Layer-Based Multi-Sensor Fusion Architecture for Cooperative and Automated Driving Application Development

Maurice Kwakkernaat, Tjerk Bijlsma and Frank Ophelders

Abstract. Development of current ADAS is focused on single functionality and independent operation. Development of next generation cooperative and automated ADAS applications requires large amounts of information to be combined and interpreted. To operate efficiently and effectively, these applications should not operate in isolation, but share resources, information and functionalities. Furthermore, development, prototyping, real-life testing and evaluation of the applications in multiple-vehicle scenarios becomes more complex. In this paper iVSP, a scalable, multi-sensor fusion and processing architecture, is proposed for efficient development, prototyping, testing and evaluation of cooperative and automated driving applications in small to medium scale pilots.

Keywords: Advanced driver assistance systems, multi-sensor fusion, layer based architecture, vehicle automation, cooperative driving, sensing and perception, prototyping.

1 Introduction

Next generation intelligent vehicles will incorporate more complex functionalities, sensors and wireless communication to interact with drivers and the environment by performing automatic and cooperative driving tasks. This is driven by the continuing trends towards improved vehicle and traffic safety, supported by updated vehicle rating schemes [1], and the need for improved traffic flow and reduced CO2 emissions. To obtain a cost effective and reliable implementation these applications should not operate in isolation, but rather share processors, resources, information and functionalities. The development of these complex applications requires challenging solutions, e.g. multi-sensor information fusion, parallel real-time processing, security, functional safety and real-life evaluation.

M. Kwakkernaat(✉) · T. Bijlsma · F. Ophelders
TNO, Steenovenweg 1, 5708 HN Helmond, The Netherlands
e-mail: {maurice.kwakkernaat,tjerk.bijlsma,frank.ophelders}@tno.nl

J. Fischer-Wolfarth and G. Meyer (eds.), *Advanced Microsystems for Automotive Applications 2014*, Lecture Notes in Mobility,
DOI: 10.1007/978-3-319-08087-1_3,

It is well recognized that the development of advanced automated driving and cooperative applications, or more generally advanced driver assistance systems (ADAS) applications, requires an architecture and platform with increased functionality, performance and interoperability compared to the current available sensor and processing platforms [2-6]. In [4] a server-client architecture was developed that consists of a local dynamic map (LDM) database server, a LDM or data fusion producer and client processes. The proposed architecture was developed to support rapid development and efficient use of resources, while decoupling algorithms and application functionality from low-level interfaces. In [5] a functional architecture was defined based on a perception layer, command layer, execution layer and HMI layer. This architecture allows effective and easy introduction of new functions compared to traditional architectures. In [6] an approach for the integration of ADAS functions was proposed based on an unified perception layer, a decision layer and an action layer. Although perception is integrated, parallel applications with independent reasoning are still allowed.

The work in this paper presents a scalable, multi-sensor fusion and processing architecture called Intelligent Vehicle Safety Platform (iVSP). The approach builds on previous work with the focus on prototyping, testing and evaluation of cooperative and automated driving applications in small to medium scale pilots. This includes the ability to efficiently develop and evaluate new applications in real-life. The objective is to define and implement a layer-based software architecture with appropriate interfaces and security that allow to collect, process and exchange information from multiple sources, such as external and internal sensors, wireless communication and digital-map data. The information is processed to generate real-time situational awareness (e.g. an LDM). This information can be made available to multiple parallel applications. The layer-based architecture decouples applications from low-level interfaces and enables developers to implement application logic relying on information and its confidence instead of low-level data from sensors. Based on the defined architecture an unique test and demonstration vehicle was instrumented for developing and evaluating cooperative and automated ADAS applications. The technology is integrated into a Toyota Prius and uses ITS-G5 communication, radar and camera sensors, information fusion, object tracking and control algorithms.

The paper is organized as follows. In Section 2, the iVSP system is described, including the general architecture and the layer-based structure. In Section 3, the results of two applications that are developed using the iVSP architecture are presented. Finally, Section 4 summarizes the work and provides future work.

2 System Description

2.1 Architecture

iVSP is an in-vehicle software architecture based on interconnected layers. The five layers are visualized in Figure 1 and consist of a sensor layer, communication

layer, information layer, application layer and interface and authorization layer. The reason to adopt a layer based design is to adapt an object oriented approach in setting up ADAS. The layer functionalities can be reused when new ADAS applications are implemented in the application layer. The sensor and communication layer provide information to the information layer, which provides information to applications running in the application layer. All layers are interconnected using facilities from the interface and authorization layer.

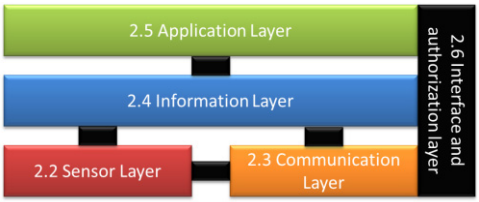

Fig. 1 iVSP layers

For the sensor layer a flexible approach has been chosen that enables the addition of sensors and interpreters for sensor data. The sensor layer performs preprocessing, filtering and low-level sensor fusion. A separate communication layer exchanges and interprets information coming from multimodal V2X wireless communication between other platforms or road-side units. Communicated sensor-based information will likely be provided to the sensor layer, while other types of information will be passed to the information layer. In the information layer the processed information can be stored in the form of *permanent static*, *transient static* and *transient dynamic* information. These different forms of information can be made available to authorized applications in the application layer at a low latency. The information layer also takes care of removing old non-relevant information using a garbage collector. In the application layer, different parallel applications can be executed.

In each layer specific functions can be defined, e.g. reading raw sensor data and interpreting the information should be handled by the sensor layer in order to pass it to the information layer. Some functions cannot be explicitly assigned to a specific layer, for example risk estimation can be assigned to either the information or application layer. In this specific example, the approach would be to run the function in the application layer when it is application specific, but when this function is requested by multiple applications it should run in the information layer to make it available for multiple applications.

Applications executed in the application layer are authorized to access certain information. To protect information, identification and security measures have been applied, such that not all applications have access to all collected information. Furthermore, for wireless communicated information certificates and basic encryption are used.

Some ADAS applications require a bounded latency for the retrieval of sensor values and predictable behavior. Therefore, iVSP has ongoing activities to apply real-time scheduling for critical parts of the layers.

In the next section the five iVSP layers and their functionalities will be detailed.

2.2 *Sensor Layer*

The sensor layer has been set up using sources, sinks between which information flows, via message interpreters, message mergers and sensor fusion processes, as depicted in Figure 2.

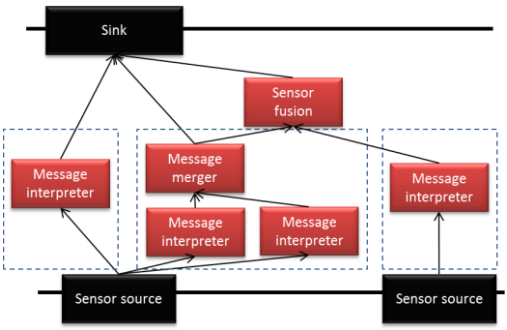

Fig. 2 Schematic drawing of the sensor layer

The sensor layer has been setup using a data flow structure. It is an acyclic event based system, where information flows from sources to the sink. This layer lends itself perfectly for parallel execution, for which the Intel thread library is used.

In the sensor layer, multiple sources are used, e.g. to receive CAN (Controller Area Network) frames and GPS NMEA string. The source receives the raw sensor information and forwards it to an available interpreter, e.g. for a CAN frame the source selects the interpreter corresponding to the ID of the CAN frame.

Interpreters extract sensor information from the raw sensor data and scale them according to the sensor specification. If sensor information is provided in multiple chunks, e.g. multiple CAN frames, a message merger is used. Additionally, it may be possible to significantly improve sensor information by combining it with information from other sensors, i.e. sensor fusion. An additional benefit of sensor fusion is that often the sensor values that serve as input for the fusion become obsolete, thus the amount of data in the system is reduced.

The sink receives sensor information from the interpreters, message mergers and sensor fusion processes. Currently a sink is used to forward information to the information layer. Additionally, sinks can be added to send information to different platforms or layers, e.g. to send information to external platforms.

The source, the sink, message interpreters, message merger and sensor fusion processes have been setup in a generic form. These can be easily adapted towards a new sensor or information. For example, the basic message interpreter class can easily be configured to add a new decoder for a specific CAN frame for which the values are scaled and decoded according to a dbc specification.

2.3 *Communication Layer*

The Communication layer has been set up using a communication source, message encoders and decoders and the observer pattern, as depicted in Figure 3.

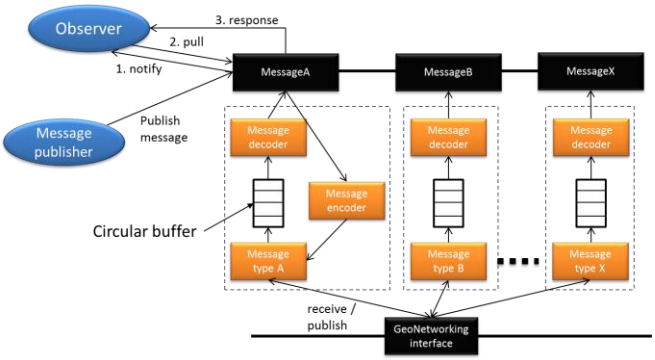

Fig. 3 Schematic drawing of the communication layer

In the current setup, communication is based on ITS-G5 with GeoNetworking, standards for vehicle to vehicle and road side communication. Multiple message types are supported and can be added to the communication layer. Processes in the information layer can register themselves as *observers* of a specific message type, meaning they will be notified if new messages of that type have arrived. Upon a notify the observer decides whether to pull the data, which forces the message decoder to parse the message and provide the observer with the decoded message. For efficiency reasons a message is only decoded when it is pulled by an observer, and the decoded message is kept in a cache to efficiently serve other observers.

Raw messages that are received through the GeoNetworking interface are put in the message buffer corresponding to the message type. In iVSP a circular buffer with a configurable size is used.

Processes in the information layer and application layer are allowed to publish messages through the communication layer. A message is provided at the interface of a specific message, which calls the message encoder to create the raw messages that are transmitted over the GeoNetworking interface.

The framework with message encoders and decoders and their source is setup in a generic form. New message types or network interfaces can easily be added. The architecture allows for addition of cellular communication.

2.4 Information Layer

The information layer stores and provides information for and to processes in all iVSP layers. Figure 4 depicts the information layer, which has been setup with the focus at three main activities: information storage, information provision and information management. These three main activities operate around the information storage that is centralized in iVSP.

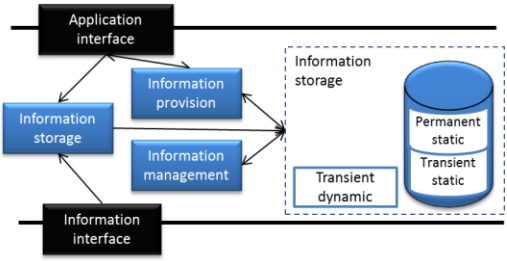

Fig. 4 Schematic drawing of the information layer

The information storage in iVSP uses a LDM, similar to the ETSI description in [7]. The iVSP LDM contains *permanent static*, *transient static* and *transient dynamic* information. Permanent static information includes information that rarely changes, e.g. map data and system configuration data. The transient static information has a life span from approximately a minute until hours or even days, e.g. signal phase and timing messages and road works warnings. Transient dynamic information has a short life-span, typically the most recent sensor layer information. Note that depending on the application of iVSP, the storage of information is determined. For example, the DPSI application in Section 3.1, requires sensor information to be analyzed for safety-critical events (SCEs), thus sensor information is stored as transient static, rather than transient dynamic.

The information storage activity stores the information received from the interfaces. The information provision retrieves requested information from the storage, for which it verifies the authorization with the interface and authorization layer. The information management process periodically reads the stored information, discards outdated information and in the future may be configurable to aggregate certain types of stored information and if necessary reallocate it. The information storage for transient dynamic information is implemented in memory, to minimize the latency and since volatile storage is sufficient. For both permanent static and transient static information a MySQL database is used. Considered alternatives include SQLite, Berkeley DB, Firebird and Redis. MySQL was chosen, due to its large user base and support. However, the choice may have to be reconsidered in case predictable behavior becomes desirable for the database.

2.5 Application Layer

In iVSP applications can be developed without knowledge of the specific sensor set used. Applications can obtain information from the information layer, which enables the application developer to use accurate information obtained by sensor fusion without having a detailed understanding of the applied algorithms.

Furthermore, applications execute independently of each other, and can be executed on different physical platforms (e.g. human machine interfaces and applications can be decoupled).

2.6 Interface and Authorization Layer

The interface and authorization layer provides means for the communication between layers and security, authentication and authorization mechanisms.

For the communication between layers and the message format specification, we use Apache Thrift, due to its flexibility. Data types and service interfaces are specified from which the Thrift framework can generate code, with which clients and servers can be setup that interact using remote procedure calls (RPCs). We use Thrift to generate code in C++, Java, Python and JavaScript. Thrift has been extended with support for the observer pattern. Alternatives considered besides Thrift were Lightweight Communications and Marshalling (LCM), Google Protocol buffers and the MPI framework, these were not selected, amongst others due to their maturity and absence of code generation for communication.

For the secure communication and authentication between layers, public key cryptography is applied. Layers and applications can register a certificate at the interface and security layer, which maintains a repository with certificates. Additionally, this layer manages an access control matrix, which lists the information for which a certificate is authorized. This matrix can be consulted by an information provider to determine if information can be shared. For the cryptography the Keyczar toolkit is used, which supports authentication and encryption and cab be used in multiple programming languages.

3 Results

3.1 Driver Behavior Monitoring Application

iVSP has been used to perform a study to monitor driver behavior [8], based upon basic vehicle sensors. iVSP was applied to monitor driver behavior, in order to determine the role of a driver during SCEs. The goal of this study was to quantify if a driver was distracted for SCEs and to determine if the event occurred due to a driver error or due to an external event.

For this study iVSP was applied in a passenger vehicle, where the vehicle sensors and a mono-camera where connected to iVSP. The sensor layer of iVSP was

configured to receive and interpret the sensor information from the vehicle sensors and the mono-camera and forward it to the information layer. The information layer stored the information and provided it to the driver safety performance indication (DSPI) application running in the application layer. The DSPI application estimates the distraction of a driver, based upon the lane-keeping and car-following behavior and in case of a detected SCE sends the estimated distraction to the information layer for storage. The information layer has been configured to store relevant sensor information and the results of the DSPI application, which enabled replaying and analyzing scenarios with SCEs.

The DSPI application was evaluated in a simulator and applied in an instrumented vehicle. In a driving simulator experiment with 18 drivers the DSPI application was evaluated. The application correctly predicted distraction in 79% of the car-following detections and in 85% of the lane-keeping detections on highway sections. The application was demonstrated with an instrumented vehicle, where the DSPI application was executed by iVSP. In the instrumented vehicle, approximately 10 hours of driving behavior and corresponding SCEs were stored. On highways and rural roads, 84 SCEs were detected, from which 80% was due to distractions. Because the DSPI algorithms are based upon lane-keeping and car-following, the results of this study can be applied to reduce the number of false positive warnings for lane keeping assist and forward collision warning systems. iVSP has been applied to validate the DSPI algorithms in real-life at the road. For follow-up studies, it enables low-cost up-scaling to a large fleet for evaluation of DSPI algorithms.

3.2 Cooperative Automated Emergency Braking Application

Based on iVSP a cooperative AEB system (C-AEB) was developed with the focus on improved bicycle collision avoidance in crossing scenarios and beyond line-of-sight and beyond field-of-view [9]. The objective is to identify a bicycle object in the direct vicinity of the host vehicle and to warn the driver several seconds before a collision is expected (visual, audible, seatbelt pre-tensioner) and actuate the vehicle by automated braking to prevent a collision. To identify the bicycle the vehicle is equipped with two on-board sensors (radar, camera) and a “virtual” sensor (ITS-G5 wireless communication). The bicycle motion data is transmitted wirelessly after its state is estimated by information fusion and tracking in the so-called Host tracking function at the bicycle. The host vehicle is also equipped with a Host tracking function that is used to calculate the vehicle state, which is used by the Object tracking and Risk estimation. The radar, camera, communicated bicycle motion data and vehicle motion data are input to the Object tracking, which fuses these types of information in order to create reliable bicycle object information that holds the relative bicycle motion state with respect to the vehicle. The bicycle object and vehicle motion information, in turn, serve as input to the Risk estimation function that determines the risk level and time-to-collision, the parameters which are used in the actual C-AEB control algorithm.

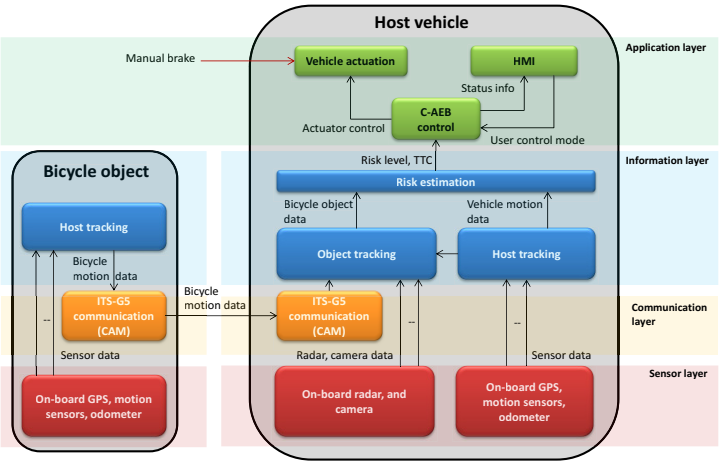

Fig. 5 Functional architecture of the C-AEB application based on iVSP

4 Summary and Outlook

This paper presents iVSP, a layer-based multi-sensor fusion architecture for rapid development of cooperative and automated driving applications. The approach collects, processes and exchanges information from different sources to generate real-time situational awareness, suitable for the execution of ADAS applications. iVSP has successfully been applied in two applications: driver behavior monitoring and cooperative automated emergency braking. An interesting addition considered for iVSP is the distribution of layers over multiple systems.

References

[1] Schram, R., Williams, A., van Ratingen, M.: Implementation of Autonomous Emergency Braking (AEB), the next step in Euro NCAP'S safety assessment. In: ESV, Seoul (2013)
[2] Valldorf, J., Gessner, W.: Advanced Microsystems for Automotive Applications. Springer, Heidelberg (2006)
[3] Meyer, G., Valldorf, J.: Advanced Microsystems for Automotive Applications. Springer, Heidelberg (2010)
[4] SAFESPOT consortium, Deliverable D1.3.4: Hardware and Software Platform Public Specification (2008)
[5] HAVEit consortium, Deliverable 12.1: Architecture (2009)

[6] Amditis, A., et al.: A Holistic Approach to the Integration of Safety Applications: The INSAFES Subproject Within the European Framework Programme 6 Integrating Project PReVENT. IEEE Trans. Int. Trans. Sys. 11(3), 554–566 (2010)
[7] ETSI, Local Dynamic Map (LDM); Rationale for and guidance on standardization. ETSI TR 102 863 V1.1.1 (2011)
[8] Van Rooij, L., et al.: Driver behaviour monitoring based on vehicle sensors for safety-critical event logging. In: FISITA 2014 (to appear, 2014)
[9] Kwakkernaat, M., et al.: Cooperative automated emergency braking for improved safety beyond sensor line-of-sight and field-of-view. In: FISITA 2014 (to appear, 2014)