Chapter 8 of the Handbook of Cognitive Task Design
pp. 147-169, (Erik Hollnagel Ed.) LEA, 2003

# Adaptive Automation: Sharing and Trading of Control

**TOSHIYUKI INAGAKI**

*University of Tsukuba*

## Abstract

Function allocation is the design decision to determine which functions are to be performed by humans and which are to be performed by machines to achieve the required system goals, and it is closely related to the issue of automation. Some of the traditional strategies of function allocation include (a) assigning each function to the most capable agent (either human or machine), (b) allocating to machine every function that can be automated, and (c) finding an allocation scheme that ensures economical efficiency. However, such “who does what” decisions are not always appropriate from human factors viewpoints. This chapter clarifies why “who does what and when” considerations are necessary, and it explains the concept of *adaptive automation* in which the control of functions shifts between humans and machines dynamically, depending on environmental factors, operator workload, and performance. Who decides when the control of function must be shifted? That is one of the most crucial issues in adaptive automation. Letting the computer be in authority may conflict with the principle of *human-centered automation* which claims that the human must be maintained as the final authority over the automation. Qualitative discussions cannot solve the authority problem. This chapter proves the need for quantitative investigations with mathematical models, simulations, and experiments for a better understanding of the authority issue.

Starting with the concept of function allocation, this chapter describes how the concept of adaptive automation was invented. The concept of *levels of automation* is used to explain interactions between humans and machines. *Sharing* and *trading* are distinguished to clarify the types of human-automation collaboration. Algorithms for implementing adaptive automation are categorized into three groups, and comparisons are made among them. Benefits and costs of adaptive automation, in relation to decision authority, trust-related issues, and human-interface design, are discussed with some examples.

## Function Allocation

Suppose we are to design a system with specific missions or goals. We first have to identify functions that are needed to accomplish the goals. We then come 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 are to be performed by machines. Various strategies for function allocation have already been proposed.

*Traditional Strategies for Function Allocation*

Rouse (1991) classified traditional function allocation strategies into three types. The first category is termed *comparison allocation*. 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 (either human or machine). The most famous MABA-MABA (what "men are better at" and what "machines are better at") list may be the one edited by Fitts (1951), see Table 8.1.

Table 8.1 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 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.

Note. After Fitts (1951), Hancock and Scallen (1998), and Price (1985).

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

The third type is named *economic allocation*. The strategies of this type try to find an allocation that ensures economical efficiency. Even when some technology is available to automate a function, if the costs of automating the function are higher than that of hiring a human operator, then 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*: Once a function is allocated to an agent, the agent is responsible for the function at all times.

### *Traditional Strategies Are Not Always Appropriate*

Suppose design decisions are made by using either the leftover or the economic allocation strategies. The strategies do not reflect any human characteristics or viewpoints, and the resulting function allocation may be elusive for operators. Some operator may ask, "Am I supposed 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 nicer for the operators than either the economic or leftover allocations. However, the comparison allocation cannot be free from criticisms. Price (1985) and Sharit (1997) claimed that the list by Fitts is overly generalized and nonquantitative. Sheridan (2002) 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 (1963), "the idea of comparing the human with the machine should be thrown out but the facts about what people do best and what machine do best should be retained," and "the main point of retaining the Fitts list is that people and machine are complementary" (p.59). A *complementary* strategy can be seen in KOMPASS (Grote, Ryser, Wafler, Windischer, & Weik, 2000).

Even though the operators are allocated only functions in which people surpass machines, the superiority may not hold at all times and on every occasion. Operators may get tired after long hours of operations, or they may find it difficult to perform the functions under time pressure. This implies that "who does what" decisions are not sufficient, but "who does what and when" considerations are needed for the success of function allocation.

### *"Who Does What and When" Decisions: An Example*

Aircraft in recent years have become equipped with various automations that can perform important functions to make flights safe, smooth, and efficient. Lateral navigation (LNAV) and vertical navigation (VNAV) are such essential functions. Pilots usually take responsibilities for both LNAV and VNAV during takeoff. In the climb phase, pilots may handle LNAV and let various automations deal with VNAV. During cruise, pilots often hand both LNAV and VNAV over to automation. In descending or landing, pilots may seize back control of either LNAV or VNAV. The two functions are allocated in different ways, depending on the situation.

What happens if aircraft must be designed in a "who does what" manner? No shift of control is allowed, and there can be only four design decisions: first, no automation is needed because the pilots are to handle both LNAV and VNAV at all times during flights; second, design the automation that performs LNAV for any circumstances; third, design the automation that performs VNAV at all times and for every occasion; fourth, design the automation that performs LNAV and VNAV all the time during flights. It is easy to see that none of these are practical.

## Notions Useful for "Who does What and When" Decisions

### *Sharing of Control*

Sharing of control means that the human and the computer work together simultaneously to achieve a single function (Sheridan, 1992). Three types of sharing are distinguishable. The first type is *extension*, in which the computer may help the human so that his or 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., "supervisory override" for some types of aircraft, in which the pilot may add control force when the maneuver by the autopilot was not perceived to be satisfactory).

The second type is *relief*, in which the computer helps the human so that his or her burden may be reduced. A lane-keeping support system for an automobile is a good example. The system detects white lane markers 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 (Kawazoe, Murakami, Sadano, Suda, & Ono, 2001).

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 velocity. Partitioning is a complementary function allocation.

### *Trading of Control*

Trading of control means that either the human or the computer is responsible for a function, and an active agent changes alternately from time to time (Sheridan, 1992). We have already seen an example on the flight deck in which the pilots and the automation trade controls for LNAV and VNAV functions occasionally during flights.

For trading of control to be implemented, it is necessary to decide when the control must be handed over and to which agent. It is also important who makes the decision. These issues are discussed later.

### *Levels of Automation*

Human-computer interactions can be described in terms of the *level of automation* (LOA), originated by Sheridan and Verplank (1978). Table 8.2 gives a simplified version (Sheridan, 1992).

Table 8.2 Scale of levels of automation

| | |
|---|---|
| 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, or |
| 8. | informs him after execution only if he asks, or |
| 9. | informs him after execution if it, the computer, decides to. |
| 10. | The computer decides everything and acts autonomously, ignoring the human. |

(After Sheridan, 1992)

## Functions that may be Automated

Parasuraman, Sheridan, and Wickens (2000) described human-computer interactions by distinguishing the following four classes of functions: (1) Information acquisition, (2) Information analysis, (3) Decision selection, and (4) Action implementation. There can be various design alternatives regarding to what extent each of the four functions may be automated. In other words, an appropriate LOA must be chosen for each function. The automated forms of functions (1)-(4) are called, respectively, acquisition automation, analysis automation, decision automation, and action automation.

### *Acquisition Automation*

When the LOA is set at the lowest, human must him/herself collect every piece of information at all instances. An example of the automated system information acquisition may be radar for automobiles or aircraft, or sonar for ships. Sometimes these systems simply collect information and display it on the screen. When the computer involves more, certain types of acquired information may be highlighted to attract a human's attention. Filtering is another important capability for acquisition automation. It is well recognized that transmitting every piece of information to the operator may lead to undesirable events. For instance, as a result of lessons learned following the accident at the Three Mile Island nuclear power plant, some alarms may better be suppressed in certain circumstances. This is also the case in a commercial aircraft. Suppose an engine catches on fire during takeoff. Even if the sensors detected the fire successfully, the acquired information may be filtered. The fire bell will not ring until the aircraft climbs to a certain altitude (e.g., 400 feet), or until some amount of time (e.g., 25 seconds) elapses after V1 (the takeoff decision speed). Until then, master warning lights are inhibited.

### *Analysis Automation*

If the LOA is set at some moderate level, the computer may be able to give humans some information by processing available raw data. One example of such information is the *prediction* of the future state of a system. There are various examples of this kind. For instance, in central control rooms of recent nuclear reactors, large screens display trend graphs for various parameters with their predicted values. When using a notebook computer, we can learn a predicted remaining life of the battery if we put a mouse cursor on the battery icon. In the cockpit of an aircraft at a level flight, a navigation display may indicate an arc that shows at which point the aircraft is to start a descent (or an ascent). If the pilot changes the flight plan, the computer replaces the old arc with a new one.

Another example of the analysis automation is a system for integrating multidimensional information into an easily understandable form. The enhanced ground proximity warning system (EGPWS) for aircraft is such an example. The EGPWS was designed to complement the conventional GPWS functionality with the addition of look-ahead terrain alerting capabilities. The EGPWS has worldwide airport and terrain databases, and they are used in conjunction with aircraft position, barometric altitude, and flight path information to determine potential terrain conflict. The terrain may be shown on the navigation display in patterns of red, amber, and green, where the colors indicate the height of the terrain relative to the current aircraft altitude (Bresley & Egilsrud, 1997).

### *Decision Automation*

As researchers in naturalistic decision making say, it is often useful to distinguish *situation-diagnostic decisions* and *action selection decisions* (Klein, Orasanu, Calderwood, & Zsambok, 1993; Zsambok & Klein, 1997). For a situation-diagnostic decision the operator needs to identify "what is going on," or to select the most appropriate hypothesis among a set of diagnostic hypotheses. Action selection means deciding the most appropriate action among a set of action alternatives. Some expert systems are equipped with capabilities to automate situation-diagnostic decisions. When an inference has to be made with imprecise information, the expert systems may give humans a set of plausible diagnostic hypotheses with degree of belief information. The LOA of the expert systems is positioned at levels 2 or 3. If, in contrast, the expert systems show humans only a single diagnostic hypothesis with the largest degree of belief among all, the LOA is set at level 4.

The traffic alert and collision avoidance system (TCAS) is an example that can make action-selection decisions automatically. When a mid air collision is anticipated and no resolution maneuver is taken, the TCAS gives pilots a *resolution advisory*, such as "climb, climb, climb." Pilots are supposed to initiate the suggested maneuver within 5 seconds. It is known, however, that the TCAS can produce unnecessary resolution advisories, though such cases do not happen frequently. Pilots may disregard resolution advisories when they are definitely sure that the advisories are wrong. In this sense, LOA of the TCAS resolution advisory is positioned at level 4.

### *Action Automation*

There are many examples of automation for action implementation. A photocopy machine, described in Parasuraman et al. (2000), is a good example for illustrating that various LOAs can be chosen in a single machine. Suppose someone needs to quickly make copies of 10 pages for five people. He or she must decide which mode to use: automatic sorting without automatic stapling, automatic sorting with automatic stapling, or manual mode to make five copies of each sheet. In the last case, he or she must sort and staple sheets manually. The time required for giving necessary directives to the machine through a touch sensitive panel differs, as does the time needed to finish the task. Once a mode has been chosen, operation starts at one of three different levels of automation.

In aviation, the LOA of action is not set high. From the viewpoint of action automation, the LOA of the TCAS is positioned at level 4, because the TCAS itself has no mechanical subordinate to initiate a collision avoidance maneuver. The GPWS does not have capability for such a maneuver, either. It may be worth considering whether a high LOA should never be allowed for automation to implement an action. Take as an example the crash of a Boeing 757 aircraft that occurred near Cali, Colombia, in 1995. The pilots performed a terrain avoidance maneuver immediately upon a GPWS alert. However, they failed to stow the speed brake that they had extended some time before under their previous intention to descend (Dornheim, 1996). The crash 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 pilot applied the maximum thrust. It is hard to imagine a situation where one would apply the speed brake and the maximum thrust simultaneously. When automation detects such a contradiction, it may be reasonable to allow the automation to adjust the configuration automatically (i.e., to stow the speed brake) so that the new configuration may fit well to the pilot's latest intention.

## Adaptive Automation

Suppose a human and a computer are requested to perform assigned functions for some period of time. The operating environment may change as time passes by, or performance of the human may degrade gradually as a result of psychological or physiological reasons. If the total performance or safety of the system is to be maintained strictly, it may be wise to reallocate functions between the human and the computer because the situation has deviated from the original one. A scheme that modifies function allocation dynamically and flexibly depending on situations is called an *adaptive function allocation.* 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 adaptive function allocation is dynamic in nature. The automation that operates under an adaptive function allocation is called *adaptive automation*. The term *adaptive aiding* is used in some literature such as that by Rouse (1988). In this chapter we treat the terms adaptive aiding and adaptive automation as synonyms.

### *Are Adaptive Function Allocation and Dynamic Function Allocation Equivalent?*

It depends on the definition. To give contrast to static function allocation, *dynamic function allocation* may be defined as a scheme that may alter function allocation occasionally in time during system operation. Then, the dynamic function allocation is not always the same as the adaptive function allocation. Consider the following case: The pilots were flying northward to the destination airport. They had already finished supplying the computer with necessary data to make an automatic landing on Runway 01. Suddenly, the air traffic controller called the pilots to tell them that, because of extremely strong south winds, they had to use Runway 19. The pilots were not very pleased with the instruction for the following reasons: (a) they have to pass over the airport to make an approach from the north, which causes at least 10 minutes delay in arrival time; (b) Runway 19 is not equipped with the navigation facility for an automatic landing; and (c) the geography north of the airport is mountainous. It is highly stressful to make a manual approach over mountainous lands under time pressure. However, because there was no alternative, the pilots had to disengage the autopilot to fly manually, and the flight path management function shifted from the automation to the human pilots. The function allocation in this case is dynamic, but not adaptive.

The point is that the adaptive function allocation assumes criteria to decide whether function must be reallocated for better or safer performance or adjustment of human workload. It would be almost useless to implement dynamic function allocation that is not adaptive. In fact, researchers use dynamic function allocation to mean almost the same as adaptive function allocation, by assuming some automation invocation criteria implicitly or explicitly.

## *History of Adaptive Automation*

The notion of adaptive allocation can be traced to 1970s: see Rouse (1976, 1977). Later, Rouse (1988) stated, "The concept of adaptive aiding … emerged in 1974 in the course of an Air Force Systems Command-sponsored project at the University of Illinois that was concerned with applications of artificial intelligence (AI) to cockpit automation" (p.432). The investigators of the project were initially concerned with "getting the technology to work, rather than with how pilots were going to interact with this system" (Rouse, 1994, p.28). During the research project, the investigators found situations in which the pilot and computer chose reasonable but mutually conflicting courses of action. "The desire to avoid conflicting intelligence and create cooperative intelligence quickly lead to questions of function allocation as well as human-computer interaction" (Rouse, 1994, p.28). At that stage, they found inadequacies, as we did in the previous section, in making design decisions on function allocation based on Fitts list. Rouse (1994) says, "Frustration with the MABA-MABA approach led to a very simple insight. Why should function, tasks, etc. be strictly allocated to only one performer? Aren't there many situations whether either human or computer could perform a task acceptably? This insight led to identification of the distinction between static and dynamic allocation of functions and tasks. Once it became apparent that dynamic invocation of automation might have advantages, it was a small step to the realization that the nature of computer assistance could also be varied with the situation" (Rouse, 1994, p.29).

The adaptive aiding concept was further investigated in the Pilot's Associate program, a joint effort of the Defense Advanced Research Project Agency and the U.S. Air Force, managed by the U.S. Air Force's Wright Laboratory. The Pilot's Associate consists of cooperative knowledge-based subsystems with capabilities to (a) determine the state of the outside world and the aircraft systems, (b) react to the dynamic environment by responding to immediate threats and their effects on the preassigned mission plan, and (c) provide the information the pilot wants, when it is needed (Banks & Lizza, 1991; Hammer & Small, 1995). The program uncovered many gaps in technology and showed "the design of adaptive aids has to be based on a thorough understanding of the other activities and functions in the cockpit" (Rouse, 1994, p.30).

## Automation Invocation Strategies

In adaptive automation, functions can be shared or traded between humans and machines in response to changes in situations or human performance. How can such sharing or trading capability be implemented? There are three classes of automation invocation strategies: first, critical-event strategies; second, measurement-based strategies; and third, modeling-based strategies.

### *Critical-Event Strategies*

Automation invocation strategies of this class change function allocations when specific events (called critical events) occur in the human-machine system. It is assumed that human workload may become unacceptably high when the critical events occur. Allocation of functions would not be altered if the critical events did not occur during the system operation. In this sense, function allocation with a critical-event strategy is adaptive.

Three types of logic are distinguished (Barnes & Grossman, 1985; Parasuraman, Bhari, Deaton, Morrison, & Barnes, 1992). The first is emergency logic, in which automation would be invoked without human initiation or intervention. The second is executive logic, in which the sub-processes leading up to the decision to activate the process are automatically invoked, with the final decision requiring the human's approval. The third is automated display logic, in which all noncritical display findings are automated to prepare for a particular event, so that the human can concentrate on the most important tasks.

LOAs differ among the three types of logic. The LOA for the emergency logic is positioned at level 7 or higher, which implies that humans may not be maintained as the final authority. In the executive logic, the LOA is positioned at level 5. The automated display logic assumes sharing (or partitioning) of tasks. The computer distinguishes "noncritical" portions of the tasks from the "the most important" ones, and it allocates the former to machines so that the workload of operators may be reduced or maintained within reasonable levels. The LOA of the automated display logic is set at level 7 or higher, because it is the computer that judges whether a task is noncritical or the most important, and operators are not usually involved in the judgment. Applying a high level of automation, such as level 7 or above, can be beneficial for reducing workload or for buying time. However, it may bring some costs, such as degradation of situation awareness or automation-induced surprises. The issue is discussed in a later section.

Adaptive automation with critical-event strategies may have several operational modes with different LOAs. For instance, the Aegis system has a small rule base that determines how the Aegis system will operate in a combat environment. The following modes are available (Parasuraman et al., 1992): (a) manual, in which the system is fully controlled by the operator; (b) automatic special, in which a weapon-delivery process is automatic, but the fire button has to be pressed by the operator; and (c) fully automated, in which the ship's defensive actions are automatically implemented without operator intervention, because of the need for a short reaction time within which the operator may not complete the required actions.

### *Measurement-Based Strategies*

Automation invocation logic of this class emerged at an early stage in the adaptive automation research. Rouse (1977) proposed the dynamic allocation of functions between operators and machines so that the moment-to-moment workload of the operators could be regulated around some optimal level. Workload levels of operators in complex systems fluctuate from moment to moment and at different mission phases. The operators may be able to achieve very high performance levels but only at the cost of high mental workload by neglecting "less critical" functions. If the situation that requires a high level of workload lasts too long, performance degradation may result. Performance may also deteriorate when other minor tasks are added. These observations give a rationale to adjust function allocation dynamically by evaluating moment-to-moment workload.

However, that does not mean that a single algorithm can be effective for all individuals. In fact, different operators will use different strategies to cope with the demands of multiple tasks under time pressure. It is thus necessary to develop *custom-tailored* adaptive automation algorithms if the system is to be compatible with and complement the strengths and weaknesses of individual operators (Parasuraman et al., 1992). Moreover, individual differences in human operator capabilities will influence the response to multiple task demands: Some operators may have sufficient resources left to cope with other tasks, whereas other operators may be operating at peak workload. This means that an algorithm developed for an "average" operator will not be suitable to either class of operators. For an adaptive system to be effective in maintaining mental workload at an appropriate level in dynamic real-time environments, it must be equipped with a workload measurement technology that is capable of (a) detecting changes in workload levels and (b) identifying what component of mental workload is under stressed or over stressed.

Adaptive automation works as follows under a measurement-based measurement strategy (Hancock & Chignell, 1988): First the task is defined and structured and subtasks are allocated to either an automated subsystem or to the operator. Next, the operator's effort is compared with the task difficulty so as to assign a criterion for adaptivity. The criterion can be expressed as a measure of mental workload, a measure of primary task performance, or a combination of both. Once the criterion is defined, the adaptive system trades task components in order to improve future measurement of the criterion. In the workload-based measurement method, adaptivity can be achieved through three main procedures: by adjusting allocation of subtasks between operators and automation; by adjusting the structure of the task; and by refining the task.

Psychophysiological measures, such as pupillary dilatation and heart rate, may be used for adjusting function allocation. The psychophysiological measures may be recorded continuously and thus be useful, unlike most behavioral measures, in measuring mental activities of human operators placed in supervisory roles that require few overt responses. In addition, the psychophysiological measures may provide more information when coupled with behavioral measures than behavioral measures would alone. For example, changes in reaction time may reflect contributions of both central processing and response-related processing to workload. Refer to Scerbo et al. (2001) for the most up-to-date guide for using psychophysiological measures in implementing adaptive automation.

### *Modeling-Based Strategies*

Operator performance models can be used 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, optimal (or, mathematical) models, and resource models.

Intent inferencing models work as follows (Rouse, Geddes, & Curry, 1987-1988): Operator actions are decoded and compared with the set of scripts. If at least one script matches, the actions are resolved. If no match is found, the unresolved actions are analysed to identify plans. If one or more plans are found that are consistent with known goals, the actions are resolved and the scripts associated with these plans (if any) are activated. If no match is found, the unresolved actions are put into the error monitor.

Optimal models include those based on queuing theory (Walden & Rouse, 1978), pattern recognition (Revesman & Greenstein, 1986), and regression (Morris, Rouse, & Ward, 1986). For example, Walden and Rouse

(1978) investigated multitask performance of a pilot, where time sharing is required between monitoring, control, and other tasks. They modeled the monitoring task as a queuing system with a “single server” and subsystem events called “customers” and with the control task incorporated as a special queue. Once the customers arrive at the control task queue they can control the service of a subsystem event. From what proceeds, a customer in this case can be defined as a “significant amount of display error.” Therefore, when a major error is displayed, the subsystem service is preempted and a control action is taken to eliminate the error.

Resource models, especially the multiple resource theory (Wickens, 1984), try to describe how performance interference occurs in information processing. Suppose an operator is trying to perform two different tasks. If the two tasks require different resources, say verbal and spatial codes, then the operators will have no difficulty performing them efficiently. However, if the two tasks require the same resources, then some conflict can occur and performance of the tasks may suffer significantly. The multiple resource theory may be used to evaluate efficacy of function allocation, or to assess the impact of possible competition that may be caused by tasks requiring the same resource. The multiple resource theory is an important tool for human-machine system design, and it has already been incorporated in discrete-event simulation software, WinCrew, to evaluate function allocation strategies by quantifying the moment-to-moment workload values (Archer & Lockett, 1997).

### *Advantages and Limitations of Automation Invocation Strategies*

Among three types of strategies, the critical-event strategies may be the most straightforward to implement, if critical events are defined properly. No investigations are needed regarding how human cognition or behavior could be modeled, what parameters must be measured to infer the human state, and how. Only we have to do is develop techniques to detect occurrence of the critical event by using information available in the human-machine system. A possible limitation of the critical-event strategies is that they may reflect human workload only partially or implicitly. Subjective workload under the critical events may differ significantly among operators.

From a viewpoint of adapting to an individual who is facing with a dynamically changing environment, measurement-based logic may be the most appropriate. It can change function allocation by explicitly reflecting the mental status of an operator at a specific circumstance. There is no need to predict in advance how the mental status of the operator may change. However, there are a few limitations. First, not all operators may welcome situations in which they are monitored by sensing devices all the time. Second, sensing devices are sometimes expensive or too sensitive to local fluctuations in operator workload or physiological states. Third, performance measurement occurs "after the fact" (Scerbo et al., 2001), that is, after a point in time when adaptation may be needed.

If good performance models are available, it may be possible to extract "leading indicators." Leading indicators refer to precursors that, when observed, imply the occurrence of some subsequent event. For instance, Kaber and Riley (1999) demonstrated the benefits of adaptive aiding on a primary task (a dynamic cognitive monitoring and control task) by taking, as a leading indicator, degradation of the secondary task (an automation-monitored task). It is not always easy, however, to develop a good performance model that represents the reality perfectly.

## Decision Authority

Who is supposed to make decisions concerning when and how function allocation must be altered? The human operator, or machine intelligence? Let us 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 follows: "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." In case of a measurement-based strategy, we have the following: "If human workload is estimated to be lower than a specified value, then function *F* must be traded from the automation to the human." Once the production rules are given, it is basically possible for machine intelligence (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. It is unrealistic to assume that the computer never fails. The failure may be caused by hardware malfunction, software errors, or inappropriate data. If the computer is nevertheless given the authority to make an automation invocation decision, human operators may have to monitor the computer carefully all the time, which produces burden on the operators in addition to their original tasks.

A second reason is related to the principle of human-centered automation which claims that a human operator must be maintained as the final authority and that only he or she may exercise decisions concerning how function allocation must be changed and when (Billings, 1997; Woods, 1989). The principle is reasonable. However, is it always the best for the human operator to bear the final decision authority at all times and for every occasion? Rouse (1988) says, "when an aid is most needed, it is likely that humans will have few resources to devote to interacting with the aid. Put simply, if a person had the resources to instruct and monitor an aid, he/she would probably be able to perform the task with little or no aiding" (p.431). There may be cases in which it is rational that "variations in levels of aiding and modes of interaction will have to be initiated by the aid rather than the human whose excess task demands have created a situation requiring aiding" (Rouse, 1988; p.432).

### *Appropriate LOAs*

The decision authority issue is related to the selection of appropriate LOAs. When the LOA is positioned at level 5 or lower, the human operator is maintained as the final authority (see Table 8.2). The human-centered automation principle is violated when the LOA may be positioned at level 6 or higher. A committee of the U.S. National Research Council discussed appropriate LOAs for new civil air traffic control systems (Parasuraman et al. 2000; Wickens et al., 1998). Sheridan (2002) said, "After much debate, the committee decided that acquisition and analysis could and should be highly automated – in fact, they already are (radar, weather, schedule information, etc.) – but that decision-making, except for certain decision aids now under development, should be done by human air traffic controllers. Implementation is in the hands of the pilots, which in turn is largely turned over to autopilots and the other parts of the Flight Management System" (p.63). The recommended LOA for each function is depicted in Fig. 8.1.

There are some systems in which the computer may initiate invocation of automation. One such example is the automatic ground collision avoidance system for combat aircraft (Scott, 1999). When a collision against the terrain is anticipated, the computer gives a "pull-up" warning. If the pilot takes a collision avoidance maneuver aggressively, then the computer does not step in any further. If the pilot does not respond to the warning, the computer takes control back from the human pilot and executes an automatic collision avoidance action. The LOA is positioned at level 6.

A clear-cut answer is hard to get for the decision authority issue. It is clear that the issue cannot be solved by qualitative discussions alone. Quantitative methods are needed for more precise understanding. The following approaches are described with some examples: (a) experiments, (b) computer simulations, and (c) mathematical modeling.

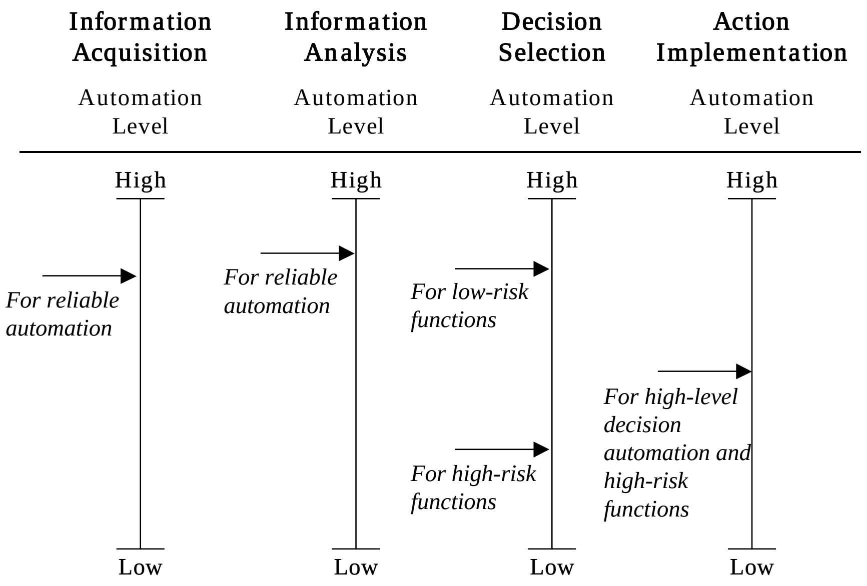

Fig. 8.1 Recommended LOAs for future air traffic control systems (From Parasuraman et al., 2000. © *2000 IEEE*).

*Laboratory Experiments*

Laboratory experiments are usually designed under settings with multiple tasks, such as resource management, system monitoring, and compensatory manual tracking. Comparisons are made among various strategies for function allocation and automation invocation. Some results suggest the efficacy of the human-initiated invocation strategies, and others suggest the need for computer-initiated invocation.

Hancock, Arthur, and Caird (1991) compared the following conditions of a multitask environment: First, the subjects must perform all tasks manually: Second, the tracking task is performed by the automation: Third, the subjects can decide whether to invoke automation for the tracking task. The investigators found that subjects' performances in resource management task were more efficient when automation invocation was initiated by humans. Hilburn, Molloy, Wong, and Parasuraman (1993) compared executive and emergency logic in the critical-event strategies, and they found a slight automation cost under the computer-initiated invocation.

Harris, Goernert, Hancock, and Arthur (1994) compared human-initiated and computer-initiated invocation strategies, and they found that human-initiated invocation of automation might be less beneficial than computer-initiated invocation when changes in the workload could be abrupt or unexpected for the human operator. Harris, Hancock, and Arthur (1993) also found that, when subjects became fatigued under a multiple-task environment, they were less likely to engage automation even when it was supposed to be used, which means that the benefits of automation may not be fully appreciated if human-initiated invocation of automation is adopted.

### *Computer Simulations*

Furukawa, Niwa, and Inagaki (2001) investigated an optimal LOA for an emergency operating procedure for steam generator tube rupture (SGTR) failures in a pressurized water reactor. The SGTR failures are typical events to be assessed for safety of nuclear power plants. The emergency operating procedure for the SGTR failures has seven steps, each of which consists of several substeps. Among them there is the substep to shut the coolant flow at a particular pipe, which requires a sequence of tasks, such as “send a command to close a valve,” “manipulate the valve,” “examine the state of the valve,” and so on. The tasks are categorized into four classes, that is, information acquisition, information integration, decision selection, and control implementation, in a similar manner as shown by Parasuraman et al. (2000). Some strategies of human-automation collaboration may be possible to perform the sequence of tasks. Furukawa et al. (2001) analyzed LOAs positioned at levels 5, 6, and 7. One of the cognitive task networks developed is shown in Fig. 8.2, in which each rectangular node represents a task that should be performed either by the automation (denoted as “A-task name”) or the human operator (“H-task name”), and in which the LOA is positioned at level 5. Each diamond-shaped node is called a decision diamond, which can distinguish types of decision-making logic, such as probabilistic, tactical, and multiple decisions.

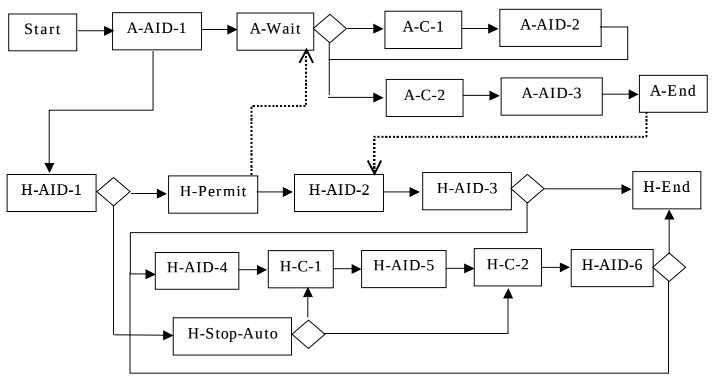

Fig. 8.2 An example of the cognitive task network

The cognitive task network was implemented with WinCrew, which is software for discrete-event simulation. WinCrew quantifies the moment-to-moment workload on the basis of the multiple resource theory (Wickens, 1984). Monte Carlo simulations were performed under various conditions of reliability for human operators and automations. It was found that the LOA positioned at level 6 was best in the sense that it could minimize the expected time for completion of the task sequence. However, the LOA positioned at level 6 was not optimal from the viewpoint of cognitive workload. The cognitive workload might be quantified in various ways, such as by the peak value, the average value, and the time-integral value over the whole period of system operation, and the time length during which the workload exceeded a specified threshold value. It was level 7 that gave the best result from the viewpoint of cognitive workload.

Another interesting computer simulation can be found in Archer, Lewis, and Lockett (1996), in which WinCrew was used to model and evaluate function allocation strategies of the bridge activities on a Navy Guided Missile Destroyer. Some function allocation strategies were defined for a crew of three and the automation, and workload was evaluated for each strategy under several scenarios to investigate the feasibility of reduction of crew size from nine (conventional size) to three.

### *Mathematical Modeling*

Inagaki (1997, 1999, 2000) made a mathematical analysis on the decision authority issue in a rejected takeoff problem of an aircraft. 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, and then make an emergency landing. 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. The V1 speed is a variable depending on weight, aircraft lift capability under various environmental conditions, length of runway used in achieving velocity, position on the runway at the velocity point, whether the aircraft can safely lift off with one engine failed, and so on.

The aforementioned rule for a Go/No Go decision is simple. However, that does not mean at all that the Go/No Go decision is easy to make, because only 1 or 2 seconds are available for the pilots to complete the following tasks: (a) recognizing an engine failure alert, (b) analyzing the situation, (c) deciding if the circumstance warrants rejection of the takeoff, and (d) initiating the first stopping action (i.e., retarding the throttles of all engines) to abort the takeoff. A probabilistic model was given on the basis of the following assumptions:

- An alert is given to the human pilot when a sensor detects an engine failure. However, the sensor can give a false alert.
- The pilot's understanding of a given situation may not be correct. In reality, an engine failure is not a single factor for rejecting the takeoff. Various events such as an engine fire with or without loss of thrust, a blown tire, or a bird strike may happen during takeoff roles. Some of these events may give symptoms similar to those of an engine failure. It is not easy to identify the situation correctly when available information and time are limited. There may be some cases in which the pilot hesitates to say either that the alert is correct or that the alert is false.
- Incorrect or late decisions cause costs that vary depending on the situation.

By evaluating the conditional expected loss, provided an engine failure alert has been set off, Inagaki (1997, 1999, 2000) proved that the Go/No Go decision should be neither fully automated nor left always to a human;

that is, the decision authority of automation invocation must be traded dynamically between human and computer. 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 an agent in charge depends on the situation.

It was also proven (Inagaki, 1997, 1999, 2000) that, for a human pilot to be given decision authority at all times, human-interface design must be changed so that the human can be supported directly in Go/No Go decision making. The suggested human-interface design was to give either a "Go" or an "Abort" message on the display, while an "Engine failure" message appears on the display in the glass-cockpit aircraft. When receiving the "Engine failure" alert, pilots have to interpret the alert: "Engine failure" means "No Go" if it was before V1, but implies "Go" after V1. It is in the interpretation task that human pilots may sometimes make errors. Inagaki, Takae, and Moray (1999) found, by experiments, that improvements in interface design alone were insufficient to attain decision-making accuracy at levels that could be obtained with dynamic trading of decision authority between humans and automation.

## Benefits and Costs of Adaptive Automation

### *Dynamic Alteration of LOAs*

One major motivation for introducing adaptive automation is to regulate operator workload, where an operator "can control a process during periods of moderate workload, and hand off control of particular tasks when workload either rises above, or falls below, some optimal level" (Hilburn et al., 1993; p.161). Another major benefit of adaptive automation lies in its ability to keep the operator in the control loop, which is done by altering the LOA. These characteristics contrast with static function allocation. When the LOA for a function is always positioned at high levels, the operator is 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 function (e.g., Endsley & Kaber, 1997; Endsley & Kiris, 1995; Gluckman, Carmody, Morrison, Hitchcock, & Warm, 1993; Kaber, et al., 1999; Parasuraman et al., 1992; Wiener, 1988). When the automation or the system is perceived as being "highly reliable," automation-induced "complacency" may arise (Parasuraman, Molloy, & Singh, 1993), where complacency refers to the self-satisfaction that may result in nonvigilance based on an unjustified assumption of satisfactory system state. Occasional alteration of the LOA may be useful to avoid the out of control loop phenomena.

What happens if the LOA is altered frequently? If the algorithm were highly sensitive, the LOA would be changed by even a small perturbation in the input value to the algorithm. In extreme cases in which only manual control and full automatic control are available, frequent cycling between automated and manual control may occur, which can lead to performance degradation. The short cycling is a possible byproduct of adaptivity in function allocation. Some researchers investigated the effects of short cycling on task performance by use of laboratory experiments under multitask environments. Parasuraman, Bhari, Molloy, and Singh (1991) demonstrated both benefits and costs of short-cycle automation on the manual performance of tasks and on the monitoring for automation failure. Glenn et al. (1994) investigated the effects on flight management task performance to show no automation deficits, and found automation benefits for reaction time in the tactical assessment task. Scallen, Hancock, and Duley (1995) analyzed the situations in which tracking task cycled between manual and automated control at fixed intervals of 15, 30, or 60 seconds. The investigators found that excessively short cycling of automation was disruptive to performance.

### *Sharing of Intentions*

As in the case of conventional automation, possible failure to share intentions between the human and the computer is one of the major concerns in adaptive automation. Such failures can be classified into two types: The first type refers to the case in which the human and the computer have "similar" but different intentions. An example can be seen in the crash of an Airbus 320 aircraft 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 (Billings, 1997; Sparaco, 1994). 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.

The second type of failure in sharing intentions refers to the case in which the human and the computer have completely conflicting intentions. An example can be seen in the crash of an Airbus 300-600R aircraft at Nagoya, Japan, in 1994 (Billings, 1997). At some point during the final approach, the pilot gave a Go-Around directive to the computer *unintentionally*. The computer started its maneuver for going around. 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. Thus the intentions of the pilot and the computer were quite opposite. The computer was ordered by the human to go around, and it tried to achieve the go-around at any cost. To the computer, the human's input force to descend was a *disturbance* that must be cancelled out by applying a stronger control input to ascend. From the viewpoint of the pilot, the aircraft did not descend smoothly and thus he applied a stronger control input. Thus the aircraft was subject to completely contradictory controls by two agents with opposite intentions.

### *Trust and Automation Surprises*

Lee and Moray (1992) distinguished between four dimensions of trust: (a) *foundation*, which represents the "fundamental assumption of natural and social order that makes the other levels of trust possible," (b) *performance*, which rests on the "expectation of consistent, stable, and desirable performance or behavior," (c) *process*, which depends on "an understanding of the underlying qualities or characteristics that govern behavior," and (d) *purpose*, which rests on the "underlying motives or intents" (p.1246).

For most technological artefacts (gadgets, devices, and machines, complex processes), the first dimension shall not cause serious problems. The technological systems usually satisfy requirements for the fourth dimension of trust. For instance, it is easy to answer the question, "For what purpose did the EGPWS have to be designed?" It would also be the case for adaptive automation. That is, human operators would accept the designer's motives or intents to design a technological system that can help users by regulating operator workload at some optimal level.

Respecting the second and the third dimensions of trust may not be straightforward. Because adaptive automation is designed to behave in a dynamic and context-dependent manner, its behavior may be obscure. Human's understanding of the automation invocation algorithms may be imperfect if the algorithm is "sophisticated" or complicated. Suppose there are two conditions, A and A*, that differ only slightly. What happens if the human operator thought that condition A had been met, whereas it was condition A* that had become true and the automation invocation algorithm detected it? The operator would be 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* (Hughes, 1995b; Sarter, Woods, & Billings, 1997; Wickens, 1994). 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 (Wiener, 1989). When the human could not be sure of fulfillment of the second and third dimensions of trust, he or she would fail to establish proper trust in the adaptive automation. Human's distrust or mistrust in automation may cause inappropriate use of automation, as has been pointed out by Parasuraman and Riley (1997).

## Avoiding Automation Surprises

How should we design a system so that it may not cause automation surprises? That is the key to success of adaptive automation. One possible way to avoid automation surprises may be to design a mechanism through which operators and the automation (the computer) may communicate. Two types of communication are distinguished (Greenstein & Revesman, 1986; Revesman & Greenstein, 1986). The first is dialogue-based communication, in which the operator provides the computer with information regarding his or her action plans. The second is model-based communication, in which the computer predicts actions of the operator based on his or her performance model, and selects its own actions so as to minimize some measure of overall system cost. Dialogue-based communication was found to be advantageous in attaining precise understanding and a high level of situation awareness. However, dialogue may increase human workload (Revesman & Greenstein, 1986). Model-based communication may not increase human workload. However, it may be a new cause of automation surprises, if the assumed human performance model did not match the operator perfectly, or if it failed to cope with time-dependent characteristics of operator performance. Designing good human interface is the most basic requirement for avoiding automation surprises.

### *Human Interface Design*

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. Many incidents and accidents in highly automated aircraft show that even well trained operators may fail to understand the intention or behavior of the automation (Dornheim, 1995; Hughes, 1995a; Sarter & Woods, 1995). Automation technology is spreading rapidly to wider areas of application in which designers may not assume that every operator is substantially trained. An automobile is such an example.

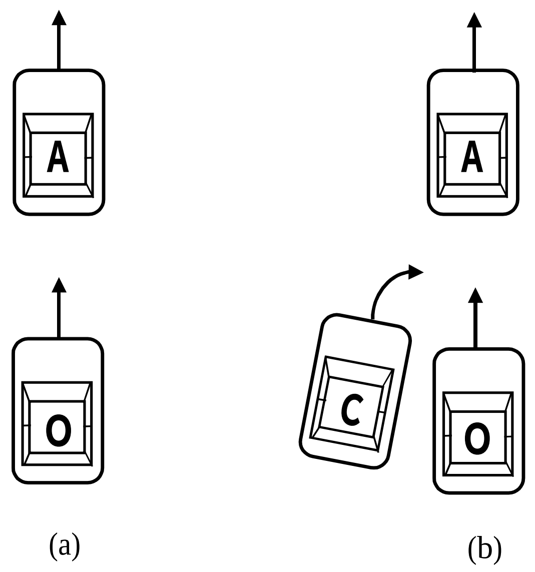

Fig. 8.3 Adaptive cruise control: (a) headway control; (b) cutting in.

*Example: Adaptive cruise control system.* Suppose an automobile company has a plan to develop an adaptive cruise control system with headway control capability, as shown in Fig. 8.3(a). The system is supposed to decelerate its own vehicle (O) when it approaches too close to the vehicle ahead (A). Designers have to design their system by enumerating various situational changes that may occur while driving. For instance, the third vehicle (C) may be cutting in between cars O and A, as shown in Fig. 8.3(b). It is important for the car O's driver to know whether the automation sensed the car C or not. Suppose the driver thought that the automation had sensed car C and had already changed its control law to follow car C, instead of car A. Then the driver may not hit the brakes, instead of expecting the automation to do so. A collision with car C may happen if the automation had not sensed the car C and was still controlling car O longitudinally so that the distance to car A might be maintained as specified. This suggests the need for a human interface that can show the automation's recognition of the situation.

When designing a human interface for the adaptive cruise control system just described, designers first need to enumerate all the possible state vectors, ($X$, $Y$), where $X$ denotes the car that the automation tries to follow, and $Y$ denotes the car that the driver thinks the automation is following. Among all the possible state vectors, some of them represent cases in which the driver and the automation do not share the situational recognition: (A, C), (C, A), (-, A), and (-, C) are some such cases, where the minus sign (-) implies that the automation has lost the target to follow, which may occur at a sharp bend in the road, or when the vehicle ahead changes its lane. Each state vector may yield a collision. For instance, in case of (A, C), collision against car C may happen if the driver of car O was overly reliant on the automation by hoping that the automation would decelerate in a moment. Even in case of (C, C), car O may collide with car C. The collision shall occur if the car C was cutting in too abruptly and too closely to car O.

In general, *fault tree analysis* techniques are useful for identifying causes of possible collision for each state vector. Though the methods were developed originally for assessing safety or risk of large-scale systems, they are now recognized as useful tools for cognitive task analyses (Kirwan & Ainsworth, 1992). Taking a state vector, say (A, C), as the *top event*, a fault tree must be constructed. The first step is to find combinations of events that may cause the top event to occur, where Boolean gates (such as AND gates and OR gates) are used in describing how events are combined. The process is repeated until no further expansion of an event is necessary. The events at the bottom of the fault tree are called *basic events*, which may represent various human factors (e.g., human errors, loss of awareness, complacency, or distrust), violations of laws, hardware/software failures, or hostile environmental conditions. The designers have to consider what means are available to prevent the basic events from occurring. The identified countermeasures must then be examined with theoretical analysis, computer simulations, or experiments. The fault trees obtained in this process are useful for systematic extraction of scenarios under which the efficacy of the design decisions must be examined.

## Conclusion: Toward Sensible Adaptive Automation

Design aspects for adaptive automation are summarized in Table 8.3. The first aspect (sharing and trading) may be regarded as physical collaboration between human operators and the automation. 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 partition, the automation is to be designed so that it can cope with every possible partition of functions. In case of extension or relief, we need a mechanism to add control force on the other agent’s force.

Table 8.3 Design aspects of adaptive automation

1. Sharing control or trading control
    - (1) Sharing control
        - (i) Extension
        - (ii) Relief
        - (iii) Partition
    - (2) Trading

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

The second aspect (automation invocation) requires analyses on the following:

(a) availability of a clear definition on the event in which it is hard for any operators to perform assigned functions properly, (b) availability of methods to measure indices precisely without placing a heavy burden on the operators, and (c) availability and precision of performance models that may represent a wide variety of operators or that may tune the model parameters dynamically.

The third aspect (decision authority) deals with mental collaboration between operators and the automation. The aspect is closely related to the principle of human-centered automation. However, it is not wise to assume, without careful analyses, that human operators must be maintained as the final authority at all times and for every occasion.

Theoretically speaking, 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, adaptive automation designers have to analyze possible consequences of design decisions:

- How do humans feel or respond when they are overridden by the automation for safety reasons?
- Do humans trust in and use the automation that works well but in a slightly different manner than anticipated?
- Do not operators become so reliant on adaptive automation that they may be reluctant to take actions themselves?
- Is appropriate human interface provided with the operators so that any automation surprises can be avoided?

These are some of questions that designers have to ask in the design stage. Because adaptive automation is more sophisticated, complex, and possibly obscure than conventional automation, the cognitive task design must be done seriously. Especially, the decision authority issue can never be solved with dogmatic or qualitative discussions. The issue has to be investigated in a rigorous manner by applying quantitative methods, such as mathematical modeling, computer simulations, and experiments.

Research on adaptive automation had its origin in aviation, and various studies have been conducted in the context of aviation applications, including military applications (Bonner, Taylot, & Fletcher, 2000; Morgan, Cook, & Corbridge, 1999). Rouse (1988), Parasuraman et al. (1992), Scerbo (1996), Wickens and Hollands (2000), and Scallen and Hancock (2001) give good surveys of those efforts. The adaptive automation concept can be applied to other areas, such as process control (Moray, Inagaki, & Itoh, 2000) and automobiles. It must be stressed here again that the design considerations for automobiles must be quite different from those for aircraft, large process plants, and military applications. Car drivers are not always well trained. They may not have perfect knowledge of the automation in their cars. The cognitive task analysis and design must take this point into account.

This chapter took the terms *function* and *task* to mean essentially the same thing. It is sometimes difficult to discriminate between the two terms without any ambiguity (Wickens et al., 1998), although there is a sort of hierarchical relation between function and task. Those readers who are interested in the hierarchical structure should refer to Sharit (1997).

**Acknowledgments**

The author expresses his thanks to Raja Parasuraman, Tom Sheridan, and Neville Moray for their useful discussions. Although they do not appear as coauthors, readers may find reflections of their ideas at many places in this chapter. The author also extends his sincere gratitude to anonymous referees. Without their critical but constructive comments and suggestions, this chapter could not be revised appropriately.

## REFERENCES

Archer, R.D., & Lockett, J.F. (1997). WinCrew – A tool for analyzing performance, mental workload and function allocation among operators. In *Proceedings of the Allocation of Functions Conference*, Galway Ireland.

Archer, R.D., Lewis, G.W., & Lockett, J.F. (1996). Human performance modeling of reduced manning concepts for Navy ships. In *Proceedings of the 40th Annual Meeting of the Human Factors and Ergonomics Society* (pp. 987-997). Santa Monica, CA: Human Factors and Ergonomics Society.

Banks, S.B. & Lizza, C.S. (1991). Pilot's associate. *IEEE Expert*, 6(3), 18-29.

Barnes, M. & Grossman, J. (1985). *The intelligent assistant concept for electronic warfare systems*. China Lake, CA: Naval Warfare Center (NWC TP 5585).

Billings, C. E. (1997). *Aviation automation – the search for a human-centered approach*. Mahwah, NJ: Lawrence Erlbaum Associates.

Bonner, M., Taylor, R., & Fletcher, K. (2000). Adaptive automation and decision aiding in the military fast jet domain. In *Proceedings of Human Performance, Situation Awareness, and Automation* (pp.154-159). Savannah, Georgia: SA Technologies, Inc.

Bresley, B. & Egilsrud, J. (1997). Enhanced ground proximity warning system. *Boeing Airliner*, July-September 1997, 1-13.

Dornheim, M. (1995, January 30). Dramatic incidents highlight mode problems in cockpits. *Aviation Week and Space Technology*, 142(5), 57-59.

Dornheim, M. (1996, September 9). Recovered FMC memory puts new spin on Cali accident. *Aviation Week and Space Technology*, 145(11), 58-61.

Endsley, M.R., & Kaber, D.B. (1997). The use of level of automation as a means of alleviating out-of-the-loop performance problems: A taxonomy and empirical analysis. In *Proceedings of the 13th Triennial Congress of the International Ergonomics Association* (pp.168-170). Helsinki: Finnish Institute of Occupational Health.

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

Fitts, P.M. (Ed.)(1951). *Human engineering for an effective air-navigation and traffic-control system*. Columbus, OH: The Ohio State University Research Foundation.

Furukawa, H, Niwa, Y., & Inagaki, T. (2001). Levels of automation in emergency operating procedures for a large-complex system. In M.J. Smith, G. Salvendy, D. Harris, & R.J. Koubek (Eds.). Usability evaluation and interface design (Vol. 1, pp.1513-1517). Mahwah, NJ: Lawrence Erlbaum Associates.

Glenn, F., Barba, C., Wherry, R.J., Morrison, J., Hitchcock, E., & Gluckman, J.P. (1994). Adaptive automation effects on flight management task performance. In M. Mouloua, & R. Parasuraman (Eds.), *Human performance in automated systems*. Erlbaum, (pp.33-39). Hillsdale, NJ: Mahwah, Lawrence Erlbaum Associates.

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 *Proceedings of the 7th International Symposium on Aviation Psychology* (pp.150-155). Columbus, OH: The Ohio State University.

Greenstein, J.S., & Revesman, M.E. (1986). Development and validity of a mathematical model of human decisionmaking for human-computer communication. *IEEE Transactions on Systems, Man, & Cybernetics*, 16, 148-154.

Grote, G., Ryser, C., Wafler, T., Windischer, A., & Weik, S. (2000). KOMPASS: A method for complementary function allocation in automated work systems. *International Journal of Human-Computer Studies*, 52, 267-287.

Hammer, J.M. & Small, R.L. (1995). An intelligent interface in an associate system. *Human/Technology Interaction in Complex Systems*. JAI Press, 7, 1-44.

Hancock, P.A., & Chignell, M.H. (1988). Mental workload dynamics in adaptive interface design. *IEEE Transactions on Systems, Man, & Cybernetics*, 18, 647-658.

Hancock, P.A. & Scallen, S.F. (1998). Allocating functions in human-machine systems. In R.R. Hoffman, et al. (Eds.). *Viewing psychology as a whole* (pp.509-539). Washington, DC: American Psychological Association.

Harris, W.C., Hancock, P.A., Arthur, E.J., & Caird, J.K. (1995). Performance, workload, and fatigue changes associated with automation. *International Journal of Aviation Psychology*, 5, 169-185.

Harris, W.C., Hancock, P.A., & Arthur, E.J. (1993). The effect of taskload projection on automation use, performance, and workload. In *Proceedings of the 7th International Symposium on Aviation Psychology* (pp.178-184). Columbus, OH: The Ohio State University.

Harris, W.C., Goernert, P.N., Hancock, P.A., & Arthur, E.J. (1994). The comparative effectiveness of adaptive automation and operator initiated automation during anticipated and unanticipated taskload increases. In M. Mouloua & R. Parasuraman (Eds.), *Human performance in automated systems* (pp.40-44). Hillsdale, NJ: Lawrence Erlbaum Associates.

Hilburn, B., Molloy, R., Wong, D., & Parasuraman, R. (1993). Operator versus computer control of adaptive automation. In *Proceedings of the 7th International Symposium on Aviation Psychology* (pp.161-166). Columbus, OH: The Ohio State University.

Hughes, D. (1995a, January 30). Incidents reveal mode confusion. *Aviation Week and Space Technology*, 142(5), 56.

Hughes, D. (1995b, February 6). Studies highlight automation ‘surprises’. *Aviation Week and Space Technology*, 142(6), 48-49.

Inagaki, T. (1997). To go no not to go: Decision under time-criticality and situation-adaptive autonomy for takeoff safety. In *Proceedings of the IASTED International Conference on Applied Modelling and Simulation* (pp. 144-147). Calgary: IASTED ACTA Press.

Inagaki, T. (1999). Situation-adaptive autonomy: Trading control of authority in human-machine systems. In M.W. Scerbo & M. Mouloua (Eds.), *Automation technology and human performance: Current research and trends* (pp.154-159). Mahwah, NJ: Lawrence Erlbaum Associates.

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

Inagaki, T., Takae, Y., & Moray, N. (1999). Automation and human interface for takeoff safety. In *Proceedings of the 10th International Symposium on Aviation Psychology* (pp.402-407). Columbus, OH: The Ohio State University.

Jordan, N. (1963). Allocation of functions between man and machines in automated systems. *Journal of Applied Psychology*, 47, 161-165.

Kaber, D.B. & Riley, J.M. (1999). Adaptive automation of a dynamic control task based on workload assessment through a secondary monitoring task. In M.W. Scerbo & M. Mouloua (Eds.), *Automation technology and human performance: Current research and trends* (pp.129-133). Mahwah, NJ: Lawrence Erlbaum Associates.

Kaber, D.B., Omal, E., & Endsley, M.R. (1999). Level of automation effects on telerobot performance and human operator situation awareness and subjective workload. In M.W. Scerbo and M. Mouloua (Eds.), *Automation* *technology and human performance: Current research and trends* (pp.165-169). Mahwah, NJ: Lawrence Erlbaum Associates.

Kawazoe, H., Murakami, T., Sadano, O., Suda, K., & Ono, H. (2001). Development of a lane-keeping support system. In *Proceedings of Intelligent Vehicle Initiative (IVI) Technology and Navigation Systems* (p.29-35). Warrendale, PA: Society of Automotive Engineers.

Kirwan, B. & Ainsworth, L.K. (1992). *A guide to task analysis*. London: Taylor & Francis.

Klein, G., Orasanu, J., Calderwood, R, & Zsambok, C.E. (Eds.) (1993). *Decision making in action: Models and methods*. Norwood, NJ: Ablex.

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

Morgan, C., Cook, C.C., & Corbridge, C. (1999). Dynamic function allocation for naval command and control. In M.W. Scerbo & M. Mouloua (Eds.), *Automation technology and human performance: Current research and trends* (pp.134-138). Mahwah, NJ: Lawrence Erlbaum Associates., Erlbaum.

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

Morris, N.M., Rouse, W.B., & Ward, S.L. (1988). Studies of dynamic task allocation in an aerial search environment. *IEEE Transactions on Systems, Man, & Cybernetics*, 18(3), 376-389.

Parasuraman, R., Bhari, T., Molloy, R., & Singh, I. (1991). Effects of shifts in the level of automation on operator performance. In *Proceedings of the 6th International Symposium on Aviation Psychology* (pp.102-107). Columbus, OH: The Ohio State Univesity.

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

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

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

Parasuraman, R., Sheridan, T.B., & Wickens, C.D. (2000). A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, and Cybernetics*, 30(3), 286-297.

Price, H.E. (1985). The allocation of function in systems. *Human Factors*, 27(1), 33-45.

Revesman, M.E. & Greenstein, J.S. (1986). Application of a mathematical model of human decision making for human-computer communication. *IEEE Transactions on Systems, Man, and Cybernetics*, 16(1), 142-147.

Rouse, W.B. (1976). Adaptive allocation of decision making responsibility between supervisor and computer. In T.B. Sheridan & G. Johannsen (Eds.), *Monitoring behavior and supervisory control* (pp.295-306). New York: Plenum.

Rouse, W.B. (1977). Human-computer interaction in multitask situations. *IEEE Transactions on Systems, Man, and Cybernetics*, 7, 384-392.

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

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

Rouse, W.B. (1994). Twenty years of adaptive aiding: Origins of the concept and lessons learned. In M. Mouloua & R. Parasuraman (Eds.), *Human performance in automated systems: Current research and trends* (pp.28-32). Hillsdale, NJ: Lawrence Erlbaum Associates.

Rouse, W.B., Geddes, N.D., Curry, R.E. (1987-1988). An architecture for intelligent interfaces: Outline of an approach to supporting operators of complex systems, *Human-Computer Interaction*, 3, 87-122.

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

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

Scallen, S.F., Hancock, P.A., & Duley, J.A. (1995). Pilot performance and preference for short cycles of automation in adaptive function allocation. *Applied Ergonomics*, 26(6), 397-403.

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

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

Scerbo, M.W., Freeman, F.G., Mikulka, P.J., Parasuraman, R., Di Nocero, F., & Prinzel III, L.J. (2001). *The efficacy of psychophysiological measures for implementing adaptive technology* (NASA/TP-2001-211018). Hampton, VA: NASA-Langley Research Center.

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

Sharit, J. (1997). Allocation of functions. In G. Salvendy (Ed.). *Handbook of* *human factors and ergonomics* (2nd ed., pp.301-339). New York: Wiley.

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

Sheridan, T.B. (2002). *Humans and automation: System design and research issues.* Santa Monica, CA: Human Factors and Ergonomics Society & Wiley.

Sheridan, T.B., & Verplank, W.L. (1978). *Human and computer control of undersea teleoperations.* (Tech. Rep.). Man-Machine Systems Laboratory, Department of Mechanical Engineering, MIT, Cambridge, MA.

Sparaco, P. (1994, January 3). Human factors cited in French A320 crash. *Aviation Week and Space Technology*, 140(1), 30-31.

Walden, R.S., & Rouse, W.B. (1978). A queueing model of pilot decisionmaking in amultutask flight management situation, *IEEE Transactions on Systems, Man, & Cybernetics*, 8(12), 867-875.

Wickens, C.D. (1984). Processing resources in attention. In R. Parasuraman & D.R. Davies (Eds.) *Varieties of attention* (pp.63-101). Orland, FL: Academic Press.

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

Wickens, C.D., Gordon, S.E., & Liu, Y. (1998). *An introduction to human factors engineering.* reading, MA: Addison-Wesley.

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

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

Wiener, E.L. (1988). Cockpit automation. In E.L. Wiener & D.C. Nagel (Eds.). *Human factors in aviation* (pp.433-461). New York: Academic Press.

Wiener, E.L. (1989). *Human factors of advanced technology (glass cockpit) transport aircraft* (NASA Contractor Rep. No. 177528). Moffett Field, CA: NASA-Ames Research Center.

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

Zsambok, C.E. & Klein, G. (1997). *Naturalistic Decision Making.* Mahwah, NJ: Lawrence Erlbaum Associates.