DESH-G model and Preliminary Architecture for Multiple ADAS system

Masao Ito 1)

1) NIL Software Corp.
   2-17-7 Kinuta, Setagaya, Tokyo, 157-0073, Japan (E-mail: nil@nil.co.jp)

Received on October 12, 2017

ABSTRACT: Various Advanced Driver Assistance Systems (ADAS) become prevailing recently. In this paper, we argue the safety analysis of combined ADAS systems. The definition of ADAS is vague, so we obey the definition of Code of Practice for the Design and Evaluation of ADAS (CoP) 1). In this definition, the ADAS have to be active, that is, it "provide(s) active support for lateral and/or longitudinal control". There is a technical report that provides the consolidation way of various warnings emitted by the several ADAS systems (ISO/TR 12204 2)). However, the problem is not limited to the warning signals. We have to consider the controller of ADAS because it is actively controlled from its definition. We need the more integrated way, and we provide a way to handle this situation by using the idea: DESH-G model.

KEY WORDS: information, communication, and control, ADAS, DESH-G, HMI, Preliminary Architecture [E1]

1. Introduction

Recently, the Advanced Driver Assistance Systems (ADAS) becomes in common, and warn the driver of hazard and highly supports his/her driving task. For example, Active Lane Change Assist (ALCA) 3) supports the lateral movement of a car by a simple user operation.

There is little discussion about the conflicts of controllers among several ADAS systems. So far, there are many arguments about the safety of single ADAS in the car, but we have to consider the consequences of several systems of ADAS. We mean that the traditionally the driver is the only manoeuvring the car, but nowadays, driver and several systems are controlling the car. Certainly, there is the technical report ISO/TR 12204, but it just discusses the consolidation of warning signals, not the controller.

1.1 The complicated situations with ADAS

DOT HS 810 905 4) reports about the situations that the car equipped with several driving systems encountered. Moreover, it says the ways of consolidation of various warning/alarm regarding the human-machine interface. Based on this report we also create a new scenario with three ADAS systems: ACC, CWB and ALCA.

Adaptive Cruise Control (ACC) system: if a vehicle with a lower speed is in front of the car the ACC will respond with a vehicle deceleration to not exceed a pre-set distance to the preceding vehicle.

Collision Warning and Brake (CWB) system: if it detects the possibility of collision, it provides a warning to the driver and takes action autonomously without any driver input (by braking or steering or both).

Active Lane Change Assistance (ALCA): with it activated, whenever the driver holds the turn signal the car will automatically change lanes.

1.2. A scenario as an example

The scenario using those ADAS systems is:

- Step 1: Subject vehicle automatically follows the lead vehicle by ACC.
- Step 2: Lead vehicle changes lanes and the ACC becomes the constant speed mode (figure 1 upper part).
- Step 3: Subject vehicle finds the block subject, and CWB warns it (ACC becomes OFF mode).
- Step 4: The driver of subject vehicle activates the turn signal and ALCA is activated.
- Step 5: The adjacent vehicle is in the blind spot of the left lane, ALCA reacts it (figure 1 lower part).

Fig. 1 An example scenario

In step 5, ALCA might force the car back to original lane to avoid the crash with the adjacent vehicle. At the same time, CWB might activate brake because of approaching the obstacle and want to move left to avoid collision with it. So, there might be a conflict between two ADAS systems.

2. The complicated Situation with ADAS

The conflicts of warning signals among several systems in a car are depicted in the ISO/TR 12204. This is a supplement of ISO/TR 16362 for the "MMI of warning systems in vehicles". It provides the approaches "for determining if integration is necessary to mitigate the possibility that signals from one or more
vehicle system may degrade the driver's comprehension of, or response to, safety critical warning signals from another system(s)" and "methods for assessing potential integration conflicts".

The technical report covered the conflicts of the warning signals, but we would like to think the conflicts among the ADAS controller in a car.

First of all, we define the ADAS by using several conditions: the Code of Practice for the Design and Evaluation of ADAS (CoP) \(^\text{(5)}\) identifies five conditions for ADAS. ADAS has to satisfy all conditions below:

- [C-1] support the driver in the primary driving task
- [C-2] provide active support for lateral and/or longitudinal control with or without warning
- [C-3] detect and evaluate the vehicle environment
- [C-4] use complex signal processing
- [C-5] direct interaction between the driver and the system

The three systems in the previous scenario satisfy all conditions. By using these criteria, we can precisely define whether a system is ADAS or not. For example, the Blind Spot Monitor (BSM) system is not ADAS, because it only provides warnings to the driver (not [C-2]). The Active Front Steering (AFS) is not ADAS because it does not use the environmental information (not [C-3]). The Rain Sensor System is not ADAS because it does not support the primary task of the driver (not [C-1]).

With this CoP ADAS definition, we can get the new framework to consider the HMI and preliminary architecture in the concept phase \(^\text{(5)}\) (this phase is the same phase of ISO 26262 \(^\text{(6)}\) part three). We call this framework as DESH-G (Driver, Environment, Software, Hardware and Goal, figure 2), which is the HMI model in Meta level \(^\text{(7)}\).

3. DESH-G

In this paper, we focus on the environment model and the driver model of the DESH-G framework. Other elements are software (controller), hardware (e.g., actuator, sensor) and the goal of the driver.

3.1 Environment model

We use the Situation-Scenario Matrix (SSM) \(^\text{(8)}\) to express the environment around the car. This matrix has two axes. The one indicates the element of the environment, and the other is the time sequence. The former has the element's type showing below:

- Road type (rural, freeway, arterial...)
- Road surface (flat, dry asphalt...)
- Neighbouring cars (type of cars forward / backwards / side...)
- Traffic condition (congestion level)
- Non-vehicle actor (bike, pedestrian...)
- Weather condition (sunny/rainy/snowy...)
- Visibility
- Traffic rules (speed limits, traffic signs...)

What is to be the environmental element depends on the target system to be analysed. For example, if a system is the Cooperative Adaptive Cruise Control (CACC) \(^\text{(5)}\), we have to add the communication state into the above element list of the environment. And if a system is the Parking Assist System (PAS), we might eliminate the road types and traffic conditions, and add the lock plate as the SSM element.

3.2. Driver model

In this paper, the primary purpose of the driver model is to analyse the hazardous situation. So, we adopt the task capability interface model \(^\text{(10)}\). If the capability of the driver is under the task demand required, the driver cannot drive the car anymore.

We divide the ability of the driver into the driver skill and the driver state \(^\text{(7)}\). The driver skill is the ability to perform a given task, and this skill doesn't change in the short term. However, the driver state easily varies by the various factors. For example, the insufficient sleep decreases the level of his state, and it affects the controllability. This variability doesn't come from the only somatic problem. If an urgent situation (for example, the driver has to go to school to pick up his/her child, but he/she doesn't have enough time to make it) occurs, the state of the driver also varies.

If the driver's ability is low and the task demand is high, the situation might be dangerous. Moreover, the environment affects the task demand. For example, if we have to drive in the rainy
night on the non-asphalt road, the task demand is high compared with running in the fine daytime on a highway.

In our approach, to calculate this task demand, we use the SSM, which shows the driver's situation at a particular time. So, we can calculate the change of task demand value in a scenario. We already proposed several formulas to compute this task demand and the driver's ability (7).

Of course, there are various drivers, and more the car type driven by her or him affect the task demand. However, it helps us consider hazardous situations relatively.

4. HMI Abstract Model

In this chapter, we think about the HMI abstract model based on the previous arguments about the DESH-G model. Especially, we will focus on two types of interfaces: One is the interface between the system and the driver; another is the interface between the system and the environment.

4.1. Interfaces with the Driver and the Environment

Figure 3 shows the correspondence between the deformed DESH-G model and the interface classes.

![Fig. 3 DESH-G with Interface Classes [model: S0]](image)

We identified the five interface classes. There are three interface classes between the driver and the system (controller) and also the two between the system and the environment. We explain them respectively.

i. Driver Information interface class (I_i)

By this interface, the system provides information to the driver. For example, traditionally we have various instruments like the speedometer and several warnings and alerts. If the car has the new functionality, it will give us another type of information. For example, the vehicle that has the adaptive cruise control (ACC) system has information of the distance between forwarding car and the subject car. The system knows information through the Is interface, and this provided information is the useful one for the driver.

ii. Observation information class (I_o)

Currently, we have another type of information flow from the driver to the system, not from the system to him/her. In the cockpit, there are sensors that observe the driver, and the system changes the behaviour. For example, in the driver monitoring system, the system has the camera to check the driver status and warns him by light or sound if it estimates that the driver status shows the low performance.

iii. Manoeuvring information class (I_m)

This interface is relating to the driver's intention for the longitudinal and lateral movement of the car by the handle, the accelerator, the brake pedal and so on. We had only the mechanical interface, but nowadays those are partially supported by electric/electronic parts and software.

iv. Environment information interface class (I_s)

This class and next one is the interface between the system and the environment. Mainly we can get the data of the environment around the car with the visual or radar sensors. Recently we also can get the information via other cars (V2V) or the infrastructure (V2I).

v. Driving interface class (I_d)

This interface relates the environment directly. For example, the tires transmit the driving and side force to the ground. The car light illuminates the front area.

4.2. HMI Abstract Model

The HMI abstract model is the model that is described by the previous interface class, and it doesn't include the other properties like usability, the device type and so on. This model is just the basic one, but we think this is useful. When we have to think of the new kind of HMI, especially, in the ADAS area, it is not simply covered by the existing HMI techniques (11). We can analyse the new HMI by the abstract model.

To explain how to use those interface classes, we use an example: CACC system. In the CACC system, the system is not closed in a single car. It communicates with other vehicles. The information from the other vehicle is useful: in the conventional auto-cruise control (ACC) system, the system has to calculate the relation between the following vehicle and subject car by using the camera or the millimetre-wave radar. It might not be accurate because of bad weather condition. Also, it sometimes has the delay in the calculation of distance. However, the CACC system can get the information about the actions of the driver of the following car almost simultaneously. So, the subject car can follow the forward car accurately and timely.
We can convert the original model [S0] to express the CACC system. Figure 4 shows the CACC model. The new model “S1~S2” includes the symbol ‘~’, and this denotes the communication between the subject car (S1) and the forward car (S2). In this figure, the left side shows the subject car, and the right side is the forward car. After establishing the communication link, the subject car can get the information of the forward vehicle (and vice versa). Those are status (Ii’) and operation (Im’) of the other driver, and sensory information (Is’) and driving information (Id’) of the lead car.

4.3. DESH-G and condition of ADAS

The conditions of ADAS endorse our DESH-G model. The conditions, [C-1] and [C-4] are relating to the “Driver” (see 3.2). ADAS supports the primary task of the driver and has the direct interaction with the driver. The interface Ii presents the current situation and receives the input of the driver (for example, the driving mode). Moreover, the driver gives the information for manoeuvring through the interface Im [C-3] requires getting information on the “Environment” (see 3.1). That is, when we consider the ADAS, we have to include the driver and environment in the model like DESH-G model.

Next, we think, the preliminary architecture of the system. In ISO 26262, the end of the concept phase, we have to create the preliminary architecture.

5. Preliminary Architecture

In the concept phase, we create the preliminary architecture, as the ISO 26262 standard requires it. We identify two points: First, the need for arbitration in the control of systems. The second is the monitoring of the driver. Through those arguments, we can identify the minimum requirements to design the preliminary architecture.

5.1. Arbitration of control

According to the CoP conditions, ADAS provides the active support for lateral control, longitudinal control, or both. If there are several ADAS systems, we consider the arbitration of those controls, especially in a critical situation. We can select several measures to consolidate them.

The simple way is the prioritisation, that is, we choose one of ADAS systems, and we cancel others. In the previous scenario, for example, if ACC is on and CWB finds the obstacle, ACC will be cancelled. ACC and CWB system is both the longitudinal control and usually CWB has high priority from the viewpoint of safety (Step 3).

In the last part of the scenario (Step 5), the situation is not so simple. We cannot easily decide which system has to be cancelled. CWB does the longitudinal control, but ALCA does the lateral control. Moreover, the purpose of both systems is the avoidance of collision.

We could cancel both systems remaining to emit the warning signals. Moreover, the decision is transferred to the driver. In this case, the task demand of the driver is high because the driver has to drive a car again abruptly in a critical situation.

The other way is finding the new path to escape this critical situation. For example, automatically the system could accelerate the car and move to left lane without colliding the adjacent vehicle and also avoid the obstacle on the current lane. Of course, it is tough to decide this is possible or not and both system, CWB and ALCA need some collaboration, or we need the other system to do this.

In any case, we need arbitration among systems.

5.2. Monitoring of the driver

We have been modelling of the driver, and this is important for the safety analysis. When we allocate the Automotive Safety Integrity Level (ASIL) in the concept phase of ISO 26262
standard, we have to decide the value of controllability of the driver in a critical situation, that is, occurring of the hazardous event.

We can think the controllability is a margin between the driver capability and task demand. If the task demand exceeds the driver capability, we cannot drive anymore.

The driver capability has two attributes: the driver state and the driver skill. The driver state also has two facets: the physical state and the psychological state. For example, if the driver is tired, his physical state is low. If she is depressed, her psychological state is low.

The task demand is the required efforts for the driver in a situation. Various situations surrounding a car need different task demand. For example, if we drive on a rainy night, the value of task demand is high compared with taking a drive in the bright daytime.

First, we calculate the task demand of the route segment. The road segment is the section where we can assume that the environment is almost the same. On the highway, the road segment is long. In the city it is short. Total task demand is given as the summation of every route segments that are needed to achieve a goal of the driver.

In figure 5, we show the relationship between the task demand and the driver capability. If a driver capability is lower than a task demand, the possibility to meet harm is high.

What the driver facing critical situation can do is not only the task demand that is calculated from the information of environment but the driver status, which is got through the interface class Io (c.f. 4.1).

5.3. Preliminary archetype pattern

By using the information from sensor interface (I_s) and manoeuvring interface (I_m), we can calculate the value of task demand.

There are two ways to estimate the driver capability. First is the direct estimation through driver monitoring interface (I_o), for example, by using the driver drowsiness detection. The other is the indirect estimation through manoeuvring information (I_m). For example, if the system detects the lane marker and also the zigzag style driving, the system can estimate the low driver capability.

The other characteristic is the arbitrator of controllers. The outputs of each ADAS and the results of the driver capability estimator are inputs of the arbitrator. Aforementioned (5.1), the arbitrator calculates the final output for the information interface (I_i) and the driving information interface (I_d).

6. Information Display

In this chapter, we show an example of information display (N.B.: it is the conceptual one, so the colour and shape does not have deep meaning here). There are several ADAS systems: CACC (including CWB) and ALCA. In figure 7, we show the interface classes: I_o, I_m, I_s and I_d. Those are corresponding to figure 3. Interface class li corresponds to this information display.
The scenario is almost the same as the previous one. Now, the subject car follows the lead car. The lead vehicle finds the obstacle and will change the lane. This obstacle information transfers to the subject car through a communication link. So, the car automatically changes the lane following the lead car. However, there is a quickly approaching car on the left lane. The ALCA warns, and provides the recommended driving goal.

Briefly, we explain the conceptual model of information display. In the centre, the human-shaped icon shows the driver state that comes from $I_D$ or $I_M$ (c.f. 5.2). The surround circles of the human shape indicate the state of the car. The interface class, $I_D$ and $I_M$, provides that information. Both sides show information about the environment. In the CACC system, we can get the information from another car, we also can see on the information display.

The aim-like square on the upper left shows the driving target that will appear when the conflict of controlling occurs. It provides the suggestive mark. The horizontal direction indicates the lateral movement; the vertical direction means the longitudinal movement. Those marks are the simple indication to the driver in the emergency mode.

![Conceptual information display](image)

**Fig. 7 Conceptual information display**

### 7. Conclusion

In this paper, we describe the design of the preliminary architecture of the car that has several ADAS systems. According to CoP conditions, to design the ADAS, we need the driver (model) and the environmental (model). So, we create the DESH-G model that use the driver model and the environmental model. We can identify the several interface classes in the DESH-G model. ADAS, in the sense of CoP, requires the arbitration of control, needless to say, with the consolidation of warning signals.

Our explanation is about the development of new ADAS systems simultaneously. However, usually, an ADAS is developed to work with other existing ADAS systems. In the near future, we would like to show the mixed development of multiple ADASs. We think that DESH-G and the extension of interface classes can solve it.

This paper is written based on a proceeding which is presented at Fourth International Symposium on Future Active Safety Technology Toward zero traffic accidents (FAST-zero 17).

### References

1. Knapp, Andreas, et al.: Code of Practice for the Design and Evaluation of ADAS., Preventive and Active Safety Applications, eSafety for road and air transport, European Commission Integrated Project, Response 3 (2009).
2. ISO, ISO/TR 12204. Road vehicles -- Ergonomic aspects of transport information and control systems -- Introduction to integrating safety critical and time critical warning signals, ISO, (2012).
3. Gerard Jude Castillo, TopGear Philippines, http://www.topgear.com.ph/news/technology-news/how-does-the-active-lane-change-assist-work (Accessed 2018.11.1)
4. DOT HS 810 905: Integrated Vehicle-Based Safety Systems (IVBSS): Human Factors and Driver-Vehicle Interface (DVI) Summary Report, (2008).
5. Masao Ito: How can we deal with the concept phase in the functional safety standard for automobiles?, in Proceedings of the Twenty-fourth Safety-critical Systems Symposium, Brighton, UK (2016).
6. ISO, ISO 26262. Road vehicles - Functional safety -, ISO, (2011).
7. Masao Ito: Controllability in ISO 26262 and Driver Model. In European Conference on Software Process Improvement. Springer International Publishing, pp. 313-321 (2015).
8. Masao Ito: Finding threats with hazards in the concept phase of product development. In European Conference on Software Process Improvement, Springer Berlin Heidelberg, pp. 277-284 (2014).
9. Gerrit Naus, et al.: Cooperative adaptive cruise control. in IEEE automotive engineering symposium Eindhoven, The Netherlands, (2009).
10. Ray Fuller & Jorge A. Santos: Psychology and the highway engineer, Human factors for highway engineers pp.1-10 (2002).
11. Lucy Suchman: Human-machine reconfigurations: Plans and situated actions. Cambridge University Press, pp.161-162 (2007).
12. Arlow, Jim, and Ila Neustadt. Enterprise patterns and MDA: building better software with archetype patterns and UML. Addison-Wesley Professional, (2004).