Chapter 12
Human–Machine Coagency for Collaborative Control

Toshiyuki Inagaki

Abstract This chapter discusses some of the issues that are at the center of designing human–machine coagency where humans and smart machines collaborate and cooperate sensibly in a situation-adaptive manner. The first is the issue of authority and responsibility. It is argued that the machine may be given authority to improve safety and to alleviate possible damage to the human–machine system, even in a framework of human-centered automation. The second is the issue of the human operator's overtrust in and overreliance on automation, where it is argued that possibilities and types of overtrust and overreliance may vary depending on the characteristics of the automated system. The importance of the design of a human–machine interface and human–machine interactions is included in the discussion.

Keywords Human supervisory control • Function allocation • Human-centered automation • Authority and responsibility • Overtrust and overreliance • Levels of automation

12.1 Introduction

Many complex industrial processes are semi-autonomous, where computers control the processes based on directives given by human operators. The configuration of such human–machine systems is called *human supervisory control* [1]. Why are these processes semi-autonomous, rather than being fully automated? The most obvious reason is that we cannot foresee in the design phase all possible events that may occur during the expected lifetime of the processes. Thus, although designers have tried to replace human operators with machines for higher efficiency or

T. Inagaki (✉)
Graduate School of Systems and Information Engineering, University of Tsukuba,
Tsukuba, Japan
e-mail: inagaki.toshiyuki.gb@u.tsukuba.ac.jp

Y. Sankai et al. (eds.), *Cybernics: Fusion of human, machine and information systems*,
DOI 10.1007/978-4-431-54159-2_12, © Springer Japan 2014

reliability, their attempts have not been entirely successful [2]. Actually, human operators have to be on-site to perform the task of "completing the system design," that is, adapting the system for situations that the designers did not anticipate [3].

It then becomes an important design decision to determine what humans should do and what machines should do. One design strategy is to allocate to machines every function that can be automated, and to allocate to operators the leftover functions for which no automation technologies are available. Another strategy is to find an allocation that ensures economic efficiency. Such design strategies are typical examples of *technology-centered automation* [4]. This may remind readers of Charlie Chaplin's *Modern Times*, which portrays a comic yet unpleasant world in which seemingly intelligent machines demand that humans obey or adapt to the machines.

Human-centered automation is what is needed to realize an environment in which humans and machines can work cooperatively in a more sound and comfortable manner [5]. However, in spite of the popularity of the term, it is still not clear what human-centered automation really means. In fact, there are several different meanings of human-centered automation, as pointed out by Sheridan [6]. He discusses the possible and even probable contradictions that may be found in the "definitions" of human-centered automation [6].

Human–machine systems are not yet free from problems, such as (a) loss of *situation awareness*, in which operators fail to grasp the process state exactly [7]; (b) *automation-induced surprises*, in which operators fail to understand what the computers are doing and why [8, 9], and finally (c) *complacency*, in which operators monitor processes less often than is required (or is optimal) [10, 11]. These problems tell us that human–machine interaction is not well designed even in modern processes created with highly advanced technologies.

This chapter discusses some of the issues that are at the center of designing human–machine coagency where humans and smart machines collaborate and cooperate sensibly in a situation-adaptive manner. One of the main topics in this chapter is the issue of authority and responsibility. It is argued that the machine may be given authority to improve safety and to alleviate possible damage to the human–machine system, even in a framework of human-centered automation. The second is the issue of the human operator's overtrust in and overreliance on automation, where it is argued that the possibilities and types of overtrust and overreliance may vary depending on the characteristics of the automated system. The importance of the design of a human–machine interface and human–machine interactions is included in the discussion.

12.2 Human Supervisory Control

The classical definition of human supervisory control of complex systems emerged in the 1960s during research on teleoperated lunar vehicles and manipulators [12]. A problem to consider in 1967 was the long delay between the ordering of a remote

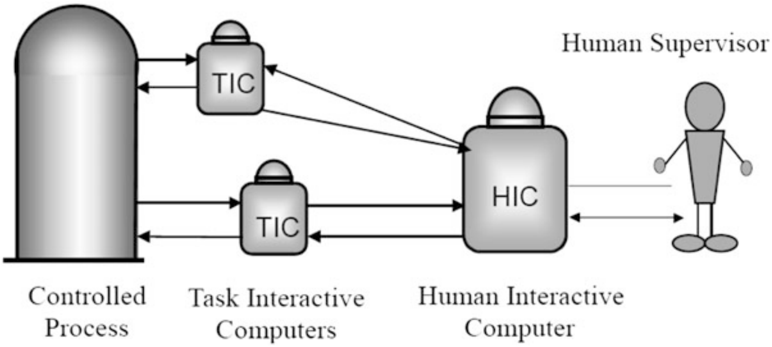


Fig. 12.1 Human supervisory control model

manipulator and the feedback from it. Instead, a supervisory control structure was proposed, where a remote computer would communicate with the manipulator, orchestrating intermediate events between orders (short-range goals) by remote closed-loop control. Back "on earth", the local computer handled the "supervisory loop", and the loop was closed by the operator. The local computer could also mimic predicted behavior of the manipulator feedback providing direct "quasi-feedback" [12].

There are many modern technical systems that are controlled by machine intelligence (or computers) under human supervision. Nuclear power plants, glass-cockpit aircraft, and computerized manufacturing systems are typical examples of such systems. These systems are neatly represented by a human supervisory control model [1].

The human supervisory control model distinguishes four units, as depicted in Fig. 12.1: (a) the human supervisor, (b) the human-interactive computer (HIC), (c) one or more task-interactive computers (TICs), and (d) the technical process to be controlled. The human supervisor decides what to do and issues commands to a HIC that has the capability to communicate with the human supervisor. The HIC understands high-level language to interpret directives given by the human supervisor, provides him/her with system state information in an integrated form, and issues decision aids or alert messages when appropriate. Upon receiving a supervisor's directive, the HIC issues the necessary commands to at least one TIC. The TIC then performs feedback control using its actuators and sensors.

Deciding what to do is one of the tasks that the human supervisor must perform. Sheridan [1] distinguished five phases of the human's supervisory control effort:

(i) Planning what needs to be done over some period of time and matching these requirements with available resources;
(ii) Teaching the computer what it needs to know to perform its assigned function for that time period;
(iii) Monitoring the automatic action to check that everything is proceeding as planned;

(iv) Intervening in the automatic action when necessary (such as in the case of an emergency situation or after completion of a planned task); and finally
(v) Learning from experience.

As can be imagined from (i) to (v), it may not be comfortable for the human operator to perform the assigned roles and tasks in human supervisory control. For instance, monitoring the automation and the controlled process, both of which are usually highly reliable, is often monotonous and wearisome. If something goes wrong, however, the human operator jumps into a highly stressful situation. He/she is requested to intervene in the process without any delay to prevent an anomaly from propagating into the process and/or the environment. At the same time, the human operator is not supposed to shut down a normal process based on a false alarm, which makes an intervention task very difficult and stressful.

12.3 Collaboration Failures Between Humans and Machines

Let us take, as an example, a highly automated aircraft. It is recognized that aviation automation has contributed to the improvement of aircraft safety. Nevertheless, aircraft incidents and accidents still occur. We realize that sometimes automation can result in incidents or accidents that were not possible in the "old days". Given below are a couple of examples.

(a) The automation (TIC) is strong enough to counteract effects caused by an anomaly occurring in the aircraft. However, the automation is sometimes *silent* [13]; it does not explicitly tell pilots how hard it is to control the aircraft. Pilots may thus often fail to recognize what is happening. An example of this is an in-flight upset incident in 1985, when a Boeing 747 aircraft dived 32,000 ft near San Francisco. The rightmost (#4) engine failed while flying at 41,000 ft on autopilot, and the aircraft began to suffer an undesirable yaw movement. The autopilot attempted to compensate the yaw movement by lowering the left wing; the rudder could not be used at the time. The pilots were busy focusing their attention on the decreasing airspeed. After some unsuccessful attempts to increase the airspeed, the captain finally decided to disconnect the autopilot so that he could fly the aircraft manually. Upon autopilot disconnection, the aircraft rolled to the right, nosed over, and dived steeply until the captain regained control of the aircraft at 9,500 ft. For further details of this incident, refer to ([5], p. 308), amongst others.
(b) The automation (HIC) may *surprise* pilots by doing what the pilots did not explicitly order. Suppose a pilot directed the HIC to do task A. The HIC may think that task B must be done simultaneously, and may perform both tasks without conveying its decision clearly to the pilot. The pilot would then be confused about the aircraft's behavior. They may say, "what is the autopilot

doing, and why is it doing that?" The crash of an Airbus A330 aircraft at Toulouse in 1994 is such an example. The accident occurred in a test flight to investigate the performance of the autopilot during an engine-out go-around. The pilot commanded the autopilot on at 6 s after takeoff. The goal of the autopilot was to climb to the 2,000 ft altitude that had already been set. The autopilot calculated at which point it had to activate the altitude acquisition transition mode (ALTSTAR) to achieve a smooth level-off. The calculation was done while both engines (the A330 is a two-engine aircraft) were operating perfectly and the aircraft was climbing very fast, at a vertical speed of 6,000 ft/min. Eight seconds after takeoff, the left engine was reduced to idle, to simulate an engine failure. At the same time, the autopilot activated the ALTSTAR mode, but the pilots did not realize the mode change. Under the simulated engine failure condition, the aircraft could climb at only 2,000 ft/min. To achieve the already calculated climb rate (6,000 ft/min), the autopilot continued pitching the aircraft up. Although the pilots realized that something was wrong, they could not understand what the autopilot was doing or why. Since there was no pitch limit in the ALTSTAR mode, the pitch angle reached 31.6°. At that stage, the captain disconnected the autopilot. It was, however, too late to regain control of the aircraft. For further details of this incident, see [14], amongst others.

12.4 Function Allocation

Suppose we need to design a human–machine system with specific missions or goals. We first have to identify functions that are needed to accomplish the goals. We then get to the stage of function allocation. *Function allocation* refers to the design decisions that determine which functions are to be performed by humans and which by machines. Various strategies for function allocation have already been proposed.

12.4.1 Traditional Strategies for Function Allocation

Rouse [15] classified traditional function allocation strategies into three types. The first category is termed *comparison allocation*. Strategies of this type compare the relative capabilities of humans versus machines for each function, and allocate the function to the most capable agent (either the human or the machine). The most famous MABA-MABA (what "men are better at" and what "machines are better at") list is possibly the one edited by Fitts [16], and reproduced in Table 12.1.

The second type is called *leftover allocation*. Strategies of this type allocate to machines any function that can be automated. Human operators are assigned the leftover functions for which no automation technologies are available.

The third type is *economic allocation*. Strategies of this type try to find an allocation that ensures economic efficiency. Even if some technology is available

Table 12.1 The fitts list

Humans appear to surpass present-day machines with respect to the following:

1. Ability to detect small amounts of visual or acoustic energy
2. Ability to perceive patterns of light or sound
3. Ability to improvise and use flexible procedures
4. Ability to store very large amounts of information for long periods and to recall relevant facts at the appropriate time
5. Ability to reason inductively
6. Ability to exercise judgment

Present-day (in the 1950s) machines appear to surpass humans with respect to the following:

1. Ability to respond quickly to control signals and to apply great forces smoothly and precisely
2. Ability to perform repetitive, routine tasks
3. Ability to store information briefly and then to erase it completely
4. Ability to reason deductively, including computational ability
5. Ability to handle highly complex operations, i.e., to do many different things at once

Taken from Fitts [16], Hancock and Scallen [17], and Price [18]

to automate a function, if the cost of automating the function is higher than that of hiring a human operator, the function is assigned to the operator.

Note here that the traditional strategies just described consider “who does what.” Such design decisions yield function allocations that are *static*. In other words, once a function has been allocated to an agent, the agent is responsible for the function at all times.

12.4.2 Static Function Allocations Are Not Always Appropriate

Suppose design decisions are made using either the leftover or the economic allocation strategies. These strategies do not reflect any human characteristics or viewpoints, and the resulting function allocation may be puzzling for operators. An operator may ask, “Am I meant to be responsible for this function, or is the automation?” Also, there is no guarantee that the allocations provide the operators with job satisfaction.

The comparison allocation may be better for the operators than either the economic or leftover allocation. However, the comparison allocation is not free from criticism either. Price [18] and Sharit [19] claimed that the list by Fitts is overly generalized and non-quantitative. Sheridan [6] pointed out that, “in order to make use of the Fitts MABA-MABA list, one needs data that are context dependent, but these data are mostly unavailable” (p. 59). He argued, referring to the ideas of Jordan [66], that “the idea of comparing the human with the machine should be thrown out but the facts about what people do best and what machines do best should be retained,” and “the main point of retaining the Fitts list is that people and machines are complementary” (p. 59). A *complementary strategy* can be seen in KOMPASS [20].

Even though operators are allocated only those functions in which people surpass machines, this superiority may not hold at all times and for every occasion. Operators may get tired after long hours of operation, or they may find it difficult to perform the functions under the given time constraints. This implies that "who does what" decisions are not sufficient; instead "who does what and when" considerations are needed for the success of function allocation, which means that function allocation must be dynamic.

12.4.3 Adaptive Function Allocation

Suppose that a human and a machine are to perform their 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. A scheme that modifies function allocation dynamically depending on the situation is called *adaptive function allocation*. The automation that operates under an adaptive function allocation is called *adaptive automation* [21–27].

Adaptive function allocation makes use of selected criteria to determine whether, how, and when functions need to be reallocated. The criteria reflect various factors, such as changes in the operating environment, loads or demands on operators, and performance of operators (see, e.g., [22, 27]).

Note that an active agent for a function may change from time to time in an adaptive function allocation. In such a case, it is said that the authority (for controlling the function) is traded from one agent to another. In other words, *trading of authority* means that either the human or the computer is responsible for a function, and an active agent changes alternately from time to time [1, 27].

Who makes the decision on trading of authority? More precisely, who decides whether the control of a function must be handed over and to which agent? Must a human operator decide, or may the machine (or the computer) decide? The former type is called the *human-initiated trading of authority*, and the latter the *machine-initiated trading of authority*. Which strategy is to be adopted is a hard problem to solve, as will be discussed later.

12.5 Machine Support for Human Information Processing

Four stages can be distinguished in human information processing: (a) perception, (b) situation understanding, (c) action selection, and (d) action implementation. It is well known that humans can fail in a variety of ways at each stage of information processing. Machines are supposed to provide humans with support at each stage. Parasuraman et al. [28] claimed that "the four-stage model of human information processing has its equivalent in system functions that can be automated," and they described human–machine interactions by distinguishing the following four classes of functions: (1) information acquisition, (2) information analysis, (3) decision and action selection, and (4) action implementation.

In order to understand how machines can perform these four classes of functions, let us consider two examples in aviation.

Example 1 The *traffic alert and collision avoidance system* (TCAS) is a family of airborne devices designed to help pilots avoid mid-air collisions [29]. Its functionality includes the following:

1. Information acquisition: The TCAS transmits interrogations at 1,030 MHz that transponders on nearby aircraft respond to at 1,090 MHz. By decoding the replies, the position and altitude of the nearby aircraft can be ascertained.
2. Information analysis: Based on the range, altitude, and bearing of a nearby aircraft, the TCAS performs range and altitude tests to determine whether the aircraft is a threat.
3. Decision and action selection: When the nearby aircraft is declared a threat, the 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: The TCAS issues a resolution advisory (RA) to inform the pilot of the appropriate avoidance maneuver. However, the TCAS does not perform any avoidance maneuvers itself.

Example 2 The *enhanced ground proximity warning system* (EGPWS) is designed to help pilots avoid a ground collision [30]. The functionality of this system is described as follows:

1. Information acquisition: The EGPWS collects air data, radio altitude, barometric altitude, and airplane position through various other systems, including the Flight Management System, GPS, and the airplane air data system.
2. Information analysis: Having received the above data, the EGPWS determines a potential terrain conflict by using its self-contained worldwide airport and terrain databases. The EGPWS displays the terrain as dotted patterns with colors indicating the height of the terrain relative to the current airplane altitude.
3. Decision and action selection: The EGPWS continuously computes terrain clearance envelopes ahead of the airplane. If these envelopes conflict with data in the terrain database, the EGPWS sets off alerts.
4. Action implementation: The EGPWS issues a caution-level alert approximately 40–60 s before a potential terrain conflict, and sets off a warning-level alert approximately 20–30 s before a conflict. However, the EGPWS does not perform any conflict avoidance maneuvers itself.

It is clear from Examples 1 and 2 that it is not the machine (TCAS or EGPWS) but the human pilot who implements a collision avoidance maneuver. Why is the action implementation stage not fully automated in these examples? The answer lies partly in the principles of human-centered automation.

12.6 Human-Centered Automation

12.6.1 Principles of Human-Centered Automation

Human-centered automation is an approach to realize a work environment in which humans and machines collaborate cooperatively (see, e.g., [4–6, 31–33]). Of the various application domains, it is aviation for which human-centered automation has been defined in the most detail. Aviation has a long history of automation and has experienced both its benefits and costs (see, e.g., [5, 34]). The principles of human-centered automation, given in Table 12.2, have resulted from studies to resolve the costs of automation, such as the out-of-the-loop performance problem, loss of situation awareness, complacency or overtrust, and automation surprises (see, e.g., [4, 5, 8, 13, 35–38]).

12.6.2 Domain-Dependence of Human-Centered Automation

Human-centered automation can be domain-dependent and thus must be established properly for each transportation mode: e.g., "human-centered automation for automobiles" can be quite different from "human-centered automation for aviation systems" as defined in Table 12.2. Such domain-dependence may stem from the quality of human operators and time criticality [39].

Quality of Human Operators The quality of human operators varies depending on whether they are professional or non-professional. Professional operators, such as airline pilots, are trained thoroughly and continuously so that their knowledge and skills are great enough to use smart and sometimes complicated machines correctly. On the other hand, in cases of non-professional operators, such as private

Table 12.2 Principles of human-centered automation in aviation

The human bears the ultimate responsibility for the safety of an aviation system
Therefore:
* The human must be in command
* To command effectively, the human must be involved
* To be involved, the human must be informed
* Functions must be automated only if there is a good reason for doing so
* The human must be able to monitor the automated system
* Automated systems must, therefore, be predictable
* Automated systems must be able to monitor the human operator
* Each element of the system must have knowledge of the others' intent
* Automation must be designed to be simple to learn and operate

car drivers, it would not be sensible to assume that their levels of knowledge and skills are high. Their understanding of the machine functionality could be incomplete, or even incorrect.

Example 3 Adaptive cruise control (ACC) systems are designed to reduce the driver's workload by freeing him/her from frequent acceleration and deceleration. Sometimes, these systems may be differentiated into two classes: high-speed range ACC and low-speed range ACC. When there is a leading vehicle to follow, both ACC systems control the speed of their host vehicle so that the time gap to the target vehicle may be maintained. Suppose the sensor loses sight of the target vehicle; the high-speed range ACC remains in its active state. In the case of low-speed ACC, the behavior differs depending on the control logic design. Two designs are possible. One allows the ACC to stay in its active state, while the other puts it into a standby state. It is hard to tell which design is better. Loss of mode awareness or automation surprises can occur in both design types, but in different ways. Inagaki and Kunioka [40] conducted an experiment with a PC-based driving simulator where no information was displayed regarding the state of the ACC. Subjects were requested to carry out procedures of perception, decision-making, and action implementation based on their mental models. Even after training or experience with the ACC systems on the simulator, loss of mode awareness and automation surprises were observed, which reflect the overtrust in and distrust of automation, and inertness of mental models.

Time Criticality Time criticality differs appreciably depending on the transportation mode. Consider the following examples in which an automated warning system is available to the operator.

Example 4 When a nearby aircraft is declared a threat, the TCAS selects an avoidance maneuver (to climb or descend) and issues an RA to inform the pilot of the appropriate avoidance maneuver (see Example 1). The estimated time to the closest point of approach is 15–35 s. The pilots are thus meant to respond to the RA within 5 s.

Example 5 Nowadays, certain types of automobile are equipped with a *forward vehicle collision warning system*. This system detects a vehicle in the front and measures its speed and the distance to it using a distance radar (mostly, a laser radar or a millimeter-wave radar) sensor mounted on the vehicle. If there is a possibility of collision with the vehicle in front, the system sets off a collision warning. The estimated time to collision is at most a few seconds.

As can be seen in the above examples, if the collision warning were against another aircraft, there would be sufficient time for the pilot to grasp the situation, validate the given warning, and initiate a collision avoidance maneuver. In the case of the automobile, however, time criticality is extremely high, and the driver has only a small amount of time to avoid a collision, as explained in Example 5.

12.7 Two Questions on Human-Centered Automation

The principles of human-centered automation given in Table 12.2 seem to be convincing. However, there are two questions that may not be easy to answer, namely: (1) Does the statement that, "The human must be in command," have to hold at all times and for every occasion? (2) What should the machine do if it detects inappropriate behavior or performance while monitoring the human? Is the machine only allowed to give warnings? Or, is it allowed to act autonomously to resolve the detected problem?

12.7.1 Who is in Charge and in Command?

Humans may not always be able to cope with the given situation. Consider the following example.

Example 6 The ITARDA (Institute for Traffic Accident Research and Data Analysis) analyzed data of automobile collisions that occurred in Japan during the period 1993–2001. Among all the collisions of four-wheeled vehicles, they extracted 359 head-on or rear-end collisions for which microscopic data were available with respect to the following: vehicle speed and location at which the driver perceived the possible danger, vehicle speed immediately before the collision, and vehicle speed at the time of the collision. They found that 13.9 % of the drivers tried to avoid the collision by steering and braking, 42.6 % by braking alone, and 5.6 % by steering alone. Surprisingly, 37.9 % of the drivers neither changed the steering direction nor applied the brakes [41].

How can we design a system that assists the driver when a collision is imminent? Consider the following two types of *advanced emergency braking system* (AEBS).

Example 7 (AEBS of type 1) The radar sensor monitors the vehicle in the front. When the system determines, based on the distance and relative speeds of the vehicles, that a collision is to be anticipated, it issues a warning to the driver and retracts the seatbelts.

Example 8 (AEBS of type 2) When the system determines that a collision is to be anticipated, it issues a warning to the driver and retracts the seatbelts, as in the case of type 1. When the system determines that a collision is imminent and that the driver is late in responding to the situation, it retracts the seatbelts firmly and applies an automatic emergency brake.

The AEBS of type 1 enhances the driver's situation awareness (SA). If the driver applies the brakes quickly enough, no collision will occur. However, if the driver fails to respond to the situation, the system provides no active help, and therefore a collision would be inevitable. The AEBS of type 2 has two layers of assistance: enhancement of the driver's SA, and trading of authority from the human to the machine, when appropriate. Trading of authority occurs in this case to support action implementation when the driver fails to take action at the right time. The system applies the emergency brakes, not based on the driver's directive, but based on its own decision. Note that one of the principles in Table 12.2, "the human must be in command," is violated here. However, that does not necessarily mean that an AEBS of type 2 should not be allowed. On the contrary, Example 6 suggests the need for machine-initiated trading of authority in emergencies.

Professional operators may also fail to respond appropriately to the situation encountered, as shown in the following example.

Example 9 An analysis of *controlled flight into terrain* incidents of commercial jet airplanes during the period 1987–1996 found that 30 % of the accidents occurred when the traditional *ground proximity warning system* (GPWS) failed to detect terrain ahead, while 38 % were due to late warning of the GPWS or improper pilot response [30].

Problems of "no warning" or "late warning" may be resolved by introducing the EGPWS (see Example 2), which enhances the pilot's understanding of the height of the terrain relative to the aircraft altitude. The problem of "pilot's late response" may not be fully resolved by the EGPWS, since it is the human pilot who is responsible for a collision avoidance maneuver. However, there is a system in which the collision avoidance maneuver is initiated automatically. The automatic ground collision avoidance system (Auto-GCAS) for combat aircraft is such an example [42]. When a collision with the terrain is anticipated, the system gives a pull-up warning. If the pilot takes aggressive collision avoidance action, the system does not step in any further. If the pilot does not respond to the warning, the system takes over control from the pilot and executes an automatic collision avoidance maneuver. When no further threatening terrain is found, the system returns control back to the pilot. Thus, the Auto-GCAS determines when to intervene and when to return command of the aircraft back to the pilot.

Even in case of aircraft, a machine-initiated action implementation has been playing important roles in relieving the pilot's physical and/or mental load. One of classical example is the *mach-trim system*. When an aircraft flies at a high speed, it receives a pitch-up moment and thus the control column needs to be pushed forward to maintain the altitude. If airspeed becomes higher than a certain value, aircraft obtains a pitch-down moment and thus the control column must be pulled back, which caused difficulty in old days in maneuvering aircraft. Now the mach-trim system handles the problem without human intervention and creates positive stability for the aircraft.

Another example may be the *thrust asymmetric compensation* (TAC) system on the twin-engine Boeing 777. The TAC helps the pilot to cope with the yawing effect when an engine fails during the takeoff role. When the TAC senses that the thrust levels differ 10 % or more between the two engines, it initiates rudder control automatically so as to minimize the yaw and to make it possible for the pilot to center the control column.

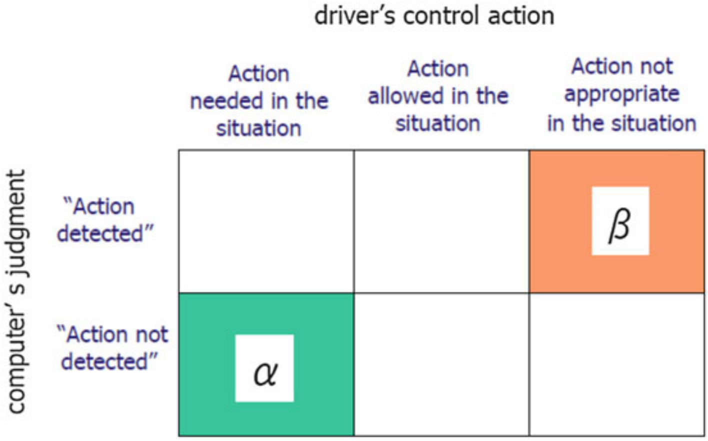
More recent examples can be found in the A350. Airbus Corporation is trying to develop an automatic system that, upon a TCAS RA, initiates an appropriate maneuver to steer the aircraft away from a potential mid-air threat without input from the flight crew [43]. Airbus is also considering equipping the A350 with an automatic system that would provide a warning to the flight crew when unsafe cabin pressure is detected. If the crew does not cancel the warning or take positive control of the aircraft, the system performs an automatic side-step maneuver to the right of the designated airway to avoid conflict, and then puts the aircraft into a rapid descent at maximum operating speed [44]. These classical and new examples show that automatic action implementation based on the machine's decision is of value even in the aviation domain.

12.7.2 What if the Machine Finds that the Human's Action is Inappropriate?

A human's control action or directive to the machine may be classified into three categories: (1) a control action that needs to be carried out in a given situation; (2) a control action that is allowable in the situation and thus may either be done or not done; and (3) a control action that is inappropriate and thus must not be carried out in the situation. Assuming some sensing technology (or machine intelligence, provided by a computer), two states may be distinguished for each control action: (a) "detected," where the computer determines that the human is performing the control action, and (b) "undetected," in which the control action is not detected by the computer.

Figure 12.2 depicts all possible combinations of a control action and its state. Among these, case α shows the situation where the computer determines that the human operator is (too) late in performing or ordering a control action that must be carried out in the given situation. A typical example of case α in the automobile domain is that, in spite of rapid deceleration by the leading vehicle, a following driver does not apply the brakes owing to some distraction. Case β indicates a situation where the computer determines that the human operator has misunderstood the given circumstances and the control action that he/she is taking or has requested does not suit the situation. A typical example of case β is when a driver is about to change the steering direction to enter an adjacent lane without noticing that a faster vehicle is approaching from behind in that lane.

A question that must be asked for case α is whether the computer should be allowed to initiate without human intervention the control action (such as applying the brakes) that the human should have taken, or whether the computer is allowed only to set off a warning to urge the human to perform manually the control action that the situation requires. A question asked for case β is whether the computer should be allowed to prohibit the control action (such as altering the steering direction to make a lane change) that the human is trying to do, or whether the



driver's control action

computer' s judgment	Action needed in the situation	Action allowed in the situation	Action not appropriate in the situation
"Action detected"			β
"Action not detected"	α		

Fig. 12.2 Control actions in a given situation

computer is allowed only to set off a warning to tell the human that his/her action should be stopped immediately.

Suppose the computer always knows what control action is appropriate, besides just detecting (or not) whether a control action has been taken. Then it would be almost obvious that the computer should be allowed to initiate the control action that the human failed to perform in case α, and to prohibit the human's control action that does not suit the given circumstances in case β, considering the following facts: (1) humans do not always respect or respond to warnings, and (2) humans need a certain amount of time to interpret the warnings and thus a time delay is inevitable before effective actions can be taken. It is, however, too optimistic to assume that the computer never makes an error in judging whether the human's response to the situation is inappropriate. Inagaki and Sheridan [45] have analyzed in a probability theoretic manner the efficacy of the computer's support in both cases α and β, under a realistic setting that the computer's judgment may be wrong. They have proven that the computer should be allowed to act autonomously in certain situations via machine-initiated trading of authority based on its decision.

There are some studies that have analyzed the efficacy of the computer's support for cases α and β by conducting cognitive experiments [46–48]. The question posed in these studies was, "What type of support should be given to a car driver when it is determined, via some sensing and monitoring technologies, that the driver's situation awareness may not be appropriate to a given traffic condition?" For cases α and β in Fig. 12.2, two types of driver support were compared: (a) warning-type support in which an auditory warning is given to the driver to enhance SA, and (b) action-type support in which an autonomous safety control action is executed to avoid a collision. Although both types of driver support were effective, the investigators also observed some problems.

The warning-type driver support is fully compatible with human-centered automation, because the driver always retains final authority over the automation. Most drivers who participated in the experiments in [46–48] accepted the warning-type support for both cases α and β. However, the warning-type of driver support sometimes failed to prevent a collision when the driver did not respect the warning. A driver's typical and 'reasonable' disregard for a correct warning occurs when the warning is based on an object that is invisible to the driver. This fact suggests a limitation of a purely human-centered automation design in which the human remains the final authority at all times and for every occasion.

The machine-initiated action taken for case α may be straightforward, viz., merely implement the control action that the human failed to perform in a timely manner. For case β, machine-initiated control actions are classified into two groups: (a) *hard protection*, in which the human is not given authority to override the computer's corrective control action initiated based on its judgment that "the human's action does not suit the situation"; and (b) *soft protection*, in which the human is given authority to override the computer's corrective control action, even though the computer has determined that "the human's action does not suit the situation".

It is reported in [46, 48] that action-type support with hard protection characteristics sometimes failed to receive *acceptance* from drivers, although it was successful in collision prevention. The most prominent reason for this lack of acceptance is as a result of the hard protection characteristic in cases when there is a conflict of intention between the human and the computer. The soft protection type action support may also fail to prevent a collision from occurring, especially when the driver misinterprets why protective action has been triggered and for which object. It may not be sensible to blame the drivers even though they 'interpret' a given situation incorrectly, because they might have to interpret the situation based on limited information collected within a limited time period. An issue that arises is how to design a human–machine interface and interaction for cases when something is invisible to the driver, yet visible to the machine.

12.8 Overtrust and Overreliance

If machines are capable of correcting and preventing 'erroneous or inappropriate' behavior of the human operator, as discussed in Sect. 12.7, the human operator will trust and rely on the machines. Problems may occur if the human operator places too great a trust in or reliance on the machine without learning or knowing its limitations. This section presents a theoretical framework for describing and analyzing the human operator's overtrust in and overreliance on smart machines, and illustrates the framework with examples of assistance systems for car drivers [49].

12.8.1 Overtrust

Overtrust in an advanced driver assistance system (ADAS) is an incorrect diagnostic decision to conclude that the assistance system is trustworthy, when it actually is not. This section gives two axes for discussing overtrust in the assistance system. The first axis is the *dimension of trust* and the second the *chance of observations*.

Dimension of Trust The first axis describes in what way the driver can overrate trust. Lee and Moray [50] distinguished four dimensions of trust: (a) foundation, representing the fundamental assumption of natural and social order; (b) performance, resting on the expectation of consistent, stable, and desirable performance or behavior; (c) process, depending on an understanding of the underlying qualities or characteristics that govern behavior; and (d) purpose, resting on the underlying motives or intents. Three types of overtrust can be distinguished depending on which of the dimensions, (b) through (d), is overrated; the first dimension, (a), is usually met in the case of an ADAS.

Overrating dimension (b) can be explained by the following thought pattern of a driver: "The assistance system has been responding perfectly to all the events that I have encountered so far. Whatever event may occur, the system will take care of it appropriately." Improper evaluation of dimension (c) is seen in the case where a driver who has been using an assistance system without having read the user manual is thinking, "It would be quite alright even if I do not know the details of how the system functions." Overestimation of dimension (d) is illustrated by the case where a driver believes that "I do not understand why my assistance system is doing such a thing. However, it must be doing what it thinks is necessary and appropriate."

Chance of Observations The second axis for investigating overtrust describes how often the driver can see the assistance system functions. The chance of observations affects the ease with which a mental model of the assistance system is constructed. The possibility of a driver's overtrust can differ depending on whether the assistance system is for use in normal driving or in an emergency.

Consider the ACC as an example of an assistance system to reduce driver workload in normal driving. Based on the large number of opportunities to observe the ACC functioning repeatedly in daily use, it would be easy for the driver to construct a mental model of the ACC. If the driver has been satisfied by the 'intelligent' behavior of the ACC, it would be natural for him/her to place trust in the assistance system. However, the trust can sometimes be overtrust. Suppose the driver encounters a new traffic condition that is seemingly similar to a previous one, but is slightly different. By expecting that the ACC should be able to cope with the situation without any intervention by the driver, the driver could be overestimating the functionality of the ACC.

Next, consider the AEBS as an example of an assistance system activated only in an emergency to ensure the driver's safety. It would be rare for an ordinary driver to see the AEBS working, and he/she may not be able to construct a complete mental model of the system owing to the limited number of chances to experience the

AEBS working. Drivers may have been told (by the car dealer, for instance) that the AEBS would be activated automatically in an emergency. However, they may not be fully convinced because of the lack of opportunities to observe for themselves that the AEBS works correctly and consistently when necessary.

12.8.2 Overreliance

Overreliance on an ADAS is an incorrect action selection decision determining to rely on the assistance system by placing overtrust in it. Regarding overreliance on automated warning systems, there are relevant studies in the aviation domain (see, e.g., [36, 51–53]). Suppose that the automated warning system almost always alerts the human when an undesirable event occurs. Although it is possible for a given alert to be false, the human can be confident that there is no undesirable event as long as no alert is given. (A similar situation can occur in the automobile domain when the driver is provided with a communication-based alert from the road infrastructure to let the driver know of an approach or existence of cars on a crossroad behind some buildings). Meyer [52] used the term ‘reliance’ to express this response by the human. If the human assumed that the automated warning system would always give alerts when an undesirable event occurred, the human’s thinking would constitute overtrust in the warning system, and the resulting reliance on the warning system would be overreliance. The definition of overreliance on the ADAS, given at the beginning of this section, is a generalization of that of overreliance on warning systems in previous studies in the sense that the goal of the assistance system is not only to set off warnings but also to execute control actions.

Two axes are provided for overreliance on assistance systems. The first axis represents the *benefits expected* and the second the *time allowance for human intervention*.

Benefits Expected The first axis describes whether the driver can derive some benefit by relying on the assistance system. Suppose the driver assigns the ACC all the tasks for longitudinal control of the vehicle. This may enable the driver to find time to relax his/her muscles after stressful maneuvering, or to allocate cognitive resources to finding the correct route to the destination in complicated traffic conditions. In this way, relying on the assistance system sometimes yields extra benefit to the driver, when the system is used in normal driving.

The discussion can be quite different in the case of the AEBS. The AEBS is activated only in an emergency, and the time duration for the AEBS to fulfill its function is very short, say only a few seconds. It is thus not feasible for the driver to allocate the time and resources, saved by relying on the AEBS, to something else to derive extra benefit from only a few seconds. A similar argument may apply to other assistance systems designed for emergencies.

Time Allowance for Human Intervention The second axis, time allowance for human intervention, describes whether the driver can intervene in the assistance system's control if the driver determines that the system performance differs from what he/she expected. In the case of the ACC, it is not difficult for the driver to intervene to override the ACC if its performance is not satisfactory. However, in the case of the AEBS, it would be unrealistic to assume that the driver could intervene in the control by the AEBS if he/she decides that the AEBS's performance is not satisfactory, because the whole process of monitoring and evaluating the AEBS's performance as well as the decision and implementation of intervention must be done within a few seconds.

12.8.3 From Collision Damage Mitigation to Collision Avoidance

Based on the framework given in Sects. 12.8.1 and 12.8.2, the design guidelines for the AEBS were revised in 2011 by a task force of the Advanced Safety Vehicle (ASV) project, Ministry of Land, Infrastructure and Transport (MLIT), Japan.

If the host vehicle is approaching the leading vehicle relatively quickly, most AEBSs first tighten the seatbelt and add a warning to urge the driver to apply the brakes. If the AEBS determines that the driver is too late in braking, it applies the brake automatically based on its decision. However, in Japan the AEBS has been implemented as a *collision damage mitigation system*, instead of a *collision avoidance system*. Behind the design decision to 'downgrade' the AEBS, there has been concern among the regulatory authorities that "if an ADAS were to perform every safety control action automatically, the driver would become overly reliant on the assistance system, without paying attention to the traffic situations himself or herself."

Although the above 'concern' seems to be reasonable, there have been some discussions in the ASV project that more precise investigations are necessary so as not to lose opportunities for drivers (especially elderly drivers) to benefit from an assistance system that would back them up or even override them when appropriate. The MLIT set up a task force in December 2009 in the ASV project to investigate future directions for driver assistance in the light of (1) driver's overtrust in and overreliance on the ADAS, and (2) authority and responsibility between the driver and the automation.

The following argument was made by the ASV task force: "Since the AEBS is activated only in cases of emergency, it would be very rare for an ordinary driver to see how the system works (i.e., chance-of-observation axis). It is thus hard for the driver to construct a precise mental model of the AEBS, and may be hard for him/her to engender a sense of trust in the system (i.e., dimension-of-trust axis). However, it is known that people may place inappropriate trust (i.e., overtrust) without having any concrete evidence proving that the object is trustworthy.

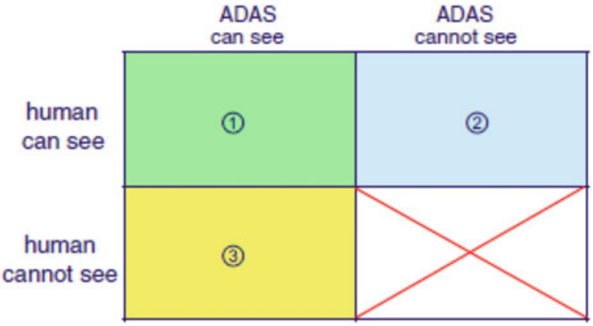
Now, let us assume that the driver places overtrust in the assistance system. We have to ask whether the driver may rely on the system excessively (i.e., overreliance). In case of AEBS, even if the driver noticed that the system's behavior was not what was expected, no time may be left for the driver to intervene and correct it (i.e., time allowance for human intervention). In spite of that, does the driver rely on the AEBS and allocate his/her resource to something else at the risk of his/her life (i.e., benefits expected)? The answer would be negative."

The ASV task force approved the above argument and decided that the AEBS should be developed as a collision avoidance system, instead of a collision damage mitigation system. The task force investigated design requirements for such a collision avoidance AEBS so that it would not interfere with the driver's own actions (by ensuring that it applied the automatic brakes at the latest time possible), but would still effectively avoid a collision with an obstacle ahead. Human factor viewpoints played major roles in determining the design requirements for the timing of the AEBS to initiate automatic emergency braking and its deceleration rate. In fact, these were determined by analyzing drivers' braking behavior in normal and critical traffic conditions. Moreover, a couple of conventional requirements for the AEBS were abolished from the human factors viewpoints (e.g., to reduce mode confusion or automation surprise). Based on the conclusions of the ASV task force, the MLIT has been revising the design guidelines for the AEBS. The new guidelines will be announced to the public in 2012.

12.9 Combination of Agents with Limited Capabilities

Suppose we are trying to design an ADAS that *makes the invisible visible* by providing information or images of objects or events that the driver cannot see directly, gives an alert when the driver is late in responding to the situation, and provides control inputs to the host vehicle when necessary. It is not an easy task to implement an ADAS that can cooperate well with the human driver. Coagency between the driver and an ADAS would fail quite easily if the interface and interaction were not appropriate. Let us consider an ensemble of a human driver and an ADAS, where each of the agents has limited ability. Figure 12.3 illustrates the combinations of limited capabilities of the driver and the ADAS [54].

Region 1 depicts the situation where both agents can see the objects. However, an automation surprise can occur if what the driver sees is different from what the ADAS sees and if the driver fails to notice this. Figure 12.4 illustrates a situation where the driver was astonished by the ACC's acceleration; the driver had expected the ACC to slow down in response to the deceleration of the leading vehicle A that the ACC had been following thus far. This case represents an actual experience by the author. A probable cause of the acceleration by the ACC could be that the ACC detected vehicle B and switched the target vehicle from A to B. In the author's host vehicle, the instrument panel displayed an indication implying that the ACC was



	ADAS can see	ADAS cannot see
human can see	①	②
human cannot see	③	×

Fig. 12.3 Combination of agents with limited capabilities

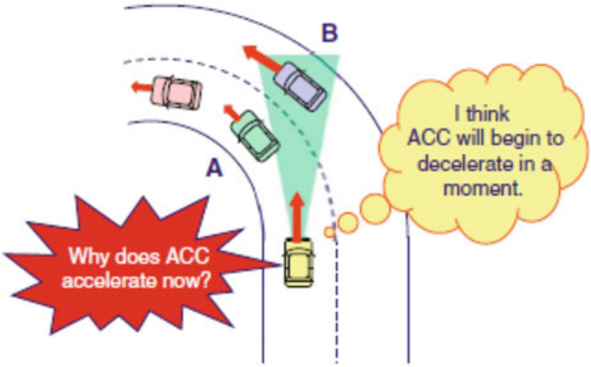

B
A
I think ACC will begin to decelerate in a moment.
Why does ACC accelerate now?

Fig. 12.4 What the driver see ≠ what the ADAS sees

following ‘a target vehicle.’ It did not explain which was the target vehicle or that the target vehicle had indeed changed.

Even when what the driver sees is exactly the same as what the ADAS sees, problems may occur if the way of thinking differs between the two agents. A typical case is illustrated in Fig. 12.5, where one agent tries to avoid a collision into an object by braking, while the other agent avoids this by changing the steering direction.

Region 2 depicts the situation where the ADAS cannot see the object that the driver sees. A typical case is shown in Fig. 12.6. While following vehicle A with the ACC, the driver noticed that vehicle B in the adjacent lane might be cutting in just ahead. The driver then expected the ACC to initiate deceleration shortly. However, the ACC did not decelerate. On the contrary, it accelerated. This case illustrates another of the author’s actual experiences. The reason for the surprising behavior by the ACC is that the ACC did not sense vehicle B because it was outside the sensor range, and accelerated in response to the acceleration of vehicle A. No human interface is currently available to visualize a sensor’s range, which means that there are no clues to help the driver recognize the extent of what the ACC can see.

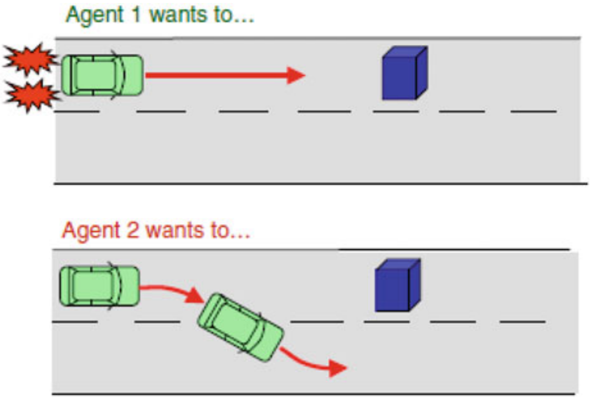


Fig. 12.5 Conflict of intentions

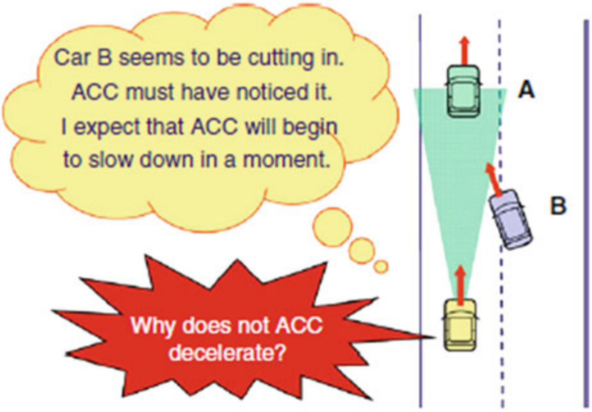

Car B seems to be cutting in. ACC must have noticed it. I expect that ACC will begin to slow down in a moment.
Why does not ACC decelerate?
A
B

Fig. 12.6 Failure to recognize limit of capability

Region 3 represents the situation where the ADAS can see what the driver cannot see. The support offered to the driver by the ADAS aims to make the invisible visible. A typical case is depicted in Fig. 12.7, where the driver tries to find the right moment to enter a through street. It is hard for the driver to check whether cars are approaching from the right because the building on the corner blocks the driver's view to the right. The ADAS provides the driver with an alert message, "A car coming from right!" based on the information obtained through vehicle-to-vehicle or vehicle-to-infrastructure communication. If the human interface is poorly designed, it may not be easy for the driver to understand whether the alert was aimed at vehicle A or B in Fig. 12.8. If the alert is given too early and the driver has experienced a *false alert* before, he/she may suspect that the given alert is yet another false alert (Fig. 12.9).

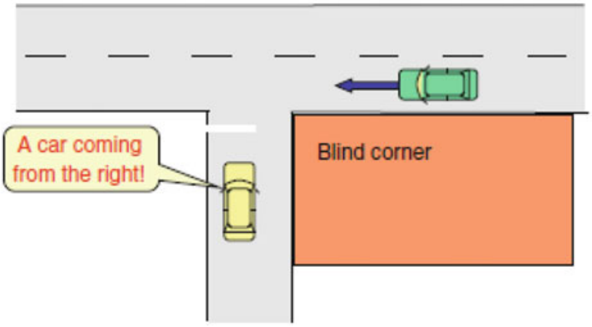


Fig. 12.7 Proximity warning through vehicle-to-vehicle communication

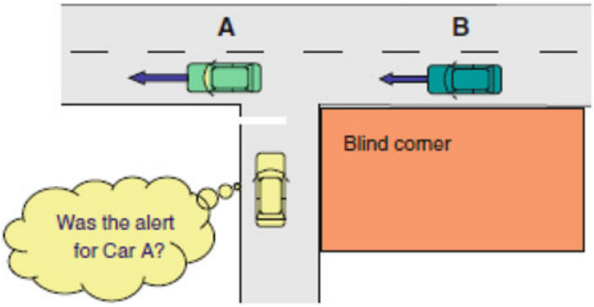


Fig. 12.8 Ambiguity caused by an imprecise interface

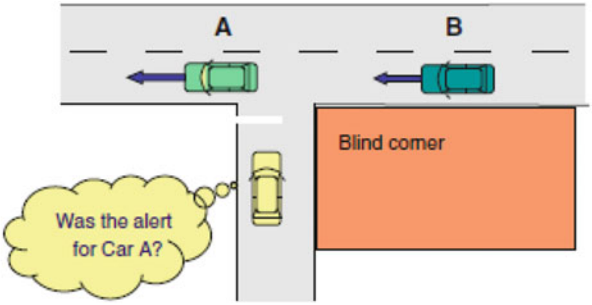


Fig. 12.9 Warning may be disregarded without any validation

The difficulties in the above cases stem from the fact that the driver cannot see the car (or cars) that the alert is attempting to highlight. No concrete evidence is available for the driver to validate the alert. All that the driver can do is either believe in the alert (i.e., wait at the intersection for a while) or ignore the alert (i.e., turn into the through road). This is an issue of *trust*; the driver's attitude toward the alert may be judged in hindsight as appropriate trust, overtrust, or distrust.

12.10 Viewpoints for Designing Human–Machine Coagency

Let us discuss how we should design the functionality to assist human operators appropriately and in a context-dependent manner. Discussions should consider two aspects: enhancement of situation awareness, and design of authority.

12.10.1 Enhancement of Situation Awareness

Human interface design is a central issue for enhancing situation awareness, avoiding automation surprises, and establishing appropriate trust in automation. The implemented human interface must enable the human to: (1) recognize the intention of the automation, (2) understand why the automation thinks what it does, (3) share the situation awareness with the automation, and (4) show the limits of the functional abilities of the automation.

Enhancement of situation awareness corresponds well with the human-centered automation concept, in which the *human locus of control* is claimed. However, as noted earlier, non-professional operators may not be able to cope with the given situation. Even professional operators may not respond to the situation appropriately; recall the mid-air crash on July 1, 2002, in which two TCAS-equipped aircraft collided over southern Germany [55, 56]. When a conflict developed between the 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, despite its TCAS instructing the pilot to climb, thus causing the mid-air collision. As described by Example 1 in Section 12.5, the TCAS is not given any authority to force a pilot to follow its resolution advisory.

12.10.2 Design of Authority

Human-computer interactions can be described in terms of the *level of automation* (LOA). Table 12.3 gives an expanded version, in which a new LOA appears between levels 6 and 7 in the original list by Sheridan [1]. The added level, called level 6.5, was first introduced in [57] to avoid automation surprises induced by automatic actions, when the actions are indispensable in ensuring system safety in emergencies.

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

Table 12.3 Scales of levels of automation (expanded version)

1.	The computer offers no assistance; the 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 after telling the human what it is going to do, or
7.	executes automatically, and then necessarily informs the human
8.	informs the human after execution only if he/she asks
9.	informs the human after execution if it, the computer, decides to
10.	The computer decides everything and acts autonomously, ignoring the human

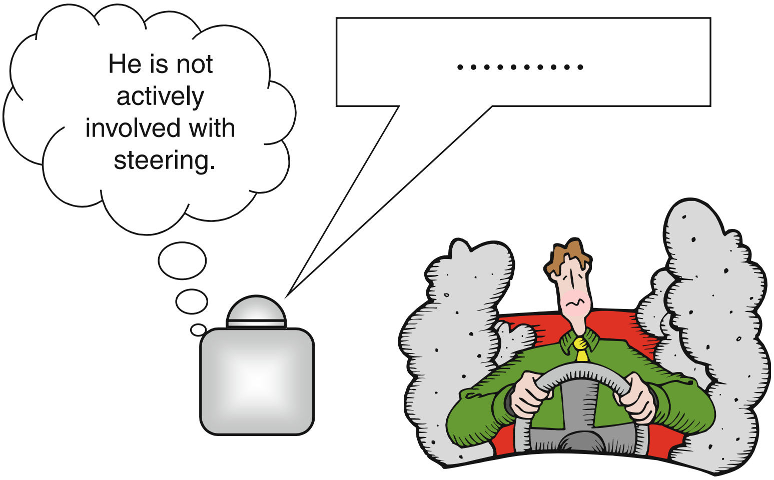


Fig. 12.10 What should the computer say to the minimally involved driver?

Example 10 Suppose a man is driving a car in which the lane keeping assistance (LKA) is operational. LKA is a system that recognizes the lane through image processing technology and provides the driver with assisted steering torque to keep the car in the center of the lane. Suppose the computer determines, by monitoring moment-to-moment steering torque, that the driver has not been actively involved in the steering task for a while. The computer decides it is appropriate to return the steering task to the driver. How should the computer return the steering task to the driver, and what should the computer say to the driver in this situation? (Fig. 12.10).

There are several alternatives for the computer's message (or action) in the above situation. The simplest alternative would be for the computer to mention to 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, either if he disagrees with the comment, or if he failed to catch the message due to drowsiness.

The second alternative would be for the computer to make a more explicit suggestion, by saying, “Shall I let you drive yourself?” The LOA of this strategy is set at level 5. If the driver does not reply, the computer cannot do anything further, and the lane-keeping task will still be performed by the automation.

The third alternative would be for the computer to give 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 is too slow in responding to the message within the allowed time, the computer puts the LKA into its standby state. Then the driver has to take over control even if he/she does not wish to do so.

The fourth alternative would be for the computer to give the following message after it has deactivated the LKA: “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/she was not ready to take over control from the automation.

The most extreme case would be for the computer to hand over control to the driver *silently*. The LOA of this strategy is set at eight or higher. In other words, the computer says nothing to the driver, even though it has already put the LKA into its standby state. Suppose the car approaches a lane boundary some time later. The driver may expect the LKA to steer the wheel appropriately, because he is under the impression that the automation is still in its active mode. The driver would be very surprised to see that the lane boundary continues approaching contrary to expectations.

As the above example illustrates, if the LOA is not chosen appropriately, some undesirable event may occur. In designing human–machine systems, it is important to predict how the design will affect humans and change their behavior [67].

There are three approaches that are useful in selecting an appropriate LOA, each of which is illustrated with an example below.

Selection of an Appropriate LOA via Theoretical Analysis Suppose an engine fails during the takeoff roll of an aircraft. The pilot must decide whether to continue the climb-out (Go) or to abort the takeoff (No-Go). The standard decision rule for 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 reached. The critical speed V1 is called the *takeoff decision speed* at which the pilot must apply the first retarding means in the case of a No-Go. Inagaki [59] has proven mathematically, based on 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 of the 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 the alert is correct, and “f” that the alert is false. In addition to the 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”, that is, the pilot hesitates in deciding whether the alert is correct or 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 to be 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 owing to an incorrect No-Go decision. (iii) An overrun accident is caused by an inappropriate RTO (rejected takeoff) action in excess of V1.

The conditional expected loss, E[Z | engine failure alert], was evaluated for each case in which a Go/No-Go 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 the engine failure alert is issued.

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

Phase 2. An engine failure alert is issued before, but close to V1. An RTO can be initiated before V1 if the human responds without any hesitation. We have $L_{DP} \leq L_{TP}$. 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 [58, 59].

Selection of an Appropriate LOA via Cognitive Experiments Another important result in Inagaki [59] is that for a human pilot to be in authority at all times and for every occasion, design of the human interface needs to be changed so that more direct information, such as “Go” or “Abort” messages, can be given explicitly to the human pilot. With the human interface, we have $L_{AS} = L_{DP} = L_{TP}$ in Phase 4.

A flight simulator for a two-engine aircraft has been implemented, and a cognitive experiment with a factorial design, mapping onto (Control mode) × (Phase) × (Human interface design) was conducted. For the control mode, a manual (M) control mode and a 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 an appropriate LOA for decision and control, and may take over control to continue the takeoff if it decides that it is not possible for humans to initiate the RTO before V1 is reached.

Experimental results show that even though the human interface, with the ability to give “Go” and “Abort” messages, was effective in allowing a correct decision to be made, some overrun accidents did occur under M-mode. Under SAA-mode, on the other hand, no overrun accidents occurred [60].

Selection of an Appropriate LOA via Computer Simulations Suppose a human is driving with the ACC and LKA operational on the host vehicle. While observing that the automation behaves correctly and appropriately, it is natural for the driver to trust the automation. Sometimes he/she may place excessive trust in the automation. In such cases, the driver may fail to focus his/her attention on the driving environment, and may pay attention inappropriately to some non-driving tasks (such as using a mobile phone, manipulation of the on-board audio system, and so on). Suppose the ACC recognizes that the deceleration rate of the target vehicle is much greater than the maximum deceleration rate with which the ACC can cope using the ordinary automatic brake. Which of the following design alternatives are appropriate?

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

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

Scheme 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}.

Based on discrete-event models for dynamic transition of driver’s psychological states and driving environments, Monte Carlo runs were performed to analyze the complacency effect and compare the efficacy of schemes 1–3. It was observed that when the driving is peaceful and the ACC continues to operate successfully in its longitudinal control, the driver is more likely to rely on the ACC, and his/her vigilance deteriorates. If the target vehicle decelerates rapidly in such cases, the driver needs time to recognize what is happening and thus may not be able to cope with the situation in a timely manner, even if an emergency-braking alert has been given. The number of collisions under LOA-4 was significantly higher than that under either LOA-6 or LOA-6.5. A higher LOA is more effective in ensuring car safety under time criticality, especially when the driver has been inattentive [61].

12.11 Concluding Remarks

This chapter discussed the issue of authority and responsibility as well as overtrust in and overreliance on the ADAS. It is not easy to obtain a clear answer for any of these issues. The arguments in this paper suggested that the phrase ‘human-centered automation’ may be misleading. Sheridan [6] distinguished 10 different meanings of human-centered automation, while Hollnagel and Woods [38] stated that "human-centeredness is an attractive but ill-defined concept" (p. 128). Human-centered automation has been developed to resolve various costs incurred by careless introduction of automation (viz., technology-centered automation). However, the term 'human-centeredness' can conjure an image of a 'humans versus machines' structure that tries to claim that the final authority is given only to the human and requires that the machine holds a subordinate position to the human.

An important viewpoint is that "humans and machines are 'equal' partners", and this seeks human–machine coagency "by shifting the focus from human and machine as two separate units to the joint cognitive system as a single unit" ([38], p. 67). If we seek human–machine cooperation toward common goals under the recognition that the human and machine have their own limitations, it would not be wise to assume strictly that "the human must be in command". Therefore, should not the term 'function-centeredness' [62] replace the phrase 'human-centeredness' in order to express human–machine coagency in the form of *function congruence* [63], where functions are distributed among agents by taking into account "the dynamics of the situation, specifically the fact that capabilities and needs may vary over time and depend on the situation" ([63], p. 44), or in the form of situation-adaptive autonomy where the human and the machine trade authority dynamically depending on the situation [64, 65]?

References

1. Sheridan TB (1992) Telerobotics, automation, and human supervisory control. MIT Press, Cambridge, MA
2. Bainbridge L (1983) Ironies in automation. Automatica 19(3):775–779
3. Rasmussen J, Goodstein LP (1987) Decision support in supervisory control of high-risk industrial systems. Automatica 23(5):663–671
4. Woods D (1989) The effects of automation on human's role: experience from non-aviation industries. In: Norman S, Orlady H (eds) Flight deck automation: promises and realities, NASA CR-10036. NASA-Ames Research Center, Moffett Field, pp 61–85
5. Billings CE (1997) Aviation automation—the search for a human-centered approach. LEA, Mahwah
6. Sheridan TB (2002) Humans and automation: system design and research issues. Human Factors and Ergonomics Society & Wiley, Santa Monica
7. Endsley MR (1995) Towards a theory of situation awareness in dynamic systems. Hum Factors 37(1):32–64
8. Wickens CD (1994) Designing for situation awareness and trust in automation. In: Proceedings of IFAC integrated systems engineering, Baden-Baden, Germany, pp 77–82
9. Sarter NB, Woods DD, Billings CE (1997) Automation surprises. In: Salvendy G (ed) Handbook of human factors and ergonomics, 2nd edn. Wiley, New York, pp 1926–1943
10. Parasuraman R, Molloy R, Singh IL (1993) Performance consequences of automation-induced 'complacency. Int J Aviat Psychol 3(1):1–23
11. Moray N, Inagaki T (2000) Attention and complacency. Theor Issues Ergon Sci 1(4):354–365
12. Ferrell WR, Sheridan TB (1967) Supervisory control of remote manipulation. IEEE Spectr 4 (10):81–88

13. Sarter NB, Woods DD (1995) How in the world did we ever get into that mode? Mode error and awareness in supervisory control. Hum Factors 37(1):5–19
14. Dornheim M (1995) Dramatic incidents highlight mode problems in cockpits. Aviat Week Space Technol 142(5):57–59
15. Rouse WB (1991) Design for success: a human centered approach to designing successful products and systems. Wiley, New York
16. Fitts PM (ed) (1951) Human engineering for an effective air-navigation and traffic-control system. The Ohio State University Research Foundation, Columbus
17. Hancock PA, Scallen SF (1998) Allocating functions in human-machine systems. In: Hoffman RR et al (eds) Viewing psychology as a whole. American Psychological Association, Washington, DC, pp 509–539
18. Price HE (1985) The allocation of function in systems. Hum Factors 27(1):33–45
19. Sharit J (1997) Allocation of functions. In: Salvendy G (ed) Handbook of human factors and ergonomics, 2nd edn. Wiley, New York, pp 301–339
20. Grote G, Ryser C, Wafler T, Windischer A, Weik S (2000) KOMPASS: a method for complementary function allocation in automated work systems. Int J Hum-Comput Stud 52:267–287
21. Rouse WB (1988) Adaptive aiding for human/computer control. Hum Factors 30(4):431–443
22. Parasuraman R, Bhari T, Deaton JE, Morrison JG, Barnes M (1992) Theory and design of adaptive automation in aviation systems, Progress report no NAWCADWAR-92033-60. Naval Air Development Center Aircraft Division, Warminster, PA
23. Scerbo MW (1996) Theoretical perspectives on adaptive automation. In: Parasuraman R, Mouloua M (eds) Automation and human performance. LEA, Mahwah, pp 37–63
24. Moray N, Inagaki T, Itoh M (2000) Adaptive automation, trust, and self-confidence in fault management of time-critical tasks. J Exp Psychol Appl 6(1):44–58
25. Scallen SF, Hancock PA (2001) Implementing adaptive function allocation. Int J Aviat Psychol 11(2):197–221
26. Scerbo MW, Freeman FG, Mikulka PJ, Parasuraman R, Di Nocero F, Prinzel III LJ (2001) The efficacy of psychophysiological measures for implementing adaptive technology. NASA/TP-2001-211018
27. Inagaki T (2003) Adaptive automation: sharing and trading of control. In: Hollnagel E (ed) Handbook of cognitive task design. LEA, Mahwah, pp 147–169
28. Parasuraman R, Sheridan TB, Wickens CD (2000) A model for types and levels of human interaction with automation. IEEE Trans Syst Man Cybern 30(3):286–297
29. FAA (2011) Introduction to TCAS II version 7.1 booklet HQ-111358. Washington, DC
30. Bresley B, Egilsrud J (1997) Enhanced ground proximity warning system. Boeing Airliner, pp 1–13
31. Billings CE (1992) Human-centered aircraft automation: a concept and guidelines, vol 103885, NASA technical memorandum. NASA-Ames Research Center, Moffett Field
32. Cacciabue PC (2004) Guide to applying human factors methods: human error and accident management in safety critical systems. Springer, London
33. Wickens CD, Lee JD, Liu Y, Becker SEG (2004) An introduction to human factors engineering, 2nd edn. Prentics-Hall, Upper Saddle River
34. Orlady HW, Orlady LM (1999) Human factors in multi-crew flight operations. Ashgate, Aldershot
35. Endsley MR, Kiris EO (1995) The out-of-the-loop performance problem and the level of control in automation. Hum Factors 37(2):3181–3194
36. Parasuraman R, Riley V (1997) Humans and automation: use, misuse, disuse, abuse. Hum Factors 39(2):230–253
37. Inagaki T, Stahre J (2004) Human supervision and control in engineering and music: similarities, dissimilarities, and their implications. Proc IEEE 92(4):589–600
38. Hollnagel E, Woods DD (2005) Joint cognitive systems: foundations of cognitive systems engineering. CRC Press, Hoboken

39. Inagaki T (2006) Design of human-machine interactions in light of domain-dependence of human-centered automation. Cognit Technol Work 8(3):161–167
40. Inagaki T, Kunioka T (2002) Possible automation surprises in the low-speed range adaptive cruise control system. In: IASTED international conference on applied modelling and simulation, Cambridge, MA, pp 335–340
41. ITARDA (2003) Anecdotal report on traffic accident investigations and analyses (in Japanese). ITARDA, Tokyo, Japan
42. Scott WB (1999) Automatic GCAS: "you can't fly any lower". Aviat Week Space Technol 150 (5):76–79
43. Kingsley-Jones M, Warnick G (2006) Airbus studies emergency traffic avoidance system to act without pilots. Flight International 22 Mar 2006
44. Kaminski-Morrow D (2009) Airbus A350 could be equipped with automatic emergency descent system. Flight International 15 Aug 2009
45. Inagaki T, Sheridan TB (2012) Authority and responsibility in human-machine systems: probability theoretic validation of machine-initiated trading of authority. Cognit Technol Work 14(1):29–37
46. Inagaki T, Itoh M, Nagai Y (2006) Efficacy and acceptance of driver support under possible mismatches between driver's intent and traffic conditions. In: Proceedings of HFES 50th annual meeting, San Francisco, CA, pp 280–283
47. Inagaki T, Itoh M, Nagai Y (2007a) Driver support functions under resource-limited situations. In: Proceedings of HFES 51st annual meeting, Baltimore, MD, pp 176–180
48. Inagaki T, Itoh M, Nagai Y (2007) Support by warning or by action: which is appropriate under mismatches between driver intent and traffic conditions? IEICE Trans Fundam E90-A (11):264–272
49. Inagaki T (2011) To what extent may assistance systems correct and prevent 'erroneous' behaviour of the driver? In: Cacciabue PC et al (eds) Human modelling in assisted transportation. Springer, Milan, pp 33–41
50. Lee JD, Moray N (1992) Trust, control strategies and allocation of function in human-machine systems. Ergonomics 35(10):1243–1270
51. Mosier K, Skitka LJ, Heers S, Burdick M (1998) Automation bias: decision making and performance in high-tech cockpits. Int J Aviat Psychol 8:47–63
52. Meyer J (2001) Effects of warning validity and proximity on responses to warnings. Hum Factors 43(4):563–572
53. Sheridan TB, Parasuraman R (2005) Human-automation interaction. In: Nickerson RS (ed) Reviews of human factors and ergonomics, vol 1. Human Factors and Ergonomics Society, Santa Monica, pp 89–129
54. Inagaki T (2010) Traffic systems as joint cognitive systems: issues to be solved for realizing human-technology coagency. Cognit Technol Work 12(2):153–162
55. Ladkin PB (2002) ACAS and the south German midair. Technical note RVS-Occ-02-02. http://www.rvs.uni-bielefeld.de/publications/Reports/
56. Learmount D (2002) Questions hang over collision. Flight International, 8
57. Inagaki T, Moray N, Itoh M (1998) Trust self-confidence and authority in human-machine systems. In: Proceedings of IFAC man-machine systems, Kyoto, Japan, pp 431–436
58. Inagaki T (1999) Situation-adaptive autonomy: trading control of authority in human-machine systems. In: Scerbo MW, Mouloua M (eds) Automation technology and human performance: current research and trends. Lawrence Erlbaum Associates, Mahwah, pp 154–159
59. Inagaki T (2000a) Situation-adaptive autonomy for time-critical takeoff decisions. Int J Model Simul 20(2):175–180
60. Inagaki T, Takae Y, Moray N (1999) Automation and human interface for takeoff safety. In: Proceedings of tenth international symposium on aviation psychology, Columbus, OH, pp 402–407
61. 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. In: Proceedings of IEEE SMC conference, The Hague, The Netherlands, pp 3932–3937

62. Hollnagel E (2006) A function-centered approach to joint driver-vehicle system design. Cognit Technol Work 8:169–173
63. Hollnagel E (1999) From function allocation to function congruence. In: Dekker SWA, Hollnagel E (eds) Coping with computers in the cockpit. Ashgate, Brookfield, pp 29–53
64. Inagaki T (1993) Situation-adaptive degree of automation for system safety. In: Proceedings of 2nd IEEE international workshop on robot and human communication, Tokyo, Japan, pp 231–236
65. Inagaki T (2000b) Situation-adaptive autonomy: dynamic trading of authority between human and automation. In: Proceedings of HFES 44th annual meeting, San Diego, CA, pp 3.13–3.16
66. Jordan N (1963) Allocation of functions between man and machines in automated systems. J Applied Psychology 47(3):161–165
67. Hollnagel E (2003) Prolegomenon to cognitive task design. In: Hollnagel E (ed) Handbook of cognitive task design. LEA, Mahwah, pp 3–15