309-318.

[10] N. Kawasaki, and U. Kiencke, "Standard platform for sensor fusion on advanced driver assistance system using bayesian network." Intelligent Vehicles Symposium, 2004 IEEE. IEEE, 2004.

[11] Rockl, M.; Robertson, P.; Frank, K.; Strang, T.; , "An Architecture for Situation-Aware Driver Assistance Systems," Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th , vol., no., pp.2555-2559, 22-25 April 2007.

[12] Wong, S. K. M., and C. J. Butz. "A Bayesian Approach to User Profiling."G. Geiser. Man Machine Interaction in Vehicles. ATZ 87, 1985. pages 74-77.

[13] X. Li, Q. Ji, "Active affective State detection and user assistance with dynamic bayesian networks," Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on , vol.35, no.1, pp. 93- 105, Jan. 2005

[14] X. Li, Q. Ji, "Active affective State detection and user assistance with dynamic bayesian networks," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* , vol.35, no.1, pp. 93- 105, Jan. 2005

[15] Yoshiyuki Umemura. Driver Behavior and Active Safety. In R&D Review of Toyota CRDL - Special Issue : Driver Behavior and Active Safety, 2004.