Design of Driver-Vehicle Interface to Reduce Mode Confusion for Adaptive Cruise Control Systems

Sang Hun Lee Technische
Graduate School of Automotive Engineering, Kookmin University
77 Jeongneung-ro, Seongbuk-gu, Seoul, 136-702, Korea
shlee@kookmin.ac.kr,
djagnltn@hanmail.net

Hwisoo Eom
Graduate School of Automotive Engineering, Kookmin University
77 Jeongneung-ro, Seongbuk-gu, Seoul, 136-702, Korea
shlee@kookmin.ac.kr,
djagnltn@hanmail.net

Abstract
Adaptive cruise control (ACC) systems have several operational modes. Drivers may be unaware of the mode where they are operating, which can cause traffic crashes. To suppress mode confusion, we developed a new interface design methodology in which the designer checks the consistency between the machine and interface model. If two models are inconsistent, the designer modify the machine model, which is called engineer-oriented approach, or modify the interface model, called user-centered approach, of modify both models. Based on engineer- and user-centered approaches, two different ACC user interfaces were designed. In addition, human-in-the-loop experiments were performed using a driving simulator. The experimental results showed that the user-oriented design approach provides a more compact, acceptable, and effective interface that can reduce the driver's mode confusions significantly.

AutomotiveUI '15 Adjunct, September 01-03, 2015, Nottingham, United Kingdom
ACM 978-1-4503-3858-5/15/09.
http://dx.doi.org/10.1145/2809730.2809757

Author Keywords
Adaptive cruise control; intelligent vehicle; mode confusion; situation awareness; user interface; formal method.

ACM Classification Keywords
H.1.2. User/Machine Systems, H.5.2. User interfaces, H.5.3. Group and Organization Interfaces

Introduction
Recently, intelligent vehicles have been equipped with adaptive cruise control (ACC) systems to enhance the safety of drivers and passengers The ACC system

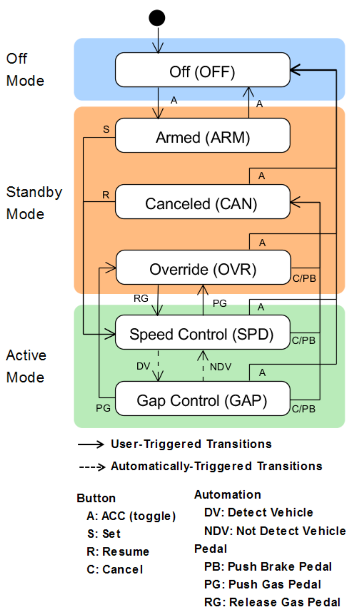

Figure 1: Initial design of ACC user interface with six states and three mode.

operates in multiple modes. If the user interface does not provide sufficient feedback about the state of the system and the driver does not have correct mental model for the system, the driver may suffer from mode confusion or automation surprise, which can cause traffic crashes. Therefore, it is necessary to consider how to reduce the possibility of mode confusion during the design of driver interfaces for ACC systems [1-4].

To meet this requirement, we developed a new interface design methodology for automated systems and verified its effectiveness thorough user interface design and driver-in-the-loop experiments for ACC systems. Consequently, user-oriented approach where the machine model was modified according to the interface model can provide users with a more compact, clear, convenient ACC interface that can reduce mode confusions significantly.

Design and Verification of User Interfaces

To prevent users from experiencing mode confusion and automation surprise at during human–machine interactions, the machine and interface models of the system should satisfy the following two criteria [4]. First, the machine's response to user-triggered events must be deterministic, i.e., starting in the same mode, identical user events should produce identical transitions between system modes. Second, mode changes that are not present in the interface model should not be triggered by users.

To facilitate checking, we propose a simple approach that uses a state and mode transition table for a combined machine and interface model that contains all the states and modes, the events triggered by the user or the system, and their resulting transitions. Two states need to be distinguished from each other if they do not satisfy the following conditions: (a) they belong to the same mode; (b) each user-triggered event that is available and active in one of the states is also available and active in the other state; and (c) starting from either of the two states and triggered by the same event, the state pairs transitioned, respectively, also satisfy conditions (a) and (b). The state pairs that satisfy these conditions are referred to as compatible [2]. Thus, the verification of an interface model designed begins by building a state and mode transition table and proceeds with checking whether these criteria are satisfied.

If any incompatible states are found in a mode, any of the following three different approaches can be employed to resolve the incompatibilities: (1) modify the interface model by partitioning the mode for incompatible states, (2) modify the states and/or their transitions in the machine model, or (3) modify the modes of the interface model and the states of the machine model at the same time.

Development of User Interfaces for ACC Systems

The design process consists of the following three steps:

(Step 1) Design of the machine and interface models. The machine model for the ACC systems implemented in most current vehicles comprises six states: off, armed, canceled, override, speed control, and gap control. The interface model accepted in common has three modes: off, standby, and active. The states are grouped into each mode as illustrated in Figure 1. The off mode includes the off state. The

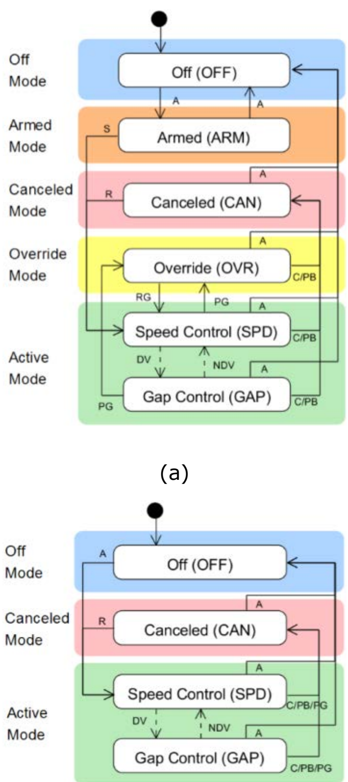

Figure 3: Design of new interface models for the ACC system: (a) Five-mode interface model, (b) Three-mode interface model.

standby mode includes the armed, canceled, and override states. The active mode contains the speed control and gap control states.

(Step 2) Detection of incompatible mode transitions. Check whether any incompatible mode transition exists using the proposed simple approach that uses a state and mode transition table for a combined machine and interface model. As a result, as shown in Figure 2, the three states in the standby mode are incompatible with each other.

Interface Model	Mode	State	Driver's Operation							Compatibility
			A	S	R	C	PG	RG	PB	
Initial Interface	Off	OFF	ARM							Y
	Standby	ARM	OFF	SPD						N
		CAN	OFF	SPD	SPD					N
		OVR	OFF			CAN		SPD		N
	Active	SPD	OFF			CAN	OVR		CAN	Y
		GAP	OFF			CAN	OVR		CAN	Y

Figure 2: Mode and state transitions caused by user-triggered events and their compatibility test.

(Step 3) Removal of incompatible mode transitions. If any incompatible mode transitions exist, modifying the interface model and/or machine model to eliminate incompatible modes using the proposed methods. We designed two types of the interface models as shown Figure 3(a) and (b). Figure 3(a) shows an interface model obtained by separating the standby mode for each incompatible state. Thus, this interface model includes the total of five modes [3]. So this model is called the five-mode interface model in this paper. This type of approach is called the machine-centered approach. The other approach is called the interface-centered approach. Here, the machine model is modified to remove incompatible mode transitions. Figure 3(b) shows the design result. Now, the system has three modes and four states and has no incompatible transitions any more as shown in Figure 4.

Interface Model	Mode	State	Driver's Operation							Compatibility
			A	S	R	C	PG	RG	PB	
Five-mode Interface	Off	OFF	ARM							Y
	Armed	ARM	OFF	SPD						Y
	Canceled	CAN	OFF	SPD	SPD					Y
	Override	OVR	OFF			CAN		SPD		Y
	Active	SPD	OFF			CAN	OVR		CAN	Y
		GAP	OFF			CAN	OVR		CAN	Y
Three-mode Interface	Off	OFF	SPD							Y
	Canceled	CAN	OFF		SPD					Y
	Active	SPD	OFF			CAN	CAN		CAN	Y
		GAP	OFF			CAN	CAN		CAN	Y

Figure 4: Mode and state transition table for two ACC interface models.

Driver-in-the-Loop Experiment Results

The experiments were conducted in a fixed-base driving simulator with TNO PreScan software [5]. The ACC system is implemented based on PreScan, using Mathworks Matlab and Simulink. The input device is a Logitech G27 racing wheel with gas and brake pedals. The buttons on the wheel are configured to produce various operations during ACC mode transitions. A road with six lanes (three lanes in each direction) was modeled using PreScan for use in the experiments conducted in the driving simulator. The road was based on the Marina Bay Street Circuit of the Singapore Grand Prix, which consist of street roads in a harbor-side location. In the scenario, ten events were designed to occur in specific regions. The scenario shows the designed traffic situation, the expected driver operation, and the ACC mode change during each event.

The experimental participants are the total of forty participants, aged between 20 and 65 years, from the university and a company. During the experiments, the experimenter covered the interface on the gauge cluster and interrupted the driving simulation to ask the participant about the mode change and the reason for the answer given. In more detail, the experimental data were collected for each event in the experiments: the participant's operation, the actual mode after the operation, the mode that the participant predicted without looking at the ACC interface, the reason the participant thought he/she was in that mode, and the mode that the participant recognized after looking at the ACC interface. The mode confusion was examined for two independent variables: the type of user interface and whether or not the participant glanced at the display. When the experiments were finished, the participants filled out questionnaires about whether they felt any mode confusion emotionally.

The experimental results show that the three-mode interface was the most effective reducing mode confusion. The total rates of mode confusion with the five-mode interfaces were 8% whereas that with the three-mode interface was only 2.3%. The experimental results show that the difference of the mode confusion rates of the five- and three-mode interfaces was significant. The highest mode confusion rate was observed in the armed mode of the five-mode interface. On the contrary, the three-mode interface shows very low confusion rates in the canceled mode as well as the other modes.

Conclusion

In this study, we proposed a new interface design methodology and applied it to design of the driver interfaces for an ACC system. Here, the consistency between the initial machine and interface models is checked using the proposed criteria, and then, if exists, one or both of the two models is/are modified to make them consistent. The results of the driver-in-the-loop experiments supported that the user-oriented method that modifies only the machine model be more suitable to produce a compact and succinct user interface than the engineer-oriented method that modifies only the interface model. For future work, there is an urgent need to develop optimal user interfaces for multiple automation levels. Therefore, it is necessary to extend this study to higher levels of autonomous vehicles.

Acknowledgments

This work was supported by a grant from the National Research Foundation of Korea (grant NRF-2013R1A2A2A0 1068766) funded by the Ministry of Science, ICT and Future Planning (MSIP) of Korea. A full paper extended from this article was published in Sensors [9]. The corresponding author is Sang Hun Lee.

References

1. Yukio Horiguchi, Ryuichi Fukuju, and Tetsuo Sawaragi. 2006. An estimation method of possible mode confusion in human work with automated control systems. *In Proceedings of the SICE-ICASE International Joint Conference*, 943-948. http://dx.doi.org/10.1109/SICE.2006.315649
2. Michael Heymann, and Asaf Degani. 2013. Automated driving aids: modeling, analysis, and interface design considerations. *In Proceedings of the 5th International Conference on Automotive User Interfaces and Interactive Vehicular Application* (AutomotiveUI, 13), 142-149. http://dx.doi.org/10.1145/2516540.2516549
3. Sang Hun Lee, Dae Ryong Ahn, and Ji Hyun Yang. 2014. Mode confusion in driver interfaces for adaptive cruise control system. *In Proceedings of the 2014 IEEE International Conference on*

Systems, Man, and Cybernetics, 4131-4132.
http://dx.doi.org/10.1109.SMC.2014.6974577

4. Michael Heymann, and Asaf Degani. 2007. Formal analysis and automatic generation of user interfaces: approach, methodology, and an algorithm. *Human Factors: The Journal of the Human Factors and Ergonomics Society*. 49, 2, 311-330.
http://dx.doi.org/10.1518/001872007X312522

5. TASS International, PreScan. Retrieved June 29, 2015 from http://www.tassinternational.com

6. Nadine B. Sarter, David D. Woods and Charles E. Billings. 1997. Automation surprises. *Handbook of human factors and ergonomics* (2nd. ed.), Gavriel Savendy. (Ed). Wiley, NY, USA, 1926-1943.

7. Matthew L. Bolton and Ellen J. Bass. 2013. Using formal verification to evaluate human-automation interaction: a review. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. 43, 3, 488-503.
http://dx.doi.org/10.1109/TSMCA.2012.2210406

8. Sang Hun Lee and Dae Ryong Ahn. 2015. Design and verification of driver interfaces for adaptive cruise control systems. *J. Mech. Sci. Technol*. 29, 6, 2451-2460.

   http://dx.doi.org/10.1007/s12206-015-0536-9

9. Hwisoo Eom and Sang Hun Lee. 2015. Human-automation interaction design for adaptive cruise control systems of ground vehicles. *Sensors*. 15, 6, 13916-13944.
http://dx.doi.org/10.3390/s150613916