Keynote Address at the AAET 2005 (Automation, Assistance and Embedded Real Time Platforms for Transportation),
February 16-17, 2005; Technical University of Braunschweig, Germany.

Design of Human-Machine Interactions for Enhancing Comfort and Safety

Toshiyuki Inagaki, Prof. Dr.
Dept. of Risk Engineering, University of Tsukuba, Tsukuba 305-8573 Japan
inagaki@risk.tsukuba.ac.jp

ABSTRACT *- Function allocation needs to be dynamic and situation-adaptive to support humans appropriately. Machines have thus been given various types of intelligence. Intelligent machines can now sense and analyze situations, decide what must be done, and implement control actions. It is true, however, humans working with such smart machines often suffer negative consequences of automation, such as the out-of-the-loop performance problem, loss of situation awareness, complacency or over-trust, and automation-induced surprises. By contrasting aircraft and automobile, this paper gives some viewpoints that are useful in designing sensible human-machine interactions.*

1. INTRODUCTION

Suppose we are to design a human-machine system. The design decision of assigning functions to human and machine is called *function allocation*. In spite of its importance, function allocation has not become a science yet, but still a kind of art. The traditional ways of function allocation are classified into three categories: The first category is termed *comparison allocation*, or, MABA-MABA (what “men are better at” and what “machines are better at”) approach. The strategies of this type compare relative capabilities of humans versus machines for each function, and they allocate the function to the most capable agent. The second type is called *leftover allocation*. The strategies of this type allocate to machines every function that can be automated, and thus human operators are assigned the leftover functions to which no automation technologies are available. The third type is *economic allocation* that tries to find an allocation ensuring economical efficiency. Even when some technology is available to automate a function, if automating the function is not cost-effective, the function is assigned to the operator. The traditional strategies described above consider “who does what.” Such design decisions yield function allocations that are *static*: viz., once a function is allocated to an agent, the agent is responsible for the function at all times.

Though the static function allocations are easy to implement, human operators may not be very happy with them. The leftover and the economic allocation strategies do not reflect human characteristics, and treat the operators as if they were machine elements. The resulting function allocation can be elusive for the operators, and they may have to adapt to the machines unwillingly. The comparison allocation seems to be nicer for the humans than either the economic or leftover allocations. Even when the operators are allocated only functions in which humans surpass machines, the superiority may not hold at all times and on every occasion. For example, humans may get tired after long hours of operations, or they may find it difficult to perform the functions under time pressure.

The above discussions imply that “who does what” design decisions are not sufficient, but “who does what and when” considerations are needed, which implies that function allocation must be dynamic. A scheme that modifies function allocation dynamically depending on situations is called an *adaptive function allocation*. Suppose that a human and a machine are to perform assigned functions for some period of time. The operating environment may change as time goes by, or performance of the human may degrade gradually as a result of psychological or physiological reasons. If the total performance or safety is to be strictly maintained, it may be wise to reallocate functions between the human and the machine. The adaptive function allocation assumes criteria to determine whether functions have to be reallocated, how, and when. The criteria reflect various factors, such as changes in the operating environment, loads or demands to operators, and performance of operators. The automation that operates under an adaptive function allocation is called *adaptive automation* (Inagaki, 2003a; Moray, Inagaki, & Itoh, 2000; Parasuraman et al., 1992; Rouse, 1988; Scallen & Hancock, 2001; Scerbo, 1996).

Adaptive automation is expected to improve comfort and safety of human-machine systems. It is well-known, however, that humans working with highly autonomous systems often suffer negative consequences of automation, such as the out-of-the-loop performance problem, loss of situation awareness, automation surprises (see, e.g., Wickens, 1994; Endsley and Kiris, 1995; Sarter and Woods, 1995; Sarter, Woods, & Billings, 1997). Adaptive automation may not also be free from those negative consequences. Moreover, some types of adaptive automation may seem to violate the assumption of *human-centered automation* claiming that “the human must be maintained as the final authority over the automation” (Woods, 1989; Billings, 1997).

By contrasting aircraft and automobile, this paper gives some viewpoints that are useful in the design of sensible human-machine interactions.

2. HUMAN-COMPUTER INTERACTIONS

Drivers or pilots perceive the situation, decide what must be done, and implement a control action. In the design of artefacts to assist drivers or pilots, it is useful to distinguish the following four classes of functions: (1) Information acquisition, (2) Information analysis, (3) Decision and action selection, and (4) Action implementation.

Example 1: Traffic alert and collision avoidance system (TCAS) is a family of airborne devices designed to help pilots to avoid a mid-air collision (US Dept. of Transportation & FAA, 2000). Its functionalities are described as follows.

(1) Information acquisition: TCAS sends interrogations at 1030 MHz that transponders on nearby aircraft respond to at 1090 MHz. By decoding the replies, the position and altitude of the nearby aircraft can be known.

(2) Information analysis: Based on the range, altitude, and bearing of nearby aircraft, TCAS performs *range and altitude tests* to determine whether the aircraft is a *threat* or not.

(3) Decision and action selection: When the nearby aircraft is declared a threat, TCAS selects an avoidance maneuver (to climb or descend) that will provide adequate vertical miss distance from the threat. If the threat aircraft is equipped with TCAS, the avoidance maneuver will be coordinated with the threat aircraft.

(4) Action implementation: TCAS issues the *resolution advisory* (RA) to let the pilot know the appropriate avoidance maneuver. However, TCAS does not perform any avoidance maneuver itself. It is the human pilot who implements the avoidance maneuver.

Example 2: The enhanced ground proximity warning system (EGPWS) is designed to help pilots to avoid a ground collision (Bresley & Egilsrud, 1997). Its functionalities are described as follows.

(1) Information acquisition: EGPWS collects air data, radio altitude, barometric altitude, and airplane position through some other systems, such as Flight Management System, GPS, the airplane air data system.

(2) Information analysis: Receiving the above data, EGPWS determines potential terrain conflict by use of its self-contained worldwide airport and terrain databases. EGPWS displays the terrain in dotted patterns with colors indicating the height of the terrain relative to the current airplane altitude.

(3) Decision and action selection: EGPWS continuously computes terrain clearance

envelopes ahead of the airplane. If these envelopes conflict with data in the terrain database, EGPWS sets off alerts.

(4) Action implementation: EGPWS issues a caution-level alert approximately 40 to 60 seconds before a potential terrain conflict, and sets off a warning-level alert approximately 20 to 30 seconds before a conflict. However, EGPWS does not perform any conflict avoidance maneuver itself.

In the above Examples, information acquisition and information analysis are highly automated. However, the decision and action selected by TCAS or EGPWS are “advices” to human pilots, and the pilots may disregard TCAS resolution advisories or EGPWS alerts. Also, either TCAS or EGPWS has no mechanical subordinate to initiate a collision avoidance maneuver: These systems are not given authority for automatic action implementation.

However, there are cases in which automatic action implementations may be essential for ensuring systems safety. One such example can be seen in the automatic ground collision avoidance system (Auto-GCAS) for combat aircraft (Scott, 1999). When a collision against the terrain is anticipated, the computer gives a “pull-up” warning. If the pilot takes a collision avoidance maneuver aggressively, then the computer does not step in any further. If the pilot does not respond to the warning, the computer takes control back from the pilot and executes an automatic collision avoidance maneuver. In order to describe or consider such automatic action implementations in an emergency, we need the concept of *trading of authority*.

3. TRADING OF AUTHORITY

Trading of authority refers to the human-computer collaboration in which either one of the human or the computer is responsible for a function, and an active agent changes alternately from time to time. Trading of authority is an essential notion in adaptive automation, because adaptive automation needs to modify function allocation between humans and machines dynamically in response to changes in situations, human workload, or performance.

A scheme to implement trading of control is called an *automation invocation strategy*. There are some types of automation invocation strategies (Inagaki, 2003a). Among them, the following two classes are important for transportation systems.

3.1 Critical-event strategies

Automation invocation strategies of this class change function allocations when specific events (called *critical events*) occur in the human-machine system. It is assumed that human workload may become unacceptably high when the critical events occur. If the critical events did not occur during the system operation, allocation of functions would not be altered.

Example 3: Suppose that a man is driving fast on a dark night, and that the computer in the car gives the driver an alert, saying "Slow down! An obstacle is ahead." Even if the driver wants to figure out why the alert was issued, that may not be possible. If the driver may slow down immediately to avoid a possible hazard, then there is no need for the computer to intervene. However, if the driver did not respond at all to the warning, the computer may have to decelerate the car automatically to avoid a collision by controlling the brakes, in which the authority of control the brakes was traded from the human to the computer (Figure 1).

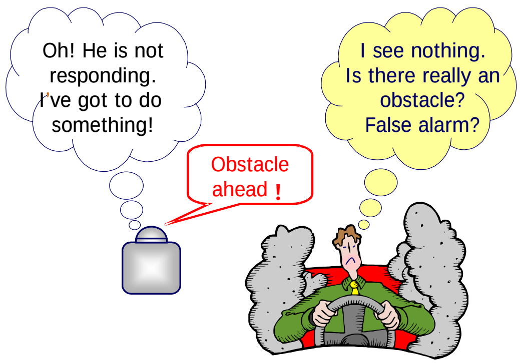

Figure 1. Driver who does not respond to an alert

3.2 Measurement-based strategies

Automation invocation strategies of this class adjust function allocation dynamically by evaluating moment-to-moment workload or total system performance. It is necessary to develop *custom tailored* algorithms if the system is to be compatible with individual operators. Individual differences in human operator capabilities will also influence the response to multiple task demands.

Example 4: Suppose a man is driving his car by letting the *adaptive cruise control* (ACC) system and the *lane-keeping support* (LKS) system at work. The ACC system is intended to reduce the driver's workload by freeing the driver from frequent acceleration and deceleration. The LKS reduces driving workload by assisting driver's steering control to keep the car center of the lane. Suppose the computer determines, by monitoring moment-to-moment steering torque, that the driver has not been actively involved with steering task for a while. Driver's inactive steering can suggest that he may be complacent, overly reliant on the automation, or may be simply drowsy, in which his situation awareness may be poor. The computer decides to return (or, trade the authority of) the steering task to the driver, by expecting that increasing the driver's involvement with the steering task may be useful in improving his situation awareness or vigilance.

4. COST OF AUTOMATION

Human operators sometimes suffer from *mode confusion* during the interaction with intelligent machines. The smart machines can sense, analyze situations, decide what must be done, and implement control actions in highly autonomous manners. It is not easy for operators to keep perfect awareness on mode and intention of intelligent machines, partly due to difficulty in constructing mental models for various context-specific functions of the automation. Operator's psychological characteristics, such as lack of vigilance, complacency, inappropriate *trust* in automation, may also contribute to loss of *mode awareness* (see, e.g., Sarter & Woods, 1995; Parasuraman & Riley, 1997; Wickens & Hollands, 2000; Inagaki & Stahre, 2004).

4.1 Automation surprises

An automation surprise can happen when: (1) designer's assumptions about operator use of automation differ the actual use of automation, (2) operators fail to understand the intention of automation, or (3) situational recognitions differ between operators and the automation. The surprised operators often ask questions, such as, "What is the automation doing?" "Why is it doing that?" "What is it going to do next?" Various *automation surprises* have been reported in aviation (e.g., Sarter et al., 1997). It must be noted that, even for pilots who are well educated and trained, automation surprises may happen.

Advanced automated systems have been introduced actively to passenger cars. It is not appropriate to assume that every ordinary car driver is fully trained and that he/she has deep understanding of onboard machine intelligence. Automation surprise thus may be observed more frequently or widely in automobiles than in aircraft.

Example 5: In some cases, two types of ACC systems are distinguished, viz., the *high-speed range ACC* and the *low-speed range ACC*. There are similarities and dissimilarities between the two types of ACC systems. For instance, when there is a forward vehicle to follow, the both ACC systems control the subject vehicle's speed so that the time gap to the target vehicle may be maintained. Upon loss of the target vehicle, the high-speed range ACC continues to stay in its active state. However, for cases in which the low-speed ACC loses sight of the target vehicle, two design decisions are possible for the ACC. One is to let the ACC stay in its active state, and the other is to put the ACC into its standby state (Figure 2). Which design is better? A clear-cut answer is hard to get. Loss of mode awareness or automation surprises can occur for each design decision; see, e.g., Inagaki & Kunioka (2001); Furukawa, Inagaki, Shiraishi, & Watanabe (2003).

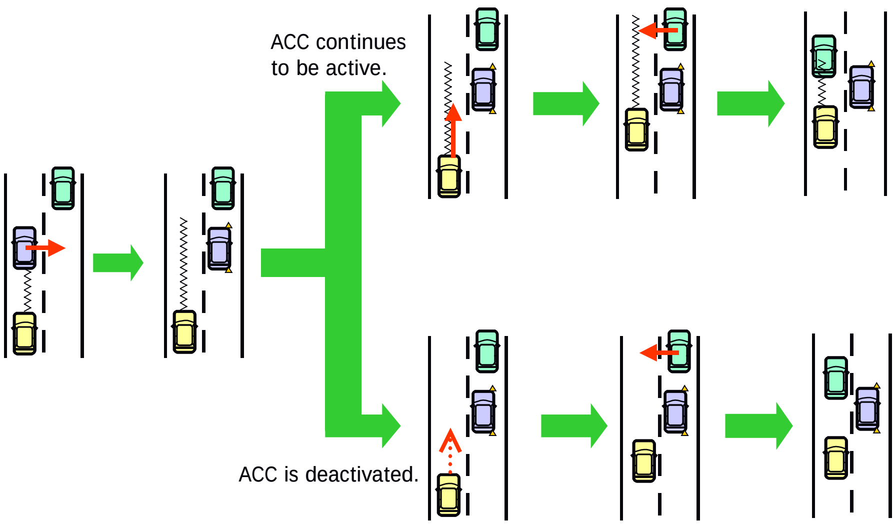

Figure 2. Design alternatives for the low-speed range ACC for cases of loss of the target

4.2 Over-trust in automation

Lee & Moray (1992) distinguished between four dimensions of trust: (a) *foundation*, which represents the "fundamental assumption of natural and social order that makes the other levels of trust possible," (b) *performance*, which rests on the "expectation of consistent, stable, and desirable performance or behavior," (c) *process*, which depends on "an understanding of the underlying qualities or characteristics that govern behavior," and (d) *purpose*, which rests on the "underlying motives or intents." The humans sometimes trust in automation overly.

Example 6: Suppose a low-speed range ACC can decelerate its subject vehicle at some certain deceleration rate, not greater than 2.5m/s^2. When the ACC system detected a target vehicle's rapid deceleration, say, at the rate of 6.0m/s^2, it has to tell the driver that the deceleration exceeds its designed ability, and has to request him/her to apply the brake him/herself hard enough to avoid a collision. Such an ACC system's message may be called a *function-limit alert*. An experiment was conducted to investigate drivers' trust in and reliance on the function-limit alert. Twenty students participated in the experiment, and each subject receives 17 trials. During Trials 1 to 3, no function-limit alerts functionality was available: Subjects had to judge themselves whether he/she has to take over control from the ACC and when, by monitoring the behavior of the target vehicle. During Trials 4 to 10, a correct function-limit alert was issued immediately when the target vehicle decelerated rapidly. In Trial 11, a function-limit alert failed to be issued even when the target vehicle made a rapid deceleration. Significant differences were found among the mean response times of subjects for Trial 3, Trial 10, and Trial 11. While receiving correct function-limit alerts consecutively over the long term, subjects became reliant on the alerts, and tended to judge based on the alerts whether an emergency brake was necessary or not. Thus, when a correct alert failed to be issued in Trial 11, subjects became significantly late in responding to the rapid deceleration of the target vehicle (Inagaki, 2003b).

5. VIEWPOINTS FOR SENSIBLE HUMAN-MACHINE COLLABORATIONS

Human-machine collaborations may be heavily dependent on the transportation modes. Some viewpoints may be necessary for identifying functionalities of operator assistance systems. Let us note here two of those viewpoints: (1) quality of human operators and (2) time-criticality.

Quality of human operators varies depending on modes of transportation. For non-professional operators, such as private car drivers, it is not wise to assume that they have high level of knowledge and skills, or thorough and continual training, which implies that required driver assistance functionalities may be quite different from those for professional operators, such as airline pilots or train drivers.

Time-criticality also differs appreciably depending on transportation modes. Suppose a warning has been set off. If it was a resolution advisory (RA) of TCAS, the estimated time to closest point of approach must be 15 to 35 seconds, and pilots are supposed to respond to the RA within 5 seconds. If it was a warning-level alert of EGPWS, it must have been issued 20 to 30 seconds before a potential terrain conflict. If it was a collision warning on the car, it may have been given just a few seconds prior to a possible collision.

Noting the above two points, let us discuss how we should design functionalities for assisting human operators appropriately and context-dependent manner. Discussions may be made on two aspects: Enhancement of situation awareness, and design of authority.

5.1 Enhancement of situation awareness

Human interface design is a central issue for enhancing situation awareness, avoiding automation surprises, establishing appropriate trust in automation. The implemented human interface must enable the human to: (1) Recognize intention of the automation, (2) Understand why the automation thinks so, (3) Share the situation awareness with the automation, and (4) Show limits of functional abilities of the automation.

The enhancement of situation awareness matches well with the *human-centered automation* concept, in which *human locus of control* is claimed. However, as has been noted earlier, non-professional operators may not be able to cope with the given situation. Even professional operators, they may not respond to the situation appropriately: Recall the mid air crash on July 1, 2002, in which two TCAS-equipped aircraft collided over south Germany (Ladkin, 2002; Learmount, 2002). When a conflict developed between two TCAS-equipped aircraft, the TCAS software determined which aircraft should climb and which should descend. One of the aircraft descended according to the TCAS resolution advisory. The other aircraft also descended, although its TCAS told the pilot to climb, which yielded the mid air collision. As described with Example 1 in section 2, TCAS is not given authority to make the pilots follow its resolution advisory.

5.2 Design of authority

Human-computer interactions can be described in terms of the *level of automation* (LOA). Table 1 gives an expanded version in which a new LOA comes between levels 6 and 7 in the original list by Sheridan (1992). The added level, called the level 6.5, has been firstly introduced in (Inagaki, Itoh, & Moray, 1997) to avoid automation surprises that may be induced by automatic actions, when the actions are indispensable to assure systems safety in emergencies.

Table 1. Scales of levels of automation (expanded version)

1.	The computer offers no assistance; human must do it all.
2.	The computer offers a complete set of action alternatives, and
3.	narrows the selection down to a few, or
4.	suggests one, and
5.	executes that suggestion if the human approves, or
6.	allows the human a restricted time to veto before automatic execution, or
6.5	executes automatically upon telling the human what it is going to do, or
7.	executes automatically, then necessarily informs humans,
8.	informs him after execution only if he asks,
9.	informs him after execution if it, the computer, decides to.
10.	The computer decides everything and acts autonomously, ignoring the human.

The following example illustrates how important it is to choose an appropriate LOA for assuring comfort and safety of semi-autonomous human-machine systems.

Example 7 (continued from Example 4): Suppose a man is driving his car by letting the LKS at work. Suppose the computer determines, by monitoring moment-to-moment steering torque, that the driver has not been actively involved with steering task for a while. The computer decides it appropriate to return the steering task to the driver. How the computer may return the steering task to the driver, and what should the computer say to the driver in the situation? (Figure 3)

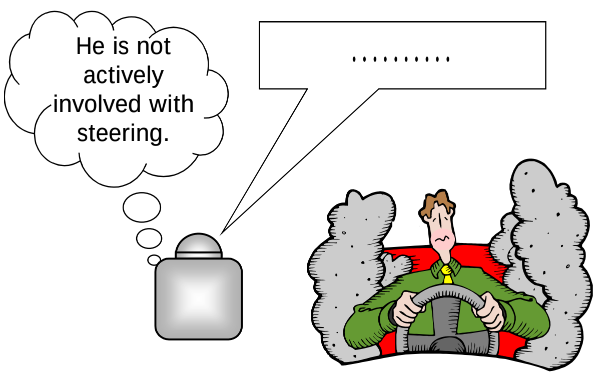

Figure 3. What should the computer say to the poorly involving driver?

There are several alternatives to the computer's message (or action) in the above situation. The simplest alternative would be that the computer tells the driver, "You seem to be bored." The LOA of this strategy is positioned at level 4. However, the driver may not respond at all, if he disagrees with the diagnosis, or if he failed to catch the message due to drowsiness.

The second alternative would be that the computer gives an offer more explicitly, by saying, "Shall I let you drive yourself?" The LOA of this strategy is set at level 5. If the driver did not reply, the computer cannot do anything further, and the lane-keeping task still has to be performed by the automation.

The third alternative may be that the computer gives a stronger message, such as, "I will hand over control to you in a few seconds." The LOA of this strategy is positioned at level 6. In this case, the driver is given the right to invoke a veto. If the driver was too slow to respond to the message within allowed time, the computer puts the lane-keeping support system into its standby state. Then the driver has to take over control even if he/she did not to do so.

The fourth alternative may be that the computer gives the following message after it deactivated the lane-keeping support system: "I have just handed over control to you." The LOA of this strategy is set at level 7. In this case, the driver may be upset if he was not ready to take over control from the automation.

The most extreme case may be that the computer hands over control to the driver *silently*. The LOA of this strategy is set at 8 or higher. In other words, the computer tells nothing to the driver, even though it has already put the lane-keeping support system into its standby state. Suppose the car approaches to a lane boundary some time later. The driver may expect that the lane-keeping support system shall steer the wheel appropriately, because he believes that the automation is still in its active mode. The driver shall be surprised to see that the lane boundary is approaching contrary to expectations.

As the above example illustrates, if the LOA was chosen inappropriately, some undesirable event may happen. In designing human-machine systems, it is important to predict how the design may affect humans and change their behaviors (Hollnagel, 2003).

There are three approaches that are useful for selecting an appropriate LOA. Each of the approaches is illustrated with an example.

Example 8: Selection of an appropriate LOA via theoretical analyses.

Suppose an engine fails while an aircraft is making its takeoff roll. The pilot must decide whether to continue the climb-out (Go) or to abort the takeoff (No Go). The standard decision rule upon an engine failure is stated as follows: (a) Reject the takeoff, if the aircraft speed is below V1, and (b) continue the takeoff, if V1 has already been achieved. The critical speed V1 is called the “takeoff decision speed” at which the pilot must apply the first retarding means in case of No Go. Inagaki (2000) has proven mathematically, under the following assumptions, that decision authority must be traded between human and automation in a situation-adaptive manner to ensure takeoff safety:

(1) An alert is given to the human pilot when a sensor detects an “engine failure.” However the sensor can give a false alert.

(2) The pilot’s understanding on given situation may not be correct. Let C denote that an alert is correct, and F that an alert is false. Let “c” denote the pilot’s judgment that an alert is correct, and “f” that an alert is false. In addition to conventional hit (“c”|C), miss (“f”|C), false alarm (“c”|F), and correct rejection (“f”|F), we introduce (“h”|C) and (“h”|F), where “h” denotes “hesitation” in which the pilot hesitates to say either that the alert is correct or that the alert is false.

(3) Two policies are distinguished for cases of “h”: (i) Trustful Policy (TP), in which the given alert is trusted and the engine is assumed failed, and (ii) Distrustful Policy (DP), in which the given alert is distrusted and the engine is assumed working.

(4) An incorrect or late decision can cause cost, Z, which varies depending on the situation. Three types of conditional expected loss are distinguished: (i) An inappropriate liftoff is made based on an incorrect Go decision, where an emergency landing is required after reducing the weight of the aircraft to its maximum landing weight by dumping fuel, (ii) An unnecessary abort of the takeoff is made due to an incorrect NoGo decision, (iii) An overrun accident is caused by an inappropriate RTO action in excess of V1.

The conditional expected loss, E[Z | engine failure alert], was evaluated for each case in which a Go/NoGo decision and its associated action is made by an Automated System (AS), a human with TP, and a human with DP, respectively. The four phases are distinguished based on the time point at which an engine failure alert is issued.

Phase 1: An engine failure alert is set off at a speed far below V1. Then $L_{DP} \le L_{TP} \le L_{AS}$, which means that the human pilot must be in authority even when there is possibility of delay or an error in his/her decision.

Phase 2: An engine failure alert is issued before but near V1. An RTO can be initiated before V1 if the human responds without any hesitation. We have $L_{DP} \le L_{TP}$. There is no fixed order relation between L_{AS} and L_{TP}, or between L_{AS} and L_{DP}.

Phase 3: An engine failure alert is issued almost at V1 where no human pilot can initiate

RTO by V1 but the automated system can. We have $L_{DP} \leq L_{TP}$, but no fixed order relation exists between L_{AS} and L_{TP}, or between L_{AS} and L_{DP}.

Phase 4: An engine failure alert is given almost at V1 where neither a human pilot nor the automated system can initiate RTO by V1. Then we have $L_{AS} \leq L_{DP} \leq L_{TP}$, which implies that the automation should have authority for decision and control (Inagaki, 1999, 2000).

Example 9: Selection of an appropriate LOA via cognitive experiments.

Another important result in Inagaki (1999, 2000) was that, for a human pilot to be in authority at all times and in every occasion, design of human interface needs to be changed so that more direct information, such as "Go" or "Abort" message may be given explicitly to the human pilot. With the human interface, we have $L_{AS} = L_{DP} = L_{TP}$ in Phase 4 in Example 8.

A flight simulator of a two-engine aircraft has been implemented, and a cognitive experiment with a factorial design, mapping onto (Control mode) x (Phase) x (Human interface design) was conducted. For the control mode, the manual (M) control mode and the situation-adaptive autonomy (SAA) mode were distinguished. In the M-mode, humans have full authority for decision and control. In the SAA-mode, on the other hand, the computer can choose appropriate LOA for decision and control, and may take over control for continuing the takeoff when it judges that it is not possible for humans to initiate an RTO before V1 is achieved. Experimental results showed that, even though the human interface that can give "Go" or "Abort" message was effective in making a decision correctly, some overrun accidents did occur under M-mode. Under SAA-mode, on the other hand, no overrun accident occurred (Inagaki, Takae, & Moray, 1999).

Example 10: Selection of an appropriate LOA via computer simulations.

Suppose a man is driving with the ACC and the LKS working on the host vehicle. While observing the automation behaves correctly and appropriately, it is natural for the driver to trust in automation. Sometimes he may place excessive trust in automation. In such cases, the driver may fail to allocate his own attention to the driving environment, and may pay attention inappropriately to some non-driving tasks (such as, using a mobile phone, manipulation of on-board audio systems). Suppose the ACC recognizes that the deceleration rate of the target vehicle is much greater than the maximum deceleration rate to which the ACC can cope with the ordinary automatic brake. Which is appropriate among the following design alternatives?

Scheme 1: Upon recognition of a rapid deceleration of the target vehicle, the ACC gives an *emergency-braking alert* that tells the driver to hit the brake pedal himself or herself hard enough to avoid a collision. The LOA of this scheme is set at level 4.

Scheme 2: Upon recognition of a rapid deceleration of the target vehicle, the ACC gives an emergency-braking alert, and if the driver does not respond within a pre-specified time, it applies an automatic emergency brake. The LOA of this scheme is positioned at 6.

Scheme 3: Upon recognition of a rapid deceleration of the target vehicle, the ACC applies its automatic emergency brake simultaneously when it issues an emergency-braking alert. The LOA of this scheme is positioned at level 6.5.

Based on discrete-event models for dynamic transition of driver's psychological states and driving environments, Monte Carlo runs were performed for analyzing complacency effect and for comparing the efficacy of schemes 1 through 3. It was observed that, when the driving was peaceful and the ACC continued to be successful in its longitudinal control, the driver was likely to rely on the ACC, and his or her vigilance degraded. When the target vehicle made a rapid deceleration in such cases, the driver needed time to recognize what is happening and thus might not be able to cope with the circumstance in a timely manner, even if an emergency-braking alert was given. The number of accidents under LOA-4 was siginificantly larger than either of those under LOA-6 and LOA-6.5. High LOA was effective to assure car safety under time-criticality, especially when the driver may be inattentive. For more details, see, (Inagaki & Furukawa, 2004).

6. CONCLUDING REMARKS

Various efforts have been exerted to improve safety of transportation systems, such as aircraft, automobile, railroad, and marine vessels. Accidents rates, however, are not low enough. One of the reasons may be that advanced functionalities and speeding up of those transportation systems impose human operators excessive claims on their abilities for situational recognition, decision-making, and action implementation.

Adaptive automation seems to be one of promising frameworks to assure safety as well as comfort of human-machine systems. However, there is no single adaptive automation that is effective to all the application domains. Appropriate adaptive automation must be sought for each application. This paper has outlined viewpoints that may be useful in designing and implementing sensible human-machine interactions and collaborations.

ACKNOWLEDGMENT

This work has been partially supported by JSPS Grant-in-Aid for Scientific Research (No. 14380186) and the Science & Technology Research Funds of the MEXT of Japanese Government for the Special Research Project on "Situation and Intention Recognition for Risk Finding and Avoidance: Human-Centered Technology for Transportation Safety." The author extends his appreciation to All Nippon Airways, JAL-Japan, and Nissan Motors.

REFERENCES

Billings, C. E. (1997). *Aviation automation – the search for a human-centered approach*. LEA.

Bresley, B., & Egilsrud, J. (1997). Enhanced Ground Proximity Warning System. *Boeing Airliner*, July-September 1997, 1-13.

Endsley, M.R. & Kiris, E.O. (1995). The out-of-the-loop performance problem and the level of control in automation. *Human Factors*, 37(2), 3181-3194.

Furukawa, H., Inagaki, T., Shiraishi, Y., & Watanabe, T. (2003). Mode Awareness of a Dual-Mode Adaptive Cruise Control System. *Proc. IEEE SMC Conference*, 832-837.

Hollnagel, E. (2003). Prolegomenon to cognitive task design. In E. Hollnagel (Ed.) *Handbook of Cognitive Task Design*, 3-15, LEA.

Inagaki, T. (1999). Situation-adaptive autonomy: Trading control of authority in human-machine systems. *Automation technology and human performance: Current research and trends* (pp.154-159). LEA.

Inagaki, T. (2000). Situation-adaptive autonomy for time-critical takeoff decisions. *International Journal of Modelling and Simulation*, 20(2), 175-180.

Inagaki, T. (2003a). Adaptive automation: Sharing and Trading of Control. In E. Hollnagel (Ed.) *Handbook of Cognitive Task Design* (pp. 147-169). LEA.

Inagaki, T. (2003b). New challenges on vehicle automation: Human trust in and reliance on adaptive cruise control systems. *Proc. IEA 2003 (CD-ROM)*, 4 pages.

Inagaki, T., & Kunioka, T. (2002). Possible automation surprises in the low-speed range adaptive cruise control system. *IASTED Int'l Conference on Applied Modelling and Simulation*, 335-340.

Inagaki, T., & Stahre, J. (2004). Human supervision and control in engineering and music: Similarities, dissimilarities, and their implications. *Proceedings of the IEEE*, 92(4), 589-600.

Inagaki, T., & Furukawa, H. (2004). Computer simulation for the design of authority in the adaptive cruise control systems under possibility of driver's over-trust in automation. *Proc. IEEE SMC Conferece*, 3932-3937.

Inagaki, T., Moray, N., & Itoh, M. (1997). Trust and time-criticality: Their effects on the situation-adaptive autonomy. *Proc. International Symposium on Artificial Intelligence, Robotics, and Intellectual Human Activity Support for Nuclear Applications*, 93-103.

Inagaki, T., Takae, Y., & Moray, N. (1999). Automation and human interface for takeoff safety. *Proc. Tenth Int. Symposium on Aviation Psychology*, 402-407.

Ladkin, P.B. (2002). *ACAS and the south German midair*. Technical Note RVS-Occ-02-02. (http://www.rvs.uni-bielefeld.de/publications/Reports/)

Learmount, D. (2002, July 9-15). Questions hang over collision. *Flight International*, 8.

Lee, J.D. & Moray, N. (1992). Trust, control strategies and allocation of function in human machine systems. *Ergonomics*, 35(10), 1243-1270.

Moray, N., Inagaki, T., & Itoh, M. (2000). Adaptive automation, trust, and self-confidence in fault management of time-critical tasks. *Journal of Experimental Psychology: Applied*, 6(1), 44-58.

Parasuraman, R., & Riley, V. (1997). Humans and automation: Use, misuse, disuse, abuse. *Human Factors*, 39(2), 230-253.

Parasuraman, R., Bhari, T., Deaton, J.E., Morrison, J.G., & Barnes, M. (1992). *Theory and design of adaptive automation in aviation systems* (Progress Rep. No. NAWCADWAR-92033-60). Naval Air Development Center Aircraft Division.

Rouse, W.B. (1988). Adaptive aiding for human/computer control. *Human Factors*, 30(4), 431-443.

Sarter, N.B. & Woods, D.D. (1995). How in the world did we ever get into that mode? Mode error and awareness in supervisory control. *Human Factors*, 37(1), 5-19.

Sarter, N.B., Woods, D.D., & Billings, C.E. (1997). Automation surprises. In G. Salvendy (Ed.). *Handbook of Human Factors and Ergonomics* (2nd ed., pp. 1926-1943). Wiley.

Scallen, S.F. & Hancock, P.A.(2001). Implementing adaptive function allocation. *International Journal of Aviation Psychology*, 11(2), 197-221.

Scerbo, M. W. (1996). Theoretical perspectives on adaptive automation. In R. Parasuraman & M. Mouloua (Eds.). *Automation and human performance* (pp.37-63). LEA.

Scott, W.B. (1999, February 1). Automatic GCAS: "You can't fly any lower." *Aviation Week and Space Technology*, 150(5), 76-79.

Sheridan, T. B. (1992). *Telerobotics, automation, and human supervisory control*. MIT Press.

US Dept of Transportation & FAA (2000). *Introduction to TCAS II Version 7*

Wickens, C.D. (1994). Designing for situation awareness and trust in automation. *Proceedings of IFAC Integrated Systems Engineering*, 77-82.

Wickens, C.D. & Hollands, J.G. (2000). *Engineering psychology and human performance* (3rd ed.). Prentice-Hall.

Woods, D. (1989). The effects of automation on human's role: Experience from non-aviation industries. In S. Norman & H. Orlady (Eds.). *Flight deck automation: Promises and realities* (NASA CR-10036, pp.61-85). NASA-Ames Research Center.