International Journal of ITS Research, Vol. 1, No. 1, December 2003

Adaptive Automation for Comfort and Safety

Toshiyuki Inagaki

Institute of Information Sciences and Electronics
University of Tsukuba
(Tsukuba 305-8573 Japan; inagaki@is.tsukuba.ac.jp)

Function allocation is the design decision to determine which functions are to be performed by humans and which are to be performed by machines for achieving the required system goals. This paper describes the concept of adaptive automation in which the control of functions shifts between humans and machines dynamically, depending on the situation encountered, human workload, and performance. This paper discusses design issues that are central and crucial in implementing sensible adaptive automation for advanced automobiles.

Keywords: *Human-centered automation; Authority; Trust and complacency; Mode awareness; Automation surprises; Human interface; Human-computer interaction; Reliability and safety*

1. Introduction

There are various semi-autonomous systems, 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 systems semi-autonomous, rather than being fully automated? A most obvious reason is that we cannot foresee in the design phase all possible events that may happen during the expected lifetime of the systems. In early days of automation, designers tried to replace human operators by machines for higher efficiency or reliability. However, they realized that human operators have to be within the system to deal with situations that the designers did not anticipate. In other words, operators are requested to complete the system design [2].

For semi-autonomous systems, it is important to determine what humans do and what machines do. Function allocation refers to design decisions that determine which functions are to be performed by humans and which are to be performed by machines. Various strategies for function allocation have already been proposed. Rouse [3] classified traditional function allocation strategies into three types. 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.

Humans may not be very happy with static function allocation. Suppose design decisions are made by using either the leftover or the economic allocation strategy. The strategies do not reflect human characteristics or viewpoints, and treat humans as if they were machine elements. The resulting function allocation may be elusive for operators. Operators may also have to adapt to machines unwillingly. Readers may have seen such *technology-centered automation* in Charlie Chaplin's *Modern Times*. The comparison allocation seems to be nicer for the operators than either the economic or leftover allocations. Even when the operators are allocated only functions in which people surpass machines, the superiority may not hold at all times and on every occasion. For example, operators 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" 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 computer 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 computer. 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* [4-6].

This paper discusses how adaptive automation is useful either in peacetime or in emergency, how it can be implemented, what technologies may be necessary to solve problems that humans often encounter while coping with smart machines.

2. Adaptive automation

2.1. Sharing and trading of control

Function allocation between human and machines can be described more precisely when sharing and trading of control are distinguished.

Sharing of control means that the human and the computer work together simultaneously to achieve a single function [1]. Three types of sharing are distinguishable. The first type is *extension*, in which the computer may help the human so that his/her capability may be extended (e.g., the power steering or the power braking of an automobile), or in which the human extends the computer's capability (e.g., pilots in some types of aircraft may add control force when the maneuver by the autopilot was not perceived satisfactory).

The second type is *relief*, where the computer helps the human so that his/her burden may be reduced. A lane-keeping support system for an automobile is a good example. The system detects lane boundaries on the road, and generates torque to assist the driver's steering action for keeping the host vehicle approximately on the center of the lane.

The third type is *partitioning*, in which a required function is divided into portions so that the human and the computer may deal with mutually complementary parts. A car driver may want to be responsible only for steering by letting the computer control the speed.

Trading of control means that either one of the human or the computer is responsible for a function, and an active agent changes alternately from time to time [1]. Suppose a man, who has been driving his car felt fatigued on his feet, activated the adaptive cruise control (ACC) system. At that time point, he traded the speed control function to the automation. Some time later, he may refresh himself and may want to seize back the speed control task by deactivating the ACC system. In this way, the speed control function can be traded occasionally between the driver and the ACC system.

2.3. Automation Invocation Strategies

In adaptive automation, functions can be reallocated to humans and machines in response to changes in situations or human performance. Three classes may be distinguished for automation invocation strategies that determine reallocation of functions: (a) critical-event strategies, (b) measurement-based strategies, and (c) model-based strategies [4, 6].

Critical-event strategies 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.

Measurement-based strategies dynamically adjust function allocation by evaluating moment-to-moment workload. It is necessary to develop *custom tailored* algorithms if the system is to be compatible with individual operators [5]. Individual differences in human operator capabilities will also influence the response to multiple task demands.

Modeling-based strategies utilize some theoretical models, such as operator performance models, to estimate current and predicted operator state and to infer whether workload is excessive or not. The models are often categorized into three groups: Intent inferencing models, mathematical models, and resource models.

Among the three classes, the critical-event and the measurement-based strategies are useful for real-world applications, including advanced automobiles.

2.4. May automation invocation be automated?

Who is supposed to make decisions concerning when and how function allocation must be altered? The human operator, or machine intelligence? Note here that the automation invocation strategies can be expressed in terms of *production rules*: For instance, a particular critical-event strategy may be represented as: "If critical-event E is detected, then function F must be handed over to the automation, if the function was dealt with by the human at that time point." Once the production rules are given, it is basically possible for the computer to implement adaptive function allocation without any help from the human operator. However, for some reasons, the reality is not so simple.

One apparent reason is *reliability*. Suppose we have developed an automated alert system that can detect critical events. In cases of aviation, the ground proximity warning system (GPWS) is a typical example for such an automated alert system (Figure 1). Though reliability of the GPWS is high, it sometimes produces nuisance or false alerts. If the computer performed an automatic collision avoidance maneuver based on an inappropriate alert, inconvenience may be caused. Presently, the GPWS is not given authority to compel pilots to obey its alerts.

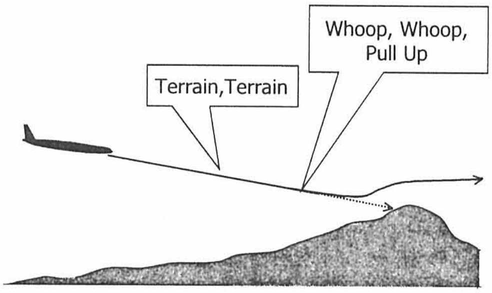

Figure 1. GPWS alert messages

A second reason is the *human-centered automation.* Human-centered automation has been sought for realizing an environment in which humans and machines can work cooperatively in more sound and comfortable manners [7-9]. The idea is popular in various application areas, and thus there are now ten different meanings of human-centered automation [10]. A common claim among those arguments is that, "the human must be at the locus of control," or, "the human must be maintained as the final authority over the automation" [7, 9].

Then, do we assume that the human operator bears the final decision authority at all times and for every occasion? The reality is not so simple, again. Rouse [11] says, "when an aid is most needed, it is likely that humans will have few resources to devote to interacting with the aid." Let us take, as an example, the traffic alert and collision avoidance system (TCAS) for aircraft. When a mid-air collision is anticipated, the TCAS gives pilots a *resolution advisory*, such as "Climb, Climb" (Figure 2).

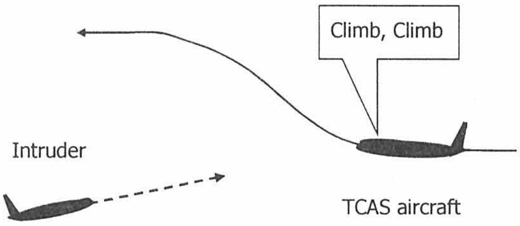

Figure 2. TCAS resolution advisory

However, pilots may disregard a resolution advisory when they are definitely sure that it is wrong. Recall the mid air crash on July 1, 2002, in which two TCAS-equipped aircraft collided over south Germany [12, 13]. When a conflict developed between two TCAS-equipped aircraft, the two TCASs communicate each other to determine which aircraft should climb and which should descend. In the above case, one of the aircraft descended according to the TCAS resolution advisory, and the other aircraft descended, too, even though its TCAS told the pilot to climb. If the TCAS were given a bit more authority over the pilots, the mid air collision might have been avoided.

We have to ask ourselves the following question: Do we claim that, "human must be maintained as the final authority over the automation *at all times and on every occasion*"? Operating environments change with time, and it may not be easy for humans to make a correct decision in a changed environment, especially when available time or information is limited. We may reach a different answer depending on what kind of human we assume as an operator of the system, and what kind of situations the operator may be faced with.

Let us investigate in the next section the authority issue by distinguishing between a professional operator who is a well-trained expert of the system, and a non-professional operator whose main job is something other than to operate the system.

3. Authority in decision and action

3.1. Cases of professional operators

If an operator is a professional expert, it can be assumed that he or she has received thorough training to operate the system.

Example 1. *A controlled flight into terrain (CFIT) accident of a Boeing 757 aircraft occurred in 1995 near Cali, Colombia. The original plan of the southbound night flight to Cali was to fly to the point locating 8 nautical miles south of the airport, and then make a U-turn for the northbound landing on Runway 1. The flight was far behind schedule. When the Cali approach controller offered a straight-in landing to Runway 19, the pilots accepted to make up for lost time. Since their altitude was too high for the new flight path, the pilots extended the speed brakes to expedite the descent. Meanwhile, the pilots became unaware where they were flying, because they supplied an inappropriate command to the flight management computer based on their misunderstanding of a clearance issued by the Cali approach controller. After a couple of unnecessary turns, the aircraft flew into a mountainous area unintentionally. The aircraft was still descending. When the GPWS* *issued a "Pull Up" warning, the pilot responded to it aggressively by pulling up his control column and applying the maximum thrust. However, the pilot failed to stow the speed brakes, and the aircraft crashed into a mountain*; for further detail, see, e.g., [14].

The CFIT accident could have been avoided if there had been an automatic mechanism to retract the speed brake if it had not yet been stowed when the maximum thrust was applied. It is almost impossible to imagine a situation where one would apply the speed brake and maximum thrust at the same time. When automation detects such a *contradiction*, it seems reasonable to allow the automation to adjust the configuration automatically so that the new configuration may fit well to the human's latest intention. The human may not have enough time to do several things, including detecting and recovering their own errors.

3.2. Cases of non-professional operators

Highly intelligent cars are now under development in some countries. These vehicles are equipped with various sensors, such as ones to detect obstacles much earlier than a human driver can.

Example 2. *Suppose a person is driving fast with such a smart car on a dark night. Suppose 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. The driver cannot see the obstacle even when it really exists because it is invisible to him or her at a range when the sensor can detect it. What should the driver do in this situation? Some drivers may slow down immediately to avoid a possible hazard. Some may postpone a hazard avoidance action until they identify the obstacle. Drivers' responses will depend on how great a value they put on safety, and how great is their trust in the automated warning system. For drivers who postpone their decisions, little time may be left for them to avoid the obstacle.*

If the driver keeps going without taking any action to avoid the hazard, how should the automation behave? (Figure 3.) Is not it permissible for the automation to decrease the speed automatically if the driver takes no action for several seconds after the alert was issued? The automatic deceleration may produce time for the driver to avoid a collision with the obstacle. Accepting such a safety related automatic action as being reasonable implies that we give final authority to the automation in this situation.

Suppose an automatic action can never be acceptable. What do we do? Do we train every driver to respect the automated alerts? Do we prepare SOP (Standard Operating Procedures) and request drivers to follow the SOP? It would be unrealistic to train perfectly all non-professional car drivers in the world. It would then be hard to believe that car safety is ensured when the final authority is given only to the driver at all times and on every occasion.

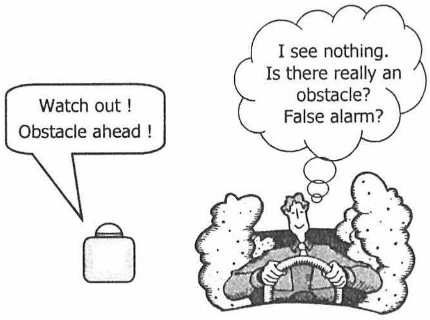

Figure 3. Poorly responding driver

3.3. Situation-adaptive autonomy

The above discussions lead to the argument, "the automation may be given the right to take an automatic action for maintaining system safety, even when a directive may not have been given explicitly by the operator at that time, if the he/she approved beforehand such automatic life-saving actions in emergency.

It is sometimes useful to use a scale for the *level of automation* (LOA) when we discuss the issue of authority. Table 1 gives the scale suggested by Sheridan [1], in which the LOA ranges from no automation to complete automation. It is easy to see that, up to level 5, the human has the ultimate authority over the automation. At level 6 or higher, however, the human may not be maintained as the final authority. For instance, the computer seizes authority at level 6 to execute an automatic action without any directive by the human.

Table 1. Levels of automation [1]

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
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.

Inagaki [15-17] has proposed the concept of *situation-adaptive autonomy* in which final authority may be traded dynamically between humans and automation in a context-specific manner. Based on a probabilistic model, it has been proven that the LOA may be altered dynamically depending on the situation encountered, and LOA at level 6 or higher may be adopted when judged to be necessary to attain system safety. The results imply that the conventional human-centered automation principle is not always right from the viewpoint of system safety.

Example 3. *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 (2000a) proved that the Go/No Go decision should be neither fully automated nor left always to a human. More concretely, (a) the human pilot must be in authority when the aircraft speed is far below V1; (b) the computer must be in authority if the aircraft is almost at V1 and if there is a possibility that the human pilot may hesitate to make decisions when he or she fails to decide whether the engine is faulty or not; and (c) when the aircraft speed is between (a) and (b), the selection of the agent in charge depends on the situation.*

The following is an example of the situation-adaptive autonomy in aviation.

Example 4. *When a collision against the terrain is anticipated, the automatic ground collision avoidance system (Auto-GCAS) gives a "pull-up" warning. If the pilot takes a collision avoidance maneuver aggressively, then the Auto-GCAS does not step in any further (i.e., the LOA stays at level 4). If the pilot does not respond to the warning, the Auto-GCAS takes control back from the pilot and executes an automatic collision avoidance action, in which the LOA goes up to level 6* [18].

The situation-adaptive autonomy example can be found also in automobile.

Example 5. *The forward collision damage mitigating braking system* [19] *for advanced safety vehicles (ASV) firstly gives the driver an alert when a collision against a forward obstacle is anticipated, in which the LOA is at level 4. If the system determines that the collision is imminent, it controls the braking system autonomously to avoid or mitigate the collision damage. The LOA of the system can thus go up to level 7.*

The Auto-GCAS and the forward collision damage mitigating braking system adopt critical-event strategies for automation invocation. If it is possible to define a critical event and if technology is available to detect the event, adaptive automation can be constructed easily with the use of critical-event strategies.

4. Benefits and costs of adaptive automation

4.1. Safety and workload control

Benefits of adaptive automation for automobiles are almost apparent. In peacetime, the adaptive automation can yield comfort to the driver by regulating workload so that it may neither be too high nor too low. The workload management can contribute to human error prevention, maintenance of situation awareness, and improvement of vigilance. In emergency, autonomous safety control action can save drivers when they are late in responding to the situation encountered under extreme time stress and pressure.

4.2. Psychological effects

Adaptive automation might bring costs as well as benefits, especially when the automation is reliable and highly autonomous. Even in conventional automation, it is well recognized that operators are likely to suffer from the *out of the control loop* phenomena, which lead to degradation of manual skill, vigilance decrements, and loss of situation awareness for the required tasks [5, 20-24]. When the automation or the system is perceived as being highly reliable, *complacency* may arise [25, 26], where complacency refers to the self-satisfaction that may result in nonvigilance based on an unjustified assumption of satisfactory system state. On the other hand, operators may feel *distrust* of the automation if they fail to understand what it is doing, why it is doing that, or what it will do next.

If adaptive automation was not carefully designed, it may not be free from similar costs of today's automation, because adaptive automation is much smarter and more sophisticated than conventional automation.

4.3. Conflict between humans and machines

Conflict between intentions of the human and the computer would be one of potential concerns in adaptive automation. Such conflicts have already been observed in conventional automation. The conflicts may be classified into two types. The first type refers to the case in which the human and the computer have "similar" but different intentions. The following is such an example.

Example 6. *An Airbus 320 aircraft crashed in 1991 near Strasbourg, France. The pilots had an intention to make an approach using a flight path angle mode of –3.3 degrees. However the computer, which was the active agent for flying at that time moment, was given a command to create an intention to make an approach by using a vertical speed mode of –3,300 feet per minute. If the pilots had carefully interpreted various clues given in their primary flight displays, they could have noticed that, although the aircraft was descending, the vertical flight path was quite different from the one that they had planned* [7, 27].

The second type refers to the case in which the human and the computer have completely conflicting intentions, which is seen in the following example.

Example 7. *An Airbus 300-600R aircraft crashed at Nagoya in 1994. At some point during the final approach, the pilot gave unintentionally a Go-Around directive to the computer. The computer started its pitch-up maneuver. However, the pilot decided to descend for landing. The pilot knew that the autopilot was in the Go-Around mode, but he did not follow an appropriate procedure to cancel the mode. The goals of the pilot and the computer were thus quite conflicting. The computer was ordered by the pilot to go around, and it tried to achieve the go-around at any cost. To the computer, the pilot's input force to descend was a disturbance that must be cancelled out by applying a stronger control to the stabilizer for ascending. From the viewpoint of the pilot, the aircraft did not descend smoothly, and he applied a stronger control input to the elevator. The aircraft was subject to completely contradictory controls by the two agents with opposite intentions. It finally stalled and crashed* [28].

4.4. Automation surprises

Lee and Moray [29] 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."

For most technological artefacts, the first dimension does not yield serious problems. The technological systems usually satisfy also requirements for the fourth dimension of trust. For instance, it is apparent for what purpose the GPWS has been designed. It is also the case for adaptive automation. Operators would understand the designer's motives to help users either in peacetime or in emergency by changing function allocation suitably to the situation encountered.

Respecting the second and the third dimensions of trust is not very straightforward. Because adaptive automation is designed to change function allocations in a context-dependent manner, its behavior can be obscure. Human's understanding of the automation invocation algorithms may be imperfect, if the algorithm is very "sophisticated" or complicated.

Suppose there are two conditions, A and A*, that differ only slightly. What happens if the operator thought that condition A had been met, whereas it was condition A* that had become actually true and the automation invocation algorithm detected it? The operator would be confused or surprised when he or she saw that the automation did not behave as he or she expected.

The phenomena in which operators are surprised by the behavior of the automation are called *automation surprises* [30, 31]. The surprised operators often ask questions such as, what the automation is doing now, why it did that, or what is it going to do next.

Example 8. *An Airbus 330 aircraft crashed at Toulouse in 1994. The accident occurred in a test flight for investigating performance of the autopilot during an engine-out go-around. The pilot commanded the autopilot on at 6 seconds after takeoff. The goal of the autopilot was to climb to the 2,000 feet altitude that had already been set. The autopilot calculated at which point it had to activate the altitude acquisition transition (ALTSTAR) mode to achieve a smooth level-off. The calculation was done while both engines were operating perfectly and the aircraft was climbing at the vertical speed of 6,000 feet/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 only at 2,000 feet/min. To achieve the already calculated climb rate (6,000 feet/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, and why. Since there was no pitch limit in the ALTSTAR mode, the pitch angle reached 31.6 degrees. At that stage, the captain disconnected the autopilot. It was too late, however, to regain control of the aircraft* [28].

5. Thought experiment

Suppose we are requested to design human-computer interaction for a lane-keeping support system. Our goal is to develop an adaptive automation that can alter, when appropriate, an agent in charge of the lane-keeping task from a human driver to the computer, or vice versa. Suppose our lane-keeping support system can detect lane boundaries on the road and can generate torque to keep the host vehicle approximately on the center of the lane, although the system is basically designed to "assist" driver's steering maneuver.

Assume also that the lane-keeping support system must be designed so that it can return the responsibility of steering task to the driver, when the system determines that the driver is not involved with the steering task appropriately. Driver's inactive steering can suggest that he/she may be complacent, overly reliant on the automation, or may be simply drowsy, in which his/her situation awareness may be poor. Increasing the driver's involvement with the steering task by returning its responsibility to the driver would be useful in improving his/her situation awareness or vigilance. The adaptive automation that we are discussing now trades control based on a measurement-based strategy.

Now, let us imagine a situation in which a man is driving by letting the lane-keeping support system at work. Suppose the computer for the system determines, by monitoring moment-to-moment steering torque, that the driver has not been actively involved with steering task for a while. What should the computer say to the driver in the situation? (Figure 4)

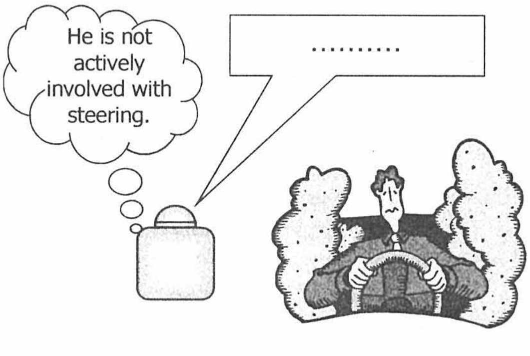

Figure 4. What should the computer say to the poorly involving driver?

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." However, the driver may not respond at all, if he/she disagrees with the diagnosis, or if he/she even 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?" 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." 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." 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 may be that the computer hands over control to the driver *silently*. 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/she 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.

The readers may have already noticed that the discussion given above is related to the LOA. The point is that the LOA that the designer wanted to achieve may not be achieved properly if careful considerations have not been given. In designing human-machine systems, it is important to predict how the design may affect humans and change their behaviors [32]. Designers may have to revise the design based on such considerations.

6. Avoiding costs of adaptive automation

6.1. Human interface

Many incidents and accidents in highly automated aircraft show that even well trained operators can fail to understand the intention or behavior of the automation [14, 33, 34]. Automobile is an application area to which adaptive automation concept is going to be applied more extensively. More precisely, adaptive automation concepts have already been implemented on some smart cars, such as the ASV. Note here again that adaptive automation does aim at semi-autonomous system in which humans and machines cooperate harmoniously, rather than a fully autonomous system that are controlled solely by the computer. In such semi-autonomous systems, human interface must be designed carefully to let operators know what the automation is doing now, why it did that, or what is it going to do next.

It would not be sensible to assume that every car driver has deep understanding on the functions and control logic of the automation on his or her car. Inagaki and Kunioka [35] have investigated driver interactions with a low-speed range ACC system. By distinguishing two types of control strategies for the case in which the ACC system loses sight of a target vehicle to follow, they have conducted experiments with a PC-based driving simulator, where no information was displayed regarding the state of the ACC system. In other words, subjects were requested to judge, decide, and act based solely on their mental models. It has been shown that, even with some training or experience in driving with the simulator, loss of mode awareness and automation surprises did happen. Some phenomena related to over-trust and lack of trust in automation were also observed in the same experiment. The results shown the need for a good human interface as a measure to complement or externalize mental models.

6.2. Intent recognition

Semi-autonomous systems controlled by computers under human supervision are neatly represented by a *human supervisory control* model [1]. The model distinguishes the following four units: (a) human supervisor, (b) human-interactive computer (HIC), (c) task-interactive computers (TIC), and (d) technical process to be controlled (Figure 5). The human supervisor decides what to do and issues commands to the HIC that has capability to communicate with the human supervisor. The HIC interprets directives given by the human supervisor, provides him/her with system state information in an integrated form, and gives decision aids or alert messages when appropriate. Upon receiving a supervisor's directive, the HIC issues necessary commands to at least one TIC. The TIC then performs feedback control by use of its actuators and sensors.

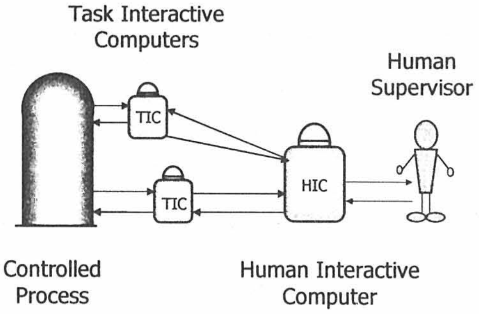

Figure 5. Human supervisory control

We have seen in previous Examples that safety of human-machine systems may be degraded when humans failed to understand the intent of automation, or when humans and automation have different goals. The examples show that, although each TIC may have a local picture, TICs may not have a whole picture of the overall system.

In the crash of Airbus A300-600R at Nagoya, the TIC focused its attention only on accomplishment of pitch-up maneuver that had been ordered by the pilot. The TIC had intelligence, sensors and actuators that were enough for performing the task. They were not sufficient enough, however, to distinguish between a disturbance to be compensated and an intentional control applied by the human pilot. The above accident suggests the following types of intelligence may be needed for each TIC: (a) intelligence to notice and investigate the possibility that some agent other than itself may be applying a control input to the system, (b) intelligence to identify at which point its own control conflicts with that of others, (c) intelligence to evaluate the outcome of the conflict and predict what may happen if the conflict continues to exist, and finally, (d) intelligence to decide what action may be taken to resolve the conflict [36].

In the crash of B757 aircraft near Cali, there was no conflict between pilots and automation until a ground proximity warning was issued. An unacceptable conflict emerged when the pilot pulled up his control column, without stowing the speed brakes. Detection of the conflict may not be difficult for the TIC controlling the speed brakes, if input command sequences to all TICs are available with time stamps. Suppose the TIC controlling the speed brakes noticed that an input command had been sent to one of TICs to advance the thrust, and that the command was still valid. Based on the assumption that no passenger aircraft needs an acrobatic maneuver in which a full thrust and the speed brakes are applied at the same time, it would be possible for the TIC to judge that the pilot does not need the speed brakes any longer.

7. Concluding remarks: Toward sensible adaptive automation

Design aspects for sensible adaptive automation are summarized in Table 2.

The first aspect can be regarded as physical collaboration between humans and machines. Required automation differs, depending on the type of collaboration. For instance, in case of trading, the automation must be designed so that it can replace the human completely. In case of extension or relief, we need a mechanism to add control force on the other agent's force.

The second aspect requires analyses on the following: (a) availability of a clear definition on the critical event, (b) availability of methods to measure indices precisely without placing a heavy burden on the operators, and (c) availability and precision of performance models for a wide variety of operators.

The third aspect deals with mental collaboration between operators and the automation. The aspect is closely related to the principle of human-centered automation. However, as has been argued several times, it is not wise to assume that human operators must be maintained as the final authority at all times and for every occasion.

Table 2. Design aspects of adaptive automation [4]

1. *Sharing control or trading control*
    - (1) Sharing control
        - (i) Extension, (ii) Relief, (iii) Partition
    - (2) Trading control
2. *Automation invocation*
    - (1) Critical-event strategies
    - (2) Measurement-based strategies
    - (3) Model-based strategies
3. *Decision authority*
    - (1) The human is maintained as the final authority at all times and for every occasion
    - (2) Either the human or the automation may have the final authority, depending on the situation

Adaptive function allocation offers more flexible design decisions than static function allocation. This very flexibility, however, may bring operators various inconveniences or undesired results when the adaptive automation is put into use. Before implementing design ideas, designers are requested to analyze possible consequences of design decisions (including design of authority) not only qualitatively but also quantitatively with various methods, such as mathematical modeling, computer simulations, and cognitive experiments [4].

References

[1] Sheridan, T. B., *Telerobotics, Automation, and Human Supervisory Control*. MIT Press, 1992.

[2] Bainbridge, L., "Ironies of automation." *Automatica*, 19, 775-779, 1983.

[3] Rouse, W.B., *Design for Success: A Human Centered Approach to Designing Successful Products and Systems*. New York: Wiley, 1991.

[4] Inagaki, T., "Adaptive automation: Sharing and trading of control." In E. Hollnagel (Ed.) *Handbook of Cognitive Task Design,* 147-169, LEA, 2003.

[5] Parasuraman, R., Bhari, T., Deaton, J.E., Morrison, J.G., & Barnes, M., *Theory and Design of Adaptive Automation in Aviation Systems* (Progress Rep. NAWCADWAR-92033-60). Naval Air Development Center Aircraft Division, 1992.

[6] Scerbo, M. W., "Theoretical perspectives on adaptive automation." In R. Parasuraman & M. Mouloua (Eds.). *Automation and Human Performance*, 37-63, LEA, 1996.

[7] Billings, C. E., *Aviation Automation – The Search for a Human-Centered Approach*. LEA, 1997.

[8] Wickens, C.D., Mavor, A., Parasuraman, R., & McGee, J., *The future of air traffic control: Human operators and automation*. Washington, DC: National Academy Press, 1998.

[9] Woods, D., "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). Moffett Field, CA: NASA-Ames Research Center, 1989.

[10] Sheridan, T.B., *Humans and automation: System design and research issues*. Human Factors and Ergonomics Society & Wiley, 2002.

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

[12] Ladkin, P.B., "ACAS and the south German midair." (http://www.rvs.uni-bielefeld.de/publications/Reports/), 2002.

[13] Learmount, D., "Questions hang over collision." *Flight International*, 8, 2003 July 9-15.

[14] Dornheim, M. "Recovered FMC memory puts new spin on Cali accident." *Aviation Week and Space Technology*, 145(11), 58-61, 1996.

[15] Inagaki, T., "Situation-adaptive degree of automation for system safety." *Proc. 2nd IEEE International Workshop on Robot and Human Communication*, 231-236, 1993.

[16] Inagaki, T., "Situation-adaptive autonomy: Trading control of authority in human-machine systems." *Automation Technology and Human Performance,* 154-159, LEA, 1999.

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

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

[19] *Guidance for commercialization of forward collision damage mitigating braking systems in ASV.* Ministry of Land, Infrastructure and Transport, 2002.

[20] Endsley, M.R., & Kaber, D.B., "The use of level of automation as a means of alleviating out-of-the-loop performance problems: A taxonomy and empirical analysis." In *Proc. of the 13th Congress of the IEA,* 168-170, 1997.

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

[22] Gluckman, J.P., Carmody, M.A., Morrison, J.G., Hitchcock, E.M., & Warm, J.S. (1993). Effects of allocation and partitioning strategies of adaptive automation on task performance and perceived workload in aviation relevant tasks. In *Proc. the 7th International Symposium on Aviation Psychology*, 150-155, The Ohio State University, 1993.

[23] Kaber, D.B. & Riley, J.M., "Adaptive automation of a dynamic control task based on workload assessment through a secondary monitoring task," *Automation Technology and Human Performance,* 129-133, LEA, 1999.

[24] Wiener, E.L., "Cockpit automation." In E.L. Wiener & D.C. Nagel (Eds.). *Human Factors in Aviation*, 433-461, Academic Press, 1988.

[25] Parasuraman, R., Molloy, R., & Singh, I.L., "Performance consequences of automation-induced complacency." *International Journal of Aviation Psychology*. 3(1), 1-23, 1993.

[26] Moray, N. & Inagaki, T., "Attention and complacency." *Theoretical Issues in Ergonomics Science*, 1(4), 354-365, 2000.

[27] Sparaco, P., "Human factors cited in French A320 crash." *Aviation Week and Space Technology*, 140(1), 30-31, 1994.

[28] Dornheim, M., "Dramatic incidents highlight mode problems in cockpits." *Aviation Week and Space Technology*, 142(5), 57-59, 1995.

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

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

[31] Wickens, C.D., "Designing for situation awareness and trust in automation." In *Proc. IFAC Integrated Systems Engineering*, 77-82, Pergamon Press, 1994.

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

[33] Hughes, D., "Incidents reveal mode confusion." *Aviation Week and Space Technology*, 142(5), 56, 1995.

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

[35] Inagaki, T., & Kunioka, T., "Possible automation surprises in the low-speed range adaptive cruise control system." *Proc. IASTED Int'l Conf. Applied Modelling & Simulation*, 335-340, 2002.

[36] Inagaki, T., & Stahre, J., "Human supervision and control in engineering and music: Similarities, dissimilarities, and their implications." *Proceedings of the IEEE*, 2004 (to appear).

Toshiyuki Inagaki received BS, MS, and Doctor's degrees in systems engineering from Kyoto University in 1974, 1976, and 1979, respectively. From 1979 to 1980 he was a Research Associate at the University of Houston, USA. In 1980, he joined the University of Tsukuba, where he has been Professor at the Institute of Information Sciences and Electronics since 1994. From 1990 to 1991 he was at the University of Kassel, Germany, as a Research Fellow of the Alexander von Humboldt Foundation. From 1996 to 2002, he conducted an international and interdisciplinary research project on human-centered automation at the Center for Tsukuba Advanced Research Alliance.

His research interests include adaptive automation, human trust in automated systems, analysis and evaluation of human interface design, reasoning and decision making with imperfect information under time stress.

Dr. Inagaki is a Senior Member of IEEE, and is Chair of IEEE Reliability Society Japan Chapter. He received the best paper awards from the ISCIE in 1994, and from the Human Interface Society in 2001.

Received: 15 October 2003
Accepted: 27 October 2003
Editor: Sadayuki Tsugawa