HMI Requirements Creation, as the Collaboration Work of Human and Machine in the Safety-Critical System

Masao Ito(✉)

NIL Software Corp., Tokyo, Japan
nil@nil.co.jp

Abstract. In the safety-critical system, the Human-Machine Interface (HMI) is tightly coupled with system requirements; the functional requirements and the non-functional requirements. As the human has some limitations in his cognitive work, we cannot generate the HMI from the requirements of the complex system in the simplistic way. In this paper, we propose the HMI abstract model from the provisional system requirements, maintaining the simplicity of HMI. We do not intend to create HMI model from the final system requirements but rather traverse the both sides with keeping the safety property. In order to show our idea clearly, we use several examples in the automobile field.

Keywords: HMI · Requirements · Safety · DESH-G · ISO 26262 · Driver model

1 Introduction

The Human-Machine Interface (HMI) has the important role when we design the embedded system of the automobile. It provides environmental information, and we can know the status of my car and neighbouring objects such as other vehicles, the pedestrian and so on. As is well known, we cannot achieve the safety of the car just improving the reliability of the system's elements. When the fault occurs, the HMI plays important role. For example, it provides the information to avoid the accident or transition to the safe state.

The automobile functional safety standard, ISO 26262 [1] requires the functional safety requirements, and it says, "The warning and degradation concept shall be specified as functional safety requirements." (8.4.2.5). Generally, in HMI design the usability is important to focusing on driving, and the usability is a part of the charm of the car. But in this paper, we concentrate on the safety facet of HMI.

In this paper, we first explain the abstract analysis schema, DESH-G [2] for the vehicle. This model covers essential elements comprising driver, environment, system and goal of her or him. Then we will explain how to construct the abstract model of HMI regarding safety.

The motivation of this research comes from two major features in the recent system in the automobile.

J. Stolfa et al. (Eds.): EuroSPI 2017, CCIS 748, pp. 61–71, 2017.
DOI: 10.1007/978-3-319-64218-5_5

The role of software is becoming more important: The software of automobile becomes vast and complicated. It uses the many sensor information and to making the complex decision to control actuators. So, we'd like to consolidate the interface of software at an abstract level.

Moving boundary: The system substitutes for the task of the user. For example, the cruise control system does the acceleration and deceleration instead of the driver. Here we can see the movement of the boundary between the system and the driver. And we know the internal state of the system through the HMI display. This boundary must be simple and easily understandable for the driver. If it is complicated, the driver might make a mistake and meet the accident.

We think that our idea fits better for the new system. The term "new" means the current Advanced Driving Assistant System (ADAS) system and beyond. In this area, the threats to the safety do not always come from the failure of the element. The various types of device and user-interface will be available. For example, we will think the remote parking assist system [15]. The driver is not in the car and uses the simple small device for parking. Such a device is easily lost or broken when he "drives" the car. This is a new type of driving experience, and we might pass over some crucial hazard or the hazardous situation.

2 Desh-G

We already proposed DESH-G [2] schema (Fig. 1) to calculate the controllability for identifying the ASIL (Automotive Safety Integrity Level). In this paper, we use this schema to create the conceptual model of HMI for the automobile. DESH-G has the five elements;

- Driver (D)
- Environment (E)
- Software (S)
- Hardware (H)
- Goal (G) of the driver

The driver is the operator of the car, and we also include the people who control the car outside of it. The environment is not only nature, but it includes elements like the other vehicles, pedestrians, signals, traffic rules and so on. The embedded system of the car consists of the software and hardware. The goal is the driver's aim, it is relating to the scenario of the driving, and we use it to count on the hazardous situations.

Next, we explain each element briefly.

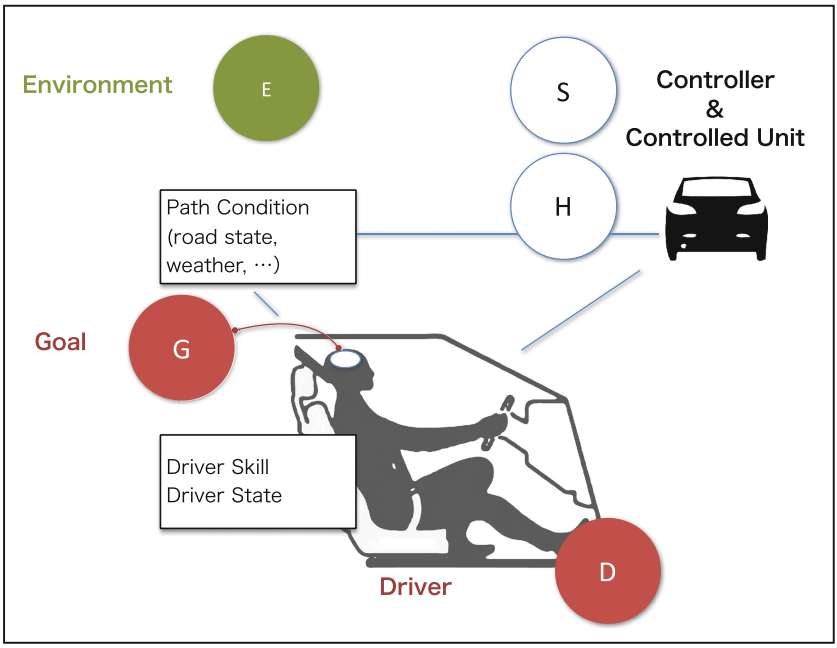

Fig. 1. DESH-G model

2.1 Environment

We use Situation-Scenario Matrix (SSM) [3] 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) [7], 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 type and traffic condition and add the presence or absence of the lock plate.

A car is a moving object and the environment varies as it goes. The SSM has a time axis and one SSM means one scenario. Several scenarios exist to achieve a goal of the driver. We need to analyse this goal of the driver, but I will not describe its detail in this paper.

There are numerous drivers, and modelling of the driver is necessary when we consider the automobile safety. The important characteristic is that the driver is different from the operator of train or aeroplane (i.e. the motorman or pilot). Most of the motorman and pilot are the trained professionals. Vice versa, a large number of drivers are non-professionals and they don't regularly train driving after get the license.

2.2 Driver Model

In this paper, the main purpose of the driver model is to analyse the hazardous situation. So, we simply adopt the task capability interface model [4]. 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. The driver skill is the ability to perform a given task, and this skill doesn't change in the short term. But the driver state easily changes by the various factors. For example, the lack of sleep decreases the level of his state, and it affects the controllability. This doesn't come from the only health problem. If an urgent situation (for example, the driver has to go to school to pick up his/her child, but he 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. And 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 a fine daytime on a highway.

In our approach, to calculate this task demand, we use the SSM, which shows the driver's situation in a particular time. So, we can calculate the change of task demand value in a scenario. We already proposed the several formulas to compute this task demand and the driver's ability [2].

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

3 HMI Abstract Model

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

3.1 Interfaces with the Driver and the Environment

Figure 2 shows the correspondence between the deformed DESH-G model and the interface classes.

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

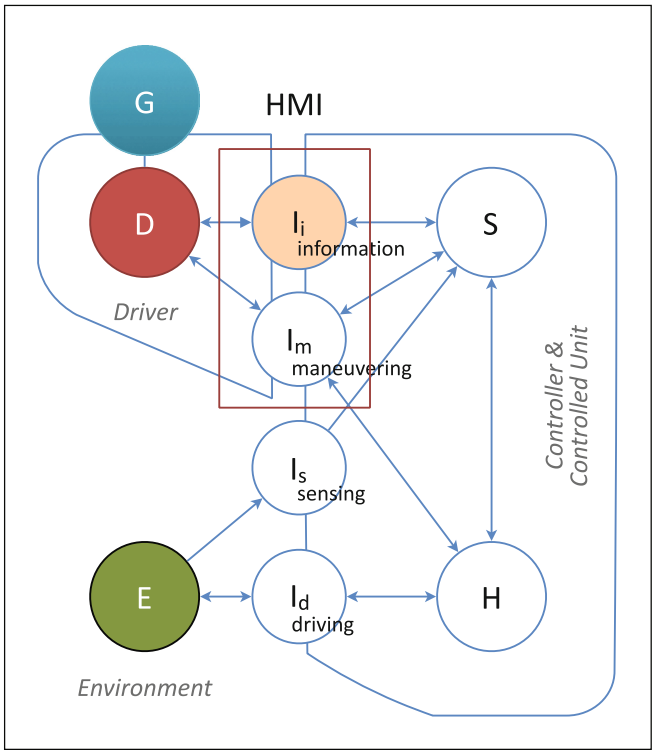

Fig. 2. DESH-G with interface classes [model: S0]

1. Driver Information interface class (I_i)
   By this interface, the system provides information to the driver. For example, traditionally we have the various instruments like speedometer and the several warnings and alerts. If the car has the new functionality, it will give us other type of information. For example, the car that has the adaptive cruise control (ACC) system has information of the distance between forwarding car and the self-car. The system knows information through the I_s interface, and this provided information is useful one for the driver.
   Recently in this class, we have another type of information flow from the driver to the system, not from the system to him. 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.
2. Maneuvering 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.

3. Environment information interface class (I_s)
   This class and next one is the interface between the system and environment. Mainly we can get the data of the environment around the car with the visual or radar sensors. Recently we can get the information via other cars (V2V) or the infrastructure (V2I).
4. Drive interface class (I_d)
   This interface relates the environment directory. For example, the tires transmit the driving and side force to the ground. The car light illuminates the front area.

3.2 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 is the basic model, and we think this is useful. When we have to think the new types of HMI, especially, in the ADAS area, it is not simply covered by the existing HMI techniques and we need another approach.

Example1: CACC
To explain how to use those interface classes. We use an example: CACC system. In the CACC system, the system is not closed in the single car. It communicates with other cars. The information from the other car is useful: in the conventional auto-cruise control (ACC) system, the system has to calculate the distance between the following car and self-car by using the camera or the millimetre-wave radar. It might not be accurate because of bad weather condition. And it has sometimes delay for the calculation of distance. But the CACC system can get the information of the actions of the driver of the following car almost simultaneously. So, the self-car can follow the forward car accurately and timely.

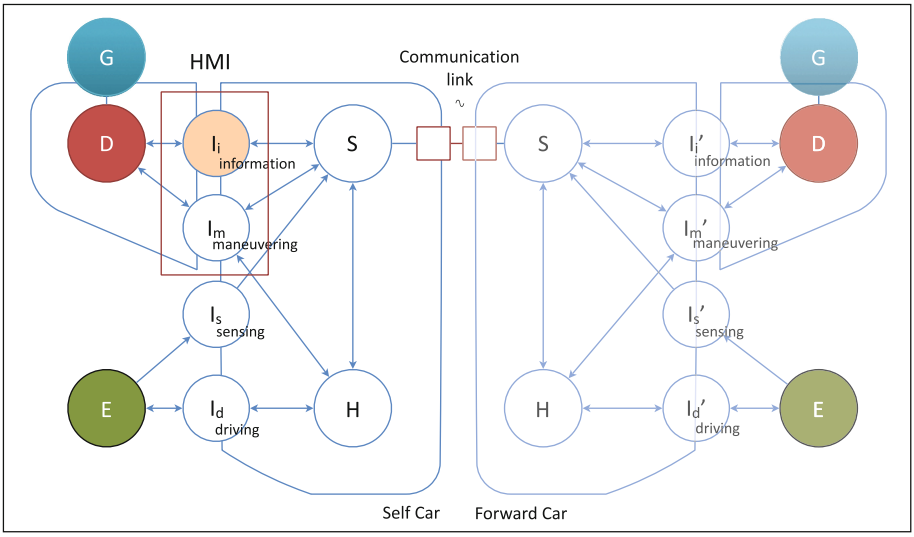

Fig. 3. CACC and DESH-G [model: S1 ∿ S2]

We convert the original model [S0] in order to express the CACC system. Figure 3 shows the CACC model. The new model name "S1 ~ S2" includes the symbol '~', and this denotes the communication between the self-car (S1) and the forward car (S2).

In Fig. 3, the left side shows the self-car, and the right side is the forward car. After establishing the communication link, the self-car can get the information of the forward car (and vice versa). Those are status (I_i') and operation (I_m') of the other driver, and sensory information (I_s') and driving information (I_d') of the forward car.

We show the sample display of this case in Fig. 4. In this figure we also indicate the relation with each interface classes. I_m' is the useful information for the driver to know about the operation of the forward car. And I_s' provides the environment information that the sensor of the self car cannot detect from his position. I_i and I_i' is the information of the driver if our car has the mechanism of driver observation. Of course, the interface information of the forward car, I_d' or I_m', is also the input of the controller of self-car and the behaviour of the self-car might change; for example, the self-car can know the

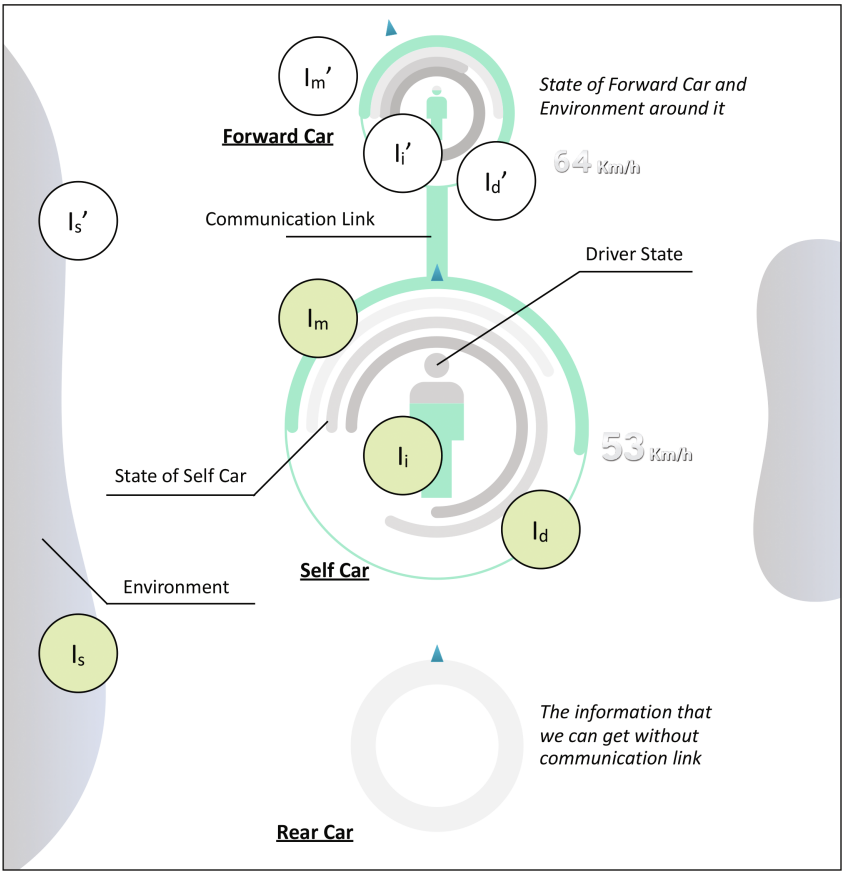

Fig. 4. Sample CACC HMI abstract model and interface classes

forward car will slow down from the information that the driver of the forward car pressed down the brake pedal, so the controller of the self-car can prepare for braking to keep the distance.

Those relations between the display information and the interface classes are useful in the preliminary design of HMI and the item, also from the viewpoint of safety. In chapter four, we take up the safety issue.

Example 2: Remote Parking Assist

Next we think about the remote parking assist system. In this system, "(a)ll drivers need to do is press and hold a button on their ignition key or smartphone. This tells the vehicle to automatically maneuver itself into the parking space" [15]. Figure 5 shows this situation.

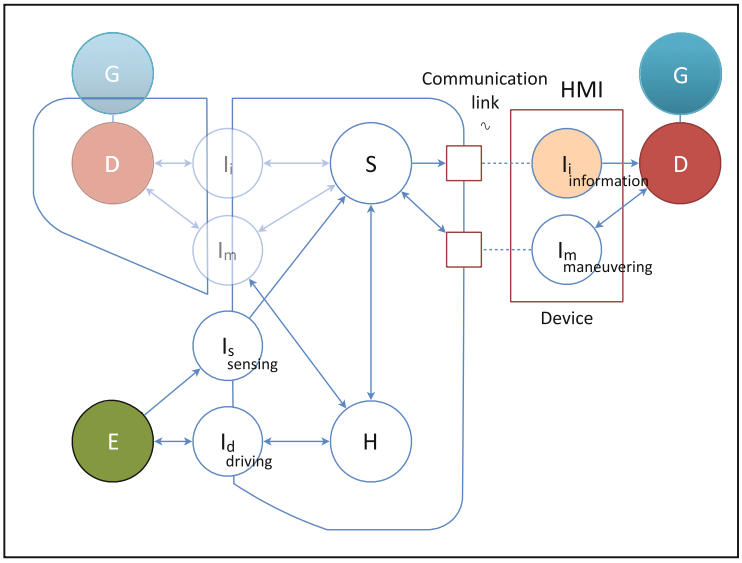

Fig. 5. Remote parking assist [model: S ∿ d]

The driver is outside the car, so the HMI is only on the small device in his hand. This device connects to the system of the car. The information displayed can be small, because the movement of a car is simple and the driver outside the car easily makes sure that the car position and the relation with the obstacles. Also, the maneuvering is restricted, that is, low speed and simple move, because we only can give the simple indication by the small device.

4 Using HMI Abstract Model for Safety Analysis

In this chapter, we consider the relationship between the HMI abstract model and safety analysis. We've already proposed the method, CARDION [3], to analyse the concept phase of system development. In this approach, we first analyse an item, which is the

abstract system in the term of ISO 26262. The item has functional and non-functional requirements. We elaborate this item by using *the item sketch* and *the goal tree*. The item sketch is the rough description on an item from the static or dynamic view. The goal tree is the tree that is obtained by the dividing a top goal of an item iteratively: a top goal is divided into the sub-goals, and the sub-goal also is divided into the sub-sub-goal. In this process, we can find the *obstacle* to achieve a goal, (or it might be the divided goal). The failure of a part of the item can be the candidate of obstacle. To do this, we use the item sketch and guideword. After calculating the effect of obstacle, we design the counter-measure to treat the obstacle. This process corresponds the hazard analysis and risk assessments (HARA) and defining the safety goals.

Figure 6 show the goal model for the CACC system.

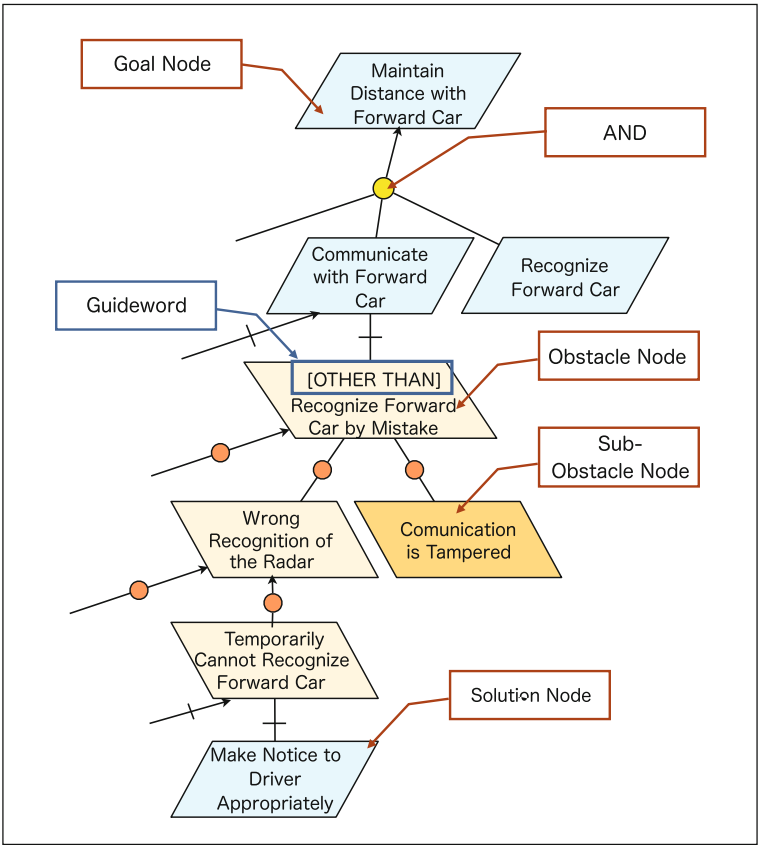

Fig. 6. CARDION goal tree

The HMI abstract model has a major role in this process. Usually, the hazard situation must be informed to the driver appropriately. If the system detects the malfunction of the item, for example, if the CACC system lost their connection link with

the forward car, it doesn't work anymore. The driver has to know that the car cannot keep the speed or distance with the forward car correctly and might be controlled by him. In the Fig. 5, the driver lost the information of the forward car (I_i', I_m', I_s' and I_d') abruptly. To know this is very important because when we calculate the automotive safety integrity level (ASIL) in the HARA process, the controllability of the driver is the key factor for calculation. The controllability is the ability to control the car in occurring the hazardous situation, and we can use the HMI abstract model with our CARDION [3] method to evaluate the controllability. We can know the place lacking information in the hazardous situation by the interface class and item sketch information.

Our approach has another valuable point about safety. In the ISO 26262, the cause of safety-threatening is the failure of a part of the system. But, there is another case. If the design or implementation has an error, we cannot make the car safe. The ADAS and its successors have a complicated structure, and those developments are the new experience for the system designer and the usage of them is challenge for the driver. So, in the early phase, it is important to find the hazard that isn't relating to the system failure, with evaluating the usability. The HMI abstract model is useful for this purpose. We think the remote parking assists system again. It is the new experience for us to drive the car outside a car. And we have neither steering wheel nor the accelerator/brake pedal. What if we drop the small device? Usually, we don't think that we drop the steering wheel. But in this parking assist case. We have to consider this situation (e.g. after dropping the device, we might not lose the measure to stop the car). This is relating to safety, but that doesn't come from system failure.

5 Conclusion and Future Work

This paper is preliminary research for the human-machine interface in the safety-critical system. Especially we are now focusing on the concept phase of the advanced system [11], such as ADAS. First, we introduce the DESH-G model. Traditionally one might use the control-plant model [12] as a fundamental schema, but we think that we have to include the driver and the environment around the car in it. Because there are numerous types of the driver, the operator of the car, compared with the other safety relating system like the power plant, the airplane, train and so on. In the DESH-G model, we distinguished four interface classes. If we use these classes with CARDION approach in order to functional safety analysis in the concept phase, we can clarify the HMI as the abstract model. We show the CACC and remote parking assist cases.

The HMI of the car is essential. If we would use the inappropriate user interface, we encounter the dangerous situation even if there is no system failure. We believe that this type of analysis becomes so important, as the system becomes more complicated and we also have to consider the collaboration with other system, that is, other cars (V2V) and infrastructure (V2I).

Without system failure, we would encounter the hazardous situation. If we misunderstand the alarm or the information displayed on the dashboard, we might do the wrong action. We think it is similar to an issue of the human computer interaction. The 'communicative breakdown' is the failure to exchange the information between the

human and machine [14]. The work of Suchman is typical. She identified two types breakdowns: the false alarm and the garden path. “In the first case, a misconception on the user’s part produces evidence of an error in her actions where none exists; in the second, a misconception on the user’s part produces an error in her action, the presence of which is masked.” [13] If we don’t understand system behaviour correctly, we cannot communicate with the machine appropriately, even when the system work rightly. To avoid these breakdowns, we have to analyse dynamically the relation between the driver and system, but this is out of scope of this paper. But, we already have the SSM (Situation-Scenario Matrix) [3] to analyse the hazardous situation, so we believe that we will report the dynamic behaviour of the relationship between HMI and the driver in future.

References

1. ISO, ISO 26262. Road vehicles - Functional safety -, ISO (2011)
2. Ito, M.: Controllability in ISO 26262 and driver model. In: O’Connor, R.V., Akkaya, M.U., Kemaneci, K., Yilmaz, M., Poth, A., Messnarz, R. (eds.) EuroSPI 2015. CCIS, vol. 543, pp. 313–321. Springer, Cham (2015). doi:10.1007/978-3-319-24647-5_26
3. Ito, M.: Finding threats with hazards in the concept phase of product development. In: Barafort, B., O’Connor, R.V., Poth, A., Messnarz, R. (eds.) EuroSPI 2014. CCIS, vol. 425, pp. 277–284. Springer, Heidelberg (2014). doi:10.1007/978-3-662-43896-1_25
4. Fuller, R., Santos, J.A.: Psychology and the highway engineer. In: Human Factors for Highway Engineers, pp. 1–10 (2002)
5. van Lamsweerde, A.: Requirements Engineering: From System Goals to UML Models to Software Specifications. Wiley, Chichester (2009)
6. Redmill, F., Chudleigh, M., Catmur, J.: System Safety: HAZOP and Software HAZOP. Wiley, Chichester (1999)
7. Naus, G., et al.: Cooperative adaptive cruise control. In: IEEE Automotive Engineering Symposium, Eindhoven, The Netherlands (2009)
8. Weitkamp, C. (ed.): LIDAR: Range-Resolved Optical Remote Sensing of the Atmosphere, vol. 102. Springer, New York (2006)
9. (2015). https://eengenious.com/2016-bmw-7-series-adds-remote-control-parking-gesture-control/
10. Dijkstra, E.W., Buxton, J.N., Randell, B. (eds.) Software Engineering Techniques, Report on a Conference Sponsored by the NATO Science Committee, Rome, Italy, 27–31 October 1969, p. 16 (1970)
11. Ito, M.: How can we deal with the concept phase in the functional safety standard for automobiles? In: Proceedings of Safety-Critical Systems Symposium 2016 (SSS 2016) (2016)
12. Irwin, D.J.: The Industrial Electronics Handbook. CRC Press (1997)
13. Suchman, L.: Human-Machine Reconfigurations: Plans and Situated Actions, pp. 161–162. Cambridge University Press (2007)
14. Philip, J., Hayes, D.: Raj Reddy, Steps toward graceful interaction in spoken and written man–machine communication. Int. J. Man-Mach. Stud. **19**(3), 231–284 (1983)
15. http://www.bosch-presse.de/pressportal/de/en/bosch-technology-makes-parking-a-piece-of-cake-44809.html. Accessed 24 April 2017