Adaptive Automation: Sharing and Trading of Control

Toshiyuki Inagaki

February 25, 2002

ISE-TR-02-187

Address for correspondence:
Professor Toshiyuki Inagaki
Institute of Information Sciences and Electronics
University of Tsukuba
Tsukuba 305-8573
inagaki@is.tsukuba.ac.jp

Adaptive Automation: Sharing and Trading of Control

TOSHIYUKI INAGAKI

Institute of Information Sciences and Electronics
University of Tsukuba, Japan

Introduction

One of the central issues in the research of human-machine systems is how functions can be allocated to human and machine. The design decision of assigning functions to human and machine is called *function allocation*. How can we determine function allocation? One of the easiest ways is to allocate each function to an agent (human or machine) with ability that is superior to that of the other. Or, we may allocate to machine every function that can be automated. Or, we may determine assignment of function just from the viewpoint of developing cost of the human-machine system. These function allocation schemes are static in nature, because, once a function is assigned to an agent, the agent must be in charge of the function forever. *Static function allocation* is easy to implement. However, the resulting function allocation may not be advantageous for humans. For instance, human may be overloaded or underloaded, or may fail to feel any job satisfaction.

Dynamic function allocation is a scheme in which a responsible agent for a function can vary occasionally during the period of system operation. Many of recent systems are designed so that human and machine can share responsibilities for functions and can cooperate dynamically. Dynamic function allocation is expected to give more flexibility and promise better performance than static function allocation. When friendliness to human is emphasized, we come to the concept of *adaptive function allocation*.

In adaptive function allocation, functions can be reassigned to human and machine in response to changes in situations or human performance (Rouse, 1988; Parasuraman, Bhari, Deaton, Morrison, & Barnes, 1992; Scerbo, 1996; Hancock & Scallen, 1998). Thus adaptive function allocation is dynamic in nature. One of main aims of adaptive automation is to regulate human workload. During periods of moderate workload, human may control a process. Human may hand off control of particular functions when workload becomes too high. Automation that operates under an adaptive function allocation scheme is called *adaptive automation*. Adaptive automation assumes criteria to determine whether functions need be reallocated, how, and when. There are some types of automation invocation algorithms that trigger automation in response to various factors, such as human performance, occurrence of specific events that may impose high workload to human, psychophysiological state of human, and so on.

It is well-known that humans working with highly autonomous systems often suffer negative consequences of automation, such as the out-of-the-loop performance problem, loss of situation awareness, automation-induced surprises (see, e.g., Wickens, 1995; Endsley & Kiris, 1995; Sarter and Woods, 1995; Sarter, Woods, & Billings, 1997). Adaptive automation may not also be free from those negative consequences. Moreover, some types of adaptive automation may violate the fundamental principle of human-centered automation claiming that, "the human must be maintained as the final authority over the automation" (Woods, 1989; Billings, 1991, 1997).

This chapter describes why and how the concept of adaptive automation was invented, starting with clarification of concept of and drawbacks in static function allocation, followed by description of dynamic function allocation. *Levels of automation* concept by Sheridan (1992) is utilized in explaining what kind of cooperation may be possible between human and automation in information acquisition, information analysis, decision and action selection, and action implementation. *Sharing* and *trading* are distinguished explicitly to clarify the style of human-automation cooperation. Algorithms for implementing adaptive automation are categorized into three groups, and comparisons are made among them. Decision authority and trust-related issues are also discussed. Benefits and costs of adaptive automation are described for better understanding and for future research activities.

Functional Allocation

Functions and tasks

Suppose we are to design a human-machine system. We have to ask ourselves various questions, and careful answers must be given to reach a good design solution. The following are some of such questions: "What are the major functions to be performed by the system, whether it be by person or machine?" "What tasks must be performed?" (Wickens, Gordon, & Liu, 1998).

What do *function* and *task* mean in the above? The term function can often be interpreted as a goal or activities of a system. Suppose we are designing an automatic teller system. Some major functions for the system could be defined as follows (Wickens, *et al.*, 1998): (1) Get person's funds into bank account; (2) Get funds from bank account.

The term task is used for a more detailed description of behavior of a system to carry out its functions. Wickens *et al.* (1998) identify the tasks for the automatic teller system as follows: (1) Withdraw money from bank checking (or, savings) account; (2) Deposit money into bank checking (or, savings) account; (3) Determine balance of the checking (or, savings) account; (4) Withdraw money from credit card account.

It may be said that a function gives a bit abstract description of an activity or behavior of a system, and a task yields a more detailed description of it in a context-specific manner. In the process of developing precise descriptions of functions, we come down to descriptions of tasks that reflect contexts in the use of the system.

In reality, however, distinction between functions and tasks are not always clear. Wickens and his colleagues (1998) say, "Often it is difficult to discriminate the function list from the preliminary task analysis list." In fact, the terms functions and tasks are often used interchangeably among human factors specialists, though there are some efforts that try to distinguish the two terms in a more rigorous manner, see, e.g., (Scallen, 1997; Hancock & Scallen, 1998). In the present article, the terms function and tasks are sometimes used interchangeably.

Allocation of functions

Suppose we have identified functions that are needed to accomplish the goals of a human-machine system. A next and very important question to be asked is who performs each function. The process to assign each function to human or machine component is called function allocation.

More specifically, "Function allocation refers to the conscious design decisions which determine the extent to which a given job, task, function, or responsibility is to be automated or assigned to human performance. Such decisions should be based upon aspects such as relative capabilities and limitations of humans versus machines in terms of reliability, speed, accuracy, strength and flexibility of response, cost, and the importance of successful and timely task or function accomplishment to successful and safe operations." (Wickens, *et al.*, 1998).

In spite of its importance, function allocation has not become a science yet, but still a kind of art. Sharit (1997) points out, "Many human factors and ergonomics specialists have more or less acknowledged that systematic methods do not exist for allocating functions to humans and machines, especially in highly complex systems." However that does not exclude possibility to classify function allocation methods in a systematic manner. Rouse (1991) gives an aspect that classifies function allocation methods that have appeared in the literatures into three categories.

The first category is called *comparison allocation*. Methods in this category firstly identify the skill requirements and performance criteria for each function, and compare relative abilities of human and machine. Then, allocate each function to an agent (human or machine) with ability that is superior to that of the other. The most famous list that compares relative abilities of humans and machines may be the one edited by Fitts (1951), see Table 1. The list is sometimes called MABA-MABA (what "men are better at" and what "machines are better at") list.

Table 1 The Fitts List

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

1. Ability to detect small amount 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 1950's) 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.

(after Fitts, 1951; Price, 1985; Hancock & Scallen, 1998)

The second category is called *leftover allocation*. Methods in this category try to automate everything that can be automated. Humans are assigned the leftover functions to which no technologies are available to automate them. This type of function allocation is technology-driven, and yields so-called *technology-centered automation* (Woods, 1989).

The third category is called *economic allocation*. Methods in this category determine function allocation from a viewpoint of costs for developing and operating the human-machine system. Even when some technology is available to automate a function, the function is not automated and is left to humans, if it costs too high to automate the function.

Some problems in function allocation

The allocation scheme seems to be simple for each category of methods. However, the resulting design may not be so simple. The design can yield complex working environment that may not be very comfortable for humans. Suppose a leftover or an economic approach is taken. The set of functions assigned to humans may produce a strange shape that is hard to remember: Imagine that the whole set of functions to be performed by the human-machine system has inherently a round shape, like a pizza. Now automations begin to *eat* their favorite parts. They eat and eat lots of parts of the pizza. What remains for human then? He or she may find odd-shaped remnants that are hard to believe that there was a round pizza. Once this situation would happen in a human-machine system, humans might get confused and ask him/herself, "Why automation is doing the task?" "Am I expected to do this task?" and so on. Unfortunately, this sometimes happens in reality. Also, there is no guarantee that the assigned functions can provide with job satisfaction to humans (Lockhart, Strub, Hawley, & Tapia, 1993).

Comparison allocation may seem to be a bit nicer, at least for a human operator, than economic or leftover allocation. However this approach is not free from criticisms, either. Price (1985) and Sharit (1997) claim that the Fitts list is overly generalized and non-quantitative. It is noted also that, even though humans are given only functions that fit to relative advantages of humans, the sum total of allocation decisions may result in a situation where humans are overloaded or underloaded (Sharit, 1997). Moreover, if a function can be equally performed, or badly performed by humans and machines, the Fitts list does not give an answer to which agent the function must be assigned. Sheridan (2001) also points out that, "in order to make use of the Fitts MABA-MABA list, one needs data which are context dependent, but these data are mostly not available."

Hancock and Scallen (1998) argue that the Fitts list has not been understood appropriately. They give nine points to be noted. Some of the points are: (1) The goal of the report was to improve air navigation and traffic control systems (in those days), with little or no consideration directed toward other domains. (2) The goal in formally stating objectives was to inspire research. (3) The report does not deal with technical feasibility, economic, manpower, and personnel issues (Hancock & Scallen, 1998). Sheridan (2001) argues, referring to the ideas of Jordan (1963), "we should throw out the idea of comparing man and machine but keep the facts about what people do best and what machine do best," and "the main point of retaining the Fitts list is that people and machine are complementary." An example for the *complementary* approach may be found in KOMPASS (Grote *et al.*, 2000), which is described in one of the chapters in this handbook.

One important point to note here is that, we have discussed so far only design decisions that determine who does what. Such design decisions yield function allocations that are *static* in nature. Once a function is allocated to an agent (human or machine), the agent is responsible for the function forever.

Static function allocations are sometimes inappropriate or ineffective. Suppose a designer allocated a human operator a function to detect small amounts of visual energy, according to the observation given in the Fitts list. Even though he or she may basically surpass any machine in the function, it may not be guaranteed that the superiority shall hold at all times and on every occasion, especially after extended operating hours. He or she may simply get tired or bored, and some events may distract his or her attention. Or, even if he or she may be good at a function when enough amount of time is available, it may be very tough for him or her to perform the function under extreme time pressure. It is thus unrealistic to assume that the MABA-MABA-type statements do not usually hold unconditionally without imposing certain conditions or context specifications. It would be easy to see that leftover and economic allocations are also basically static in nature, and that they share the same drawback with comparison allocation.

Why static function allocations may not be recommended can be discussed from a mathematical point of view. Suppose we are going to develop a human-machine system the supposed operation period of which is ten years. Consider the following design decision problem.

Problem 1: Find an optimal (or the best) function allocation between human and machine so that the performance index may be maximized subject to the following constraints:

(1) Every goal of the human-machine system is satisfied.

(2) Allocation scheme is feasible for either humans or machines; e.g. the sum total of resources that allocation decisions impose may never violate resource constraint of any agent (human or machine) in the system.

(3) Function allocation among agents may never be changed during the period of ten years.

In the above statement of the decision problem, *performance index* (or, an *objective function*) to be maximized may represent various measures, such as, the expected profit the system may produce, safety of the system. For any optimization problem, the following two properties hold:

Property A: The more we get constrains, the smaller a feasible region becomes, where a feasible region refers to the whole set of decision alternatives that satisfy imposed constraints.

Property B: The smaller a feasible region becomes, the smaller the maximum attainable value of the objective function becomes.

Now we define a new decision problem.

Problem 2: Find an optimal function allocation between humans and machines so that the performance index may be maximized subject to the following constraints:

(1) Every goal of the human-machine system is satisfied.

(2) Allocation scheme is feasible for either humans or machines.

Problem 2 is exactly the same as Problem 1, except Problem 1's third constraint, "Function allocation among agents may never be changed during the period of ten years" has been removed in Problem 2. Noting Properties A and B, a designer can be sure that his or her design decision for Problem 2 will give a better result than that for Problem 1 can.

What does it mean to discard the third constraint? That means that a designer is allowed to consider a function allocation scheme in which responsibilities for functions may be exchanged between human and machine occasionally during the period of ten years. The exchange may happen several times a year, or a month, or a week, or a day, or even an hour, which means that the function allocation is *dynamic*. Static function allocation just states who does what. However, dynamic function allocation determines who does what, when, and how. It is worth noting that Problem 2 may give a static function allocation as its optimal solution, because static function allocations are special cases of dynamic allocations.

Dynamic function allocation

Need of dynamic function allocation

Dynamic function allocation is defined as a scheme in which a responsible agent for a function may vary from time to time during the period of system operation. It is easy to find such dynamic function allocations in real human-machine systems. Take a commercial aircraft, as an example. Aircraft in recent years are equipped with various computers that can perform important functions to make flights safe, smooth, and efficient. Management of lateral flight path (LNAV) and vertical flight path (VNAV) are such essential functions for flying. Human pilot and computer share responsibilities for these functions, and cooperate dynamically. More concretely, human pilot usually takes responsibilities for both of LNAV and VNAV during takeoff. In the climb phase, the pilot may take only LNAV and ask the computer to deal with VNAV. During cruise, pilot often hands both LNAV and VNAV over to the computer. In descending or landing, pilot may seize control of either LNAV or VNAV back again. The two functions are allocated in different ways depending on situations.

What happens, if only a static function allocation was allowed in the above example? Since no time-dependent alteration is allowed for assignment of functions, there would be only the following four possible design alternatives: (1) No machine needs to be designed because human pilot is to take care of both LNAV and VNAV all the time during flight; (2) Design a machine that can perform LNAV at all times and in every occasion; (3) Design a machine that can perform VNAV at all times and in every occasion; (4) Design a computer that can perform LNAV and VNAV all the time during flight. Among these alternatives, the forth is almost infeasible or inappropriate. It is easily seen how odd and stiff a resulting working environment can be for human pilots.

Rouse (1976) investigated human-computer interaction in multitask situations based on an insight that, "while some tasks are best performed by the human and others are best performed by the computer, there are many tasks that could be successfully performed by either human or computer." The multitask situations investigated are summarized as follows: Each task is characterized by a state vector. By scanning appropriate displays, an agent (human or computer) can obtain the best estimate of a state vector, where information obtained through observation may be noisy. Moreover, the agent may not scan all of displays during every scan. Given an observation, the agent must decide whether an undesirable event is occurring and the corresponding task needs some countermeasure action. If attention was not carefully distributed over tasks, there may be possibility that the event may be overlooked and it is too late to take a necessary action to the corresponding task.

The problem that has to be solved is how responsibility must be allocated between human and computer so that a specific measure of performance defined over a planning horizon could be maximized subject to the constraint that human workload must be below a specific level. Rouse (1976) and Chu & Rouse (1979) formulated the problem by applying a queuing theory. They have derived a threshold policy for turning the computer on and off, and have shown that the policy minimizes event-waiting cost subject to human workload constraint. The result is a proof that dynamic allocation of function is meaningful and effective, compared to static allocation of function.

There are some other studies that prove mathematically that the framework of dynamic function allocation is more natural and can give better human-computer interaction than that of static function allocation; see, e.g., (Millot & Willaeys, 1985;

Millot, Taborin, Kamoun, & Willaeys, 1988; Inagaki & Johannsen, 1992; Inagaki, 1993).

Needless to say, even in a dynamic function allocation, a single specific agent, human or computer may always perform some function. Suppose we have three functions, A, B, and C for three agents. Let an allocation scheme be such that function A is always assigned to agent 1, while agents 2 and 3 trade functions B and C occasionally. This scheme is dynamic, because an agent responsible for function B (or C) may vary, though function A is always assigned to agent 1. This kind of dynamic allocation with a partially static characteristic can be sometimes of importance.

One of such dynamic function allocations can be seen in (Sharit, 1998) that discusses a function allocation for visual inspection systems. The investigated situation is as follows: An appropriate inspection is needed to achieve high quality standards of products that consist of computer-generated 3-D height maps of electronic circuit boards. Human inspector may not have perfect capabilities for detecting flaws, such as missing components, wrong-sized components, and misaligned components. However, an automated inspection system can be worse than human inspectors depending on the product. Thus a hybrid inspection system may be the third possibility, in which human and machine capabilities are utilized jointly. An inspection task has two primary functions: *search* and *decision making* that are performed sequentially. Each of the two primary functions may be allocated to human, computer, or hybrid. Among nine possible design alternatives, Sharit (1998) discarded five of them by taking into account "the relative advantages of computers in performing search functions and of humans in performing decision-making functions" (Sharit, 1998). The following four systems are selected for further investigation. (1) Pure human inspection. The computer only presents the images to the human inspector; (2) The computer performs the search task. When it detects flaws, they are shown to the human inspector who decides on their status; (3) The computer performs both the search and decision-making functions. When the computer is uncertain about its decision, the human intervenes in the decision-making process; (4) Fully automated inspection.

Sharit (1998) investigated, with three subjects, 27 inspection scenarios in which each of the following three factors may take three different levels: product complexity represented in terms of the number of components on the circuit board, contrast level between circuit board and background, and visual noise in the displayed image. The four design alternatives (1) - (4) were compared in terms of speed (seconds per board inspected) and accuracy (hits and false alarms), where the two viewpoints were combined into a single objective function (or, performance index) that expresses the inspection cost per board. The conclusion obtained was that "the computer-search human-decision hybrid system was faster and more accurate than either the purely human or the joint human-computer decision-making inspection system" (Sharit, 1998).

Choosing levels of automation for each function

In the example of Sharit (1998), we have seen descriptions (1)-(4) stating what human or computer does and when, which are context-specific statements for the roles of human and computer. Context-free descriptions may be found in the lists of levels of automation (LOA), the concept of which was originated by Sheridan & Verplank (1978). Table 2 gives a compact list of LOA, a bit simplified by Sheridan (1992) based on that given in (Sheridan & Verplank, 1978).

Table 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.

(Sheridan, 1992)

In terms of LOA, we may be able to say that Sharit (1998) tries to select an appropriate LOA for two sequentially performed primary functions, searching and decision making. A bit more extended argument on design decisions for complex-human machine systems can be found in (Parasuraman, Sheridan, & Wickens, 2000; Sheridan, 2001).

Parasuraman *et al.* (2000) and Sheridan (2001) give a model for describing various types and levels of human interaction with computer. They distinguish four classes of functions in complex human-machine systems: (1) Information acquisition: (2) Information analysis: (3) Decision and action selection: (4) Action implementation. The distinction of these four classes of functions is a reflection of a four-stage view of human information processing, which is depicted in Figure 1. The functions (1) - (4) have been extracted as those that may be automated in the human information processing.

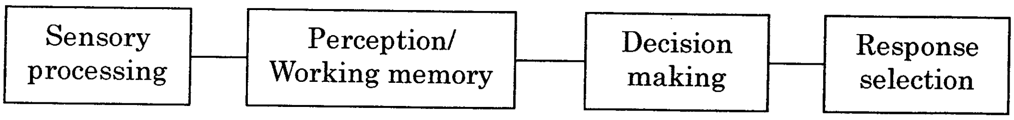

Figure 1 Simple four-stage model of human information processing (Parasuraman, *et al.*, 2000)

As was the case of the visual inspection systems investigated in (Sharit, 1998), there can be various design alternatives regarding to what extent each of the four functions (1)-(4) may be automated. In other words, a designer has to select an appropriate LOA for each of functions, (1)-(4). LOA may differ from function to function. According to (Parasuraman, *et al.*, 2000), automated forms of functions (1)-(4) are called respectively as acquisition automation, analysis automation, decision automation, and action automation. Characteristics of each of automation will be described next.

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

Filtering is usually useful for human to keep concentrate to some other important tasks. At the same time, filtering has some costs. Filtering may mislead operator understanding on the situation. There is an aircraft incident in which rapid decompression occurred in the cabin due to an explosion of a grenade. The explosion caused some failures of mechanisms for control wings. Warning messages appeared immediately on the cockpit display to let the pilots know failures of some mechanisms. Pilots took at once necessary countermeasure actions. However, they did not give oxygen masks to passengers in the cabin for a while, because the information "decompression" was not displayed to the pilots. The priority of the message for cabin decompression had been set at much lower values than those of messages announcing control system failures that may threat aircraft safety directly and immediately.

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 of such processed information may be *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, a large-screen shows trend graphs for various parameters with their predicted values. On our notebook computers, battery information is shown on the display that can tell, when we click its icon, how long the battery will be alive. In the cockpit of aircraft that is at a level flight, its navigation display may indicate an arc that shows at which point the aircraft is to begin a descend (or, a climb). If the pilot supplies an input to let the computer know a change of the flight plan, the computer replaces the old arc with a new one. For implementing capabilities of performing prediction, the computers must be given system dynamics models of the object (e.g., a nuclear reactor, battery on a notebook computer, aircraft).

Another type of analysis automation may be the one that can integrate multidimensional information into an easily understandable form. The resulting form may be a single value, or may be a graph. Graphical fusion of multidimensional information has been already seen in Chernoff's faces (Chernoff, 1973). The DURESS-like ecological interface (Vicente & Rasmussen, 1992) is also a good example of analysis automation. A more recent example may be the enhanced ground proximity warning system (EGPWS) for aircraft. The EGPWS is designed to complement the current GPWS functionality with the addition of look-ahead terrain alerting and terrain display. 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 is shown on the navigation display, or it equivalents, in dotted 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. Automation for decision and action selection may be already easy to imagine. However, if we distinguish two classes for decision, our discussion might become a bit clearer. Researchers in naturalistic decision making distinguish *situation-diagnostic decisions* and *course of action selection decisions* (Klein, Orasanu, Calderwood, & Zsambok, 1993; Zsambok, 1997; Klein, 1998). A situation-diagnostic decision refers to identify "what is going on," or, to pick up the most appropriate hypothesis among a set of diagnostic hypotheses. A course of action selection decision means to select the most appropriate action among a set of action alternatives. Among traditional expert systems, MYCIN (Shortliffe, 1976) is a typical example for automating situation-diagnostic decisions. When inference has to be done with imprecise information, an expert system may give humans a set of plausible diagnostic hypotheses with degree of belief information. The LOA of the expert system is positioned at levels 2 or 3 in Table 2. If, on the other hand, the expert system shows humans only a single diagnostic hypothesis with the largest degree of belief among all, the LOA is set at level 4. Traffic alert and collision avoidance system (TCAS) is a typical example of automation for course of action selection decision. The TCAS gives pilots a resolution advisory (RA), such as "Climb, Climb, Climb," when a mid-air collision may be anticipated if no resolution maneuver is taken. Pilots are supposed to initiate the suggested maneuver within five seconds. It is known, however, that the TCAS can produce unnecessary RA, though such cases do not happen frequently. Pilot may disregard RA of the TCAS when he or she is definitely sure that no resolution maneuver is necessary. In this sense, LOA of the TCAS RA is positioned at level 4. There is also automation with very high LOA for action selection. Computer software is often too dignified. Many of us have experience in which computer never allows us to escape from supplying information one after another till the end, even when we have lost interest in doing so at some point in the middle of the long sequence of actions.

Action automation. Automation for action implementation is easy to imagine. A photocopy machine, described in (Parasuraman, *et al.*, 2000), is a good example to illustrate that various LOA can be chosen in a single machine. Suppose a man was asked by his boss to photocopy documents of ten pages for five people. His boss needs them as soon as possible for distribution at an important meeting in progress. He must decide which mode to use, automatic sorting without automatic stapling, automatic sorting with automatic stapling, or manual mode to make five copies for each sheet. In the last case, he himself must sort and staple sheets manually. Time required for giving necessary directive to the machine through a touch sensitive panel differs from mode to mode. Time needed to finish the task differs also from mode to mode. Once he has chosen one of the modes, operation starts at one of three different levels of automation.

In aviation, LOA of action automation is not set high. From the viewpoint of action automation, 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 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 almost impossible 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.

Appropriate levels of automation. Parasuraman *et al.* (2000) argues that LOA may differ between automation for information acquisition, information analysis, decision and action selection, and action implementation. A committee of U.S. National Research Council has discussed appropriate levels of automation for new civil air traffic control systems (Wickens, Mavor, Parasuraman, & McGee, 1998). Sheridan (2001) reports, "After much debate the committee decided that acquisition and analysis could and should highly automated – in fact they already are (radar, weather, schedule information, etc.) However 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." Their recommended LOA for each function is depicted in Figure 2.

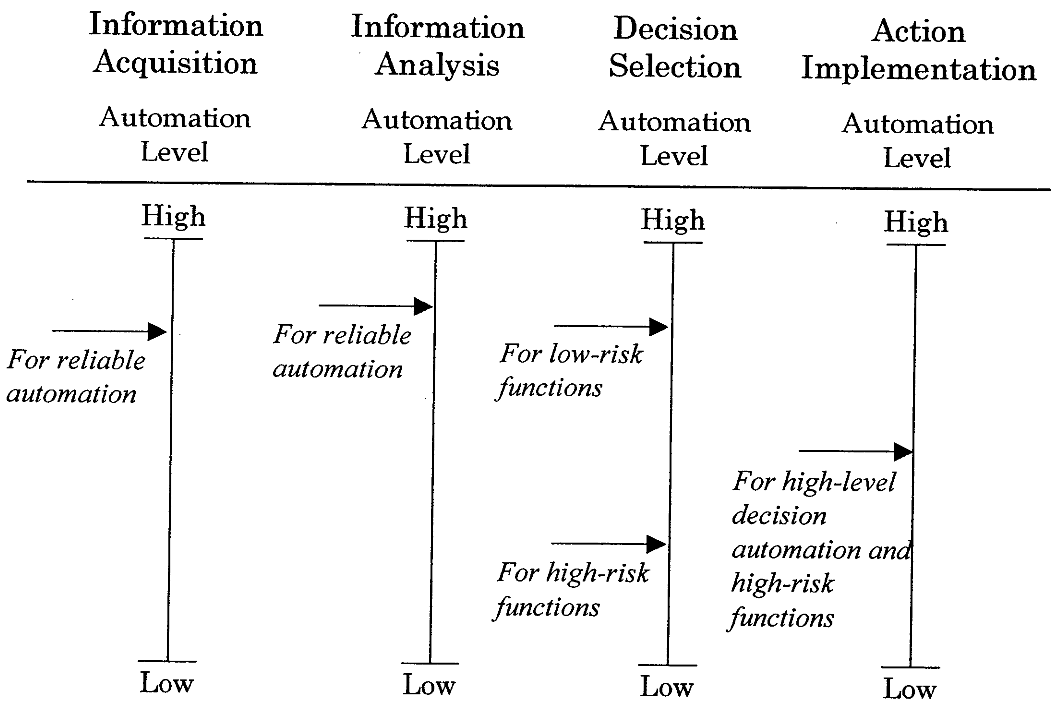

Figure 2 Recommended types and levels of automation for future ATC systems (Parasuraman et al., 2000)

Morgan, Cook, & Corbridge (1999) have investigated a potential design option in future Naval Command and Control Systems, the model of which has some similarities with that of Parasuraman *et al.* (2000). The investigators formulate a dynamic function allocation problem in which functions are distributed variably in real time between the human and the system to achieve optimal system performance on countering all incoming threats to ensure own force safety. Three-stage model is adopted to describe naval command and control tasks: (1) Compilation of the tactical picture; (2) Situation assessment and threat prioritisation; (3) Allocation of resources to meet the assessed threat. The first and the second stages are already highly automated, which is compatible with the suggestion of Parasuraman *et al.* (2000). For the third stage, various LOA can be adopted. Morgan *et al.* (1999) have shown, through an experiment with realistic scenarios, that the overall system performance was much better when human decision maker was able to allocate resource allocation task to the computer dynamically, compared to the cases in which he or she was required to perform the task manually throughout the experiment.

It must be noted that choosing different LOA between four functions does not necessarily mean that function allocation is dynamic. For instance, function allocation between human and computer is static if the selected LOA is fixed at either one of levels 1 to 4, or 7 to 10. Human and computer may exchange responsibility for a function only when level 5 or 6 was chosen. As long as one of the levels 5 and 6 may be selected for some function, it is necessary for us to classify the above model into dynamic function allocation. Also, the model does not exclude possibilities that LOA may be changed from one position to another, depending on the situation encountered. The resulting function allocation can be dynamic in general. For instance, there can be useful cases in which computer makes a decision and implements its associated action when human does not do so. Further discussions on this point will be made in later sections.

It is also important to note the inter-dependence of automation for decision and action selection and automation for action implementation for high-risk functions (Parasuraman *et al.*, 2000). Moreover, automation for information analysis may not be chosen independently from automation for decision and action selection, because it has been shown that there exist cases in which an information processing algorithm and an control action may not be selected independently with each other when safety of the system is a factor (Inagaki, 1991).

Some readers might think that the four-stage model in Fig. 1 may be overly simplified, comparing with real information processing that can occur in the human brain. One of critical comments might be that human information processing does not always proceed in a strict serial sequence. Suppose an operator noticed, while monitoring a plant, that one of plant parameters started to deviate rather swiftly from a nominal value. Something is definitely wrong, but the operator fails to figure out what is going on in the plant. If a serious failure has occurred in the plant, she must shut the plant down immediately. However, she hesitates to do so, because it is only a single parameter that is drifting, and thus there is possibility of failure at a sensor or simply at an indicator. The operator must not cause an unnecessary shutdown, but must not be late in taking a necessary countermeasure action to avoid a disaster. In order to identify the cause of the strange phenomenon, the operator may iterate stages of information acquisition, information analysis, and situation diagnosis (or, hypothesis generation) repeatedly until she reaches a final situation-diagnostic decision. Only after that, she will be finally ready to come to the action implementation stage.

If our aim is to give a model that describes how human process information to make a decision, then a non-serial model, such as the cyclical model by Neisser (1976), might be the one that we have to seek for. A considerable part of human activity is carried out in anticipation of something that may happen in the future (Hollnagel & Bye, 2000). However, if our aim is to give a device for design decision on human-computer cooperation, the simple four-stage model may be reasonable. We have seen how the simple model can contribute well to distinguish automation for four classes of functions. Moreover, it is worth noting that the discussion on selection of LOA for each class of automation holds in principle for a more complex model of information processing. As long as a *dual-control* like policy will not be taken for situation-diagnosis, automation for information acquisition, information analysis, and decision and action selection do not affect the state of the system even when the stages in Fig.1 appear repeatedly in a non-serial manner before entering the stage of action implementation.

Sharing and Trading

In the discussion of function allocation, it is sometimes useful to distinguish two classes of cooperation between human and computer: Sharing and trading (Sheridan, 1992, 2001).

Sharing refers to the cooperation in which the human and the computer work together at the same time to share the load for a single task. Some types of sharing are possible. In the first type, the computer may help the human so that human's capability may be extended. The power steering and the power braking systems for an automobile are typical examples. It is also possible for human to extend computer's capability. Supervisory override for some types of aircraft is such an example, in which the pilot can add control force when he or she judges the control by the autopilot is insufficient.

In the second type, the computer may help human so that human's burden may be relieved. An example of the second type of sharing is seen in a lane-keeping system developed for an advanced automobile. The lane-keeping system has been designed to reduce driver's workload. It 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). In the examples of the first and the second type of sharing, the human and the computer cooperate in controlling the system on the same degrees of freedom. In the following type of sharing, however, the human may deal with some degrees of freedom, and the computer the remaining degrees of freedom.

The third type of sharing is partitioning, in which a given task is divided into portions so that human and computer can only be responsible for mutually complementary parts. We have seen such an example already, in which the human pilot controls the lateral flight path, and the computer the vertical flight path. When driving a car, human may want to be responsible only for steering by letting the computer control the velocity, which is also partitioning. Sharing by partitioning is one form of complementary cooperation between human and computer.

Trading refers to the cooperation in which either one of human and computer is responsible for a task, and an active agent changes alternately from time to time. Suppose we are driving a car equipped with an adaptive cruise control (ACC) system. If we want to keep certain distance to a car running ahead, we may let the computer do it. When we notice that a car running on our right side is trying to cut in, we may disengage the ACC to seize control back from it to slow down manually before the ACC may make rather a steep deceleration. In cases of aircraft, the pilot manages the flight path in a very early stage of the flight. He or she hands the control over to the computer shortly after the takeoff, and may seize it back to him or her occasionally later during the flight.

In order to implement trading control, it is necessary to decide, when the control must be handed over and to which agent. At the same time, it is also important who makes the decision. In the above two examples, the decision was made by human, and trading of control was initiated by human. However, it is not always human who is given right to do so. The computer may step in to the decision and trading control implementation on a temporary basis. This kind of autonomous behavior of the computer can be seen in a system that are used in a highly risky environment under high time stress. One of such examples is the automatic ground collision avoidance (auto GCAS) system for combat aircraft. When a collision against the terrain was anticipated, the computer gives a "pull-up" warning. If the pilot took a collision avoidance maneuver aggressively, then the computer will not step in any further. If the pilot did not respond to the warning, the computer takes control back from the human pilot and executes an automatic collision avoidance action. It is not easy to judge, as a design decision, whether the computer may be given authority to decide when to seize control back from human and to implement an associated action. This issue will be discussed later.

Sharing and trading of control are essential notions in dynamic function allocation. The notions are inherently indispensable for the adaptive automation discussed in the next section.

Adaptive Automation

Definition

Suppose human and computer are requested to perform assigned functions for some period of time. Operating environment surrounding the system may change as time passes by, or performance of the human may degrade gradually due to psychological or physiological reasons. If the total performance or safety of the system is to be maintained strictly, it may be wise to re-allocate 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*, and automation that operates under an adaptive function allocation scheme is called *adaptive automation*. The term *adaptive aiding* is used in some literatures; see, e.g., (Rouse, 1988). We treat in this chapter the terms adaptive aiding and adaptive automation as synonyms.

An adaptive function allocation scheme assumes criteria to determine whether functions need be reallocated, how, and when. The criteria reflect various factors, such as, changes in environmental factors, task loads or demands to human, performance of human. Adaptive function allocation is inherently dynamic in nature. Note that, however, the concept of adaptive function allocation is not exactly the same as that of dynamic function allocation. Suppose no criteria were violated one day during the operation of a system. No functions were re-allocated, and thus the function allocation might look completely static. Suppose that one of criteria was violated several times on the following day. Functions were reallocated every time the criterion was violated, and thus the allocation scheme would look very dynamic. If we had applied different threshold values to the criteria, function reallocation might have been done in a completely different manner. Thus, dynamic characteristics of adaptive function allocation depend on the situations encountered. On the other hand, dynamic function allocation is not always adaptive. Suppose human and computer exchange responsibilities for functions randomly at random time points. Such a (rather odd) function allocation scheme is dynamic, but is definitely not adaptive in any sense.

The notion of adaptive allocation is not quite new. We can trace its origin back to 1970's. Rouse (1988) states, "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." 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). During the research project, the investigators found situations in which 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). At that stage, they found inadequacies, as we did in the previous section, in making design decisions on function allocation based on the 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, 1976, 1977).

Implementing Adaptive Automation

In adaptive automation, functions can be shared or traded between human and machine flexibly and dynamically in response to changes in situations or human performance. How can such sharing or trading capability implemented? Currently available logic for adaptive allocation or automation invocation can be classified into three major classes: (1) Critical-event logic, (2) Measurement-based logic, and (3) Modeling-based logic.

Critical-event logic. Automation invocation methods of this class change function allocations when pre-defined specific events (called *critical-events*) occur in the human-machine system. An implicit assumption made here is that human workload may become unacceptably high when critical-events happen. Allocation of functions would not be altered if the critical-events did not occur during system operation. In this sense, function allocation with critical-event logic is not only dynamic but also adaptive.

Three types of critical-event logic are distinguished (Barnes & Grossman, 1985; Parasuraman, Bahri, Deaton, Morrison, & Barnes, 1992): (1) *Emergency logic*, in which a control process is executed without human initiation or intervention. (2) *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 input. (3) *Automated display logic*, in which all non-critical display findings are automated to prepare for a particular event, so that the human can concentrate on the most important tasks. Levels of automation (LOA) differ among the three types of critical-event logic. The LOA for the emergency logic is positioned at level 7 or higher, which means that human may not be maintained as the final authority (see, Table 2). In case of executive logic, the LOA is positioned at level 5, in which function allocation is never altered if human disapproves the computer's suggestion. The automated display logic assumes sharing (or, partitioning) of tasks. The computer distinguishes "non-critical" portions of the tasks from the "the most important" ones, and allocates the former to machine so that the workload of the human 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 non-critical or the most important, and human is not usually involved in the judgment. Adoption of high level of automation, such as level 7 or above, can be beneficial for reducing task-load of the human or for buying time. However, it may bring some costs, such as degradation of situation awareness, automation-induced surprises. The issue will be discussed in a later section.

It is possible to design adaptive automation with critical-event logic so that it may have several operational modes with different LOA. For instance, the AEGIS system has a small rule base that determines how the AEGIS system will operate in a combat environment. The following three modes of operation are available (Parasuraman, *et al.*, 1992): (1) Manual, in which the system is fully controlled by the operator: (2) Automatic special, in which a weapon-delivery process is automatic, but the fire button needs to be pressed by the operator: (3) Fully automated, in which the ship 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.

The critical-event logic takes into account human workload in an implicit way. The measurement-based logic, described in the next section, reflects the workload aspect in an explicit manner.

Measurement-based logic. Automation invocation logic of this class emerged at the very early stage of adaptive automation research. Rouse (1977) proposed an idea to allocate tasks dynamically between human and machines so that moment-to-moment workload of the human may be regulated around some optimal level. Workload levels of operators in complex systems fluctuate from moment to moment and at different mission phases. Operators may be able to achieve very high performance levels but only at the cost of high mental workload by neglecting "less critical" tasks. If the situation that requires high level of workload lasts long, performance degradation may result. Performance may also deteriorate when the addition of other minor tasks is made. 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 to all individuals. In fact, different operators will use different strategies to cope with the demands of multiple tasks under time pressure. Thus it is 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 some other operators may be operating at peak workload, which means 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 the workload measurement technology that has high sensitivity and diagnosticity. The sensitivity refers to the capability to detect changes in workload levels, and the diagnosticity the ability to identify what component of mental workload is under or over-stressed.

An example of adaptive automation with a measurement-based logic is proposed by Hancock and Chignell (1988). The adaptive automation works as follows: First the task is defined, structured and subtasks 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, an adaptive policy is implemented. In other words, the criterion remains dynamic and changes every time there is an alteration in the operator's performance and /or task complexity. The adaptive system trades task components in order to improve future measurement of the criterion. According to the workload-based measurement method, adaptivity can be achieved through three main procedures: by adjusting the allocation of subtasks between human and automation; by adjusting the structure of the task; and by refining the task.

Psychophysiological measures, such as papillary dilation, heart rate, can be used in adjusting allocation of functions. Psychophysiological measures offer two main advantages over the other measures. First, psychophysiological measures can be obtained continuously. In many systems where the operator is placed in a supervisory role, very few overt responses may be made even though the operator is engaged in considerable cognitive activity. In such a situation the behavioral measure provides an impoverished sample of the mental activity of the operator. Psychophysiological measures, on the other hand, may be recorded continuously without respect to overt responses and may provide a measure of the covert activities of the human operator. Second, in some instances, psychophysiological measures may provide more information when coupled with behavioral measures than behavioral measures alone. For example, changes in reaction time may reflect contributions of both central processing and response-related processing to workload.

A psychophysiologically based adaptive system would work as follows: It is presumed that pre-existing *profiles* can be established for each operator indicating the correspondences between a specific operator state (such as reduced vigilance, increased workload, etc) and the measured physiological signals. A psychophysiological adaptive system would assess these states "on-line," feeding this information to a secondary logic system (e.g., an expert system) that would determine whether adaptive changes are required.

Modeling-based logic. Operator performance models may be used to decide when function allocation must be adjusted. If a good model is available, it is possible to estimate current and predicted operator state and to infer whether workload is excessive or not. Operator performance models are often categorized into three groups: Intent inferencing models, optimal (or, mathematical) models, and resource models.

If operator's intention could be inferred, it would be possible to identify what tasks must be done for achieving the operator's goal. Information on resources available to the operator would also make it possible to decide whether machine intelligence must intervene to support the operator, and when. *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; Chu & Rouse, 1979), pattern recognition (Revesman & Greenstein, 1986), regression (Morris, Rouse, & Ward, 1986). For example, Walden & Rouse (1978) investigated multitask performance of a pilot, where time sharing is required between monitoring, control, and other tasks, where they modeled the monitoring task as a queuing system which can be modeled as a "single server" with subsystem events called "customers" and with the control task incorporated as a special queue. The adaptive system with optimal models works as follows: Once the customers arrive at the control task queue they can control the service of a subsystem event. From what proceeded, 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, such as multiple resource theory by Wickens (1984) try to estimate the operator's current and projected resource utilization. The models describe how performance interference occurs in information processing. Suppose an operator are trying to perform two different tasks, in which the task compete for the same human information processing resource. If the two tasks require different resources, say verbal and spatial codes, then there will be no difficulty in 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 model is used to evaluate whether function allocation is appropriate or not, or to assess the impact of possible competition that may be caused by tasks requiring the same resource. With the similar reason, the multiple resource theory has been recognized as an important tool for human-machine system design. For example, the multiple resource theory has been implemented in a discrete event simulation-modelling tool, WinCrew (a computer software) that has the capability to change function allocation dynamically based on the moment-to-moment workload values (Archer & Lockett, 1997; Laughery, 1999). Application of WinCrew to one of realistic problems can be found in (Archer, Lewis, & Lockett, 1997), where activities of bridge personnel on a Navy Guided Missile Destroyer were described as a task network, and moment to moment workload values were assessed for four different manning, automation, automation, and task allocation configurations under several scenarios.

We have discussed so far in this section how the performance models may be utilized. If performance models are good, it is sometimes possible to extract "leading indicators." The leading indicators refer to precursor the observation of which implies the occurrence of some subsequent events. Kaber & Riley (1999) have demonstrated the benefits of adaptive aiding on a primary task (a dynamic cognitive monitoring and control task) that is based on degradation of an automation-monitored secondary task, in which degradation of the secondary task is a precursor of that of the primary task.

Comparison of three types of logic. Among three types of logic, the critical-event logic may be the easiest and most straightforward to implement, if there is a consensus that a specific event is actually a "critical-event." No investigations are needed regarding how human cognition or behavior could be modeled, what parameters must be measured to infer human state, and how. Once a critical-event is defined, occurrence of the event can be detected via objective information that is usually available even in a conventional human machine system. A possible disadvantage of the critical-event logic is that it may reflect human workload or performance viewpoints only partially. Though the critical-event is regarded as an event that requires high human workload, the recognition may not always be true for everybody at all circumstances.

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 reflecting explicitly the mental status of an operator at a specific circumstance. There is no need to predict in advance how mental status of the operator may change. However, there are some points to be noted. One is that not all operators may welcome situations in which they are monitored by sensing devices all the time. The second point is that sensing devices are sometimes expensive and/or are too sensitive to local fluctuations in operator workload or physiological states.

If good performance models are available, they can contribute well to successful design of interaction between human and automation. Comparing with the measurement-based logic, the model-based automation invocation logic is easier to implement, because no measurement technologies or devices will be needed. However, as can be easily imagined, it is not possible in general to develop a performance model representing the reality perfectly. For instance, a mathematical model needs assumptions to make the model mathematically tractable. In case of modeling in terms of production rules, it is not obvious whether such a model can be constructed for human behavior in a complicated task environment.

Who has decision authority?

We have seen, in Section 4.2, three types of automation invocation methods for changing function allocation dynamically in a situation-adaptive manner. Who is supposed to make decisions when function allocation must be altered? Human operator, or machine intelligence? Each method in Section 4.2 can be expressed with production rules: For instance, "If critical-event A is detected, then function B must be handed over to the automation, if the function was dealt with by the human at that time moment." Or, "If the estimated human workload becomes lower than a specified threshold value, then function C must be returned from the automation to the human." Once an automation invocation scheme is written as a production rule, it is basically true that the scheme can be fully automated, and thus machine intelligence (or, the computer) can manage adaptive function allocation without any aid of the human operator. However, the reality is not so simple. There are some reasons for why the computer is not given authority to decide when to change function allocation.

One apparent reason is reliability. It is unrealistic to assume that the computer never fails. The failure may be due to hardware malfunction, or may be caused by some errors or "bugs" in the program, or may stem from input data with undesired characteristics (such as, lack of precision, corruption with noise, lack of the very data). Even with such possibilities, if the computer is assigned a task to make an automation invocation decision, the human will have to monitor the computer carefully all the time, which produces burden on the human in addition to his or her original tasks.

A second reason is philosophy in human-machine systems. From a viewpoint of human-centered automation (Woods, 1987; Billings, 1991,1997; Billings & Woods, 1994), human operator must be maintained as the final authority, and only he or she may exercise decisions how function allocation must be changed and when. In aviation, for instance, pilots decide when to use various automation, such as the autopilot, the auto-throttle. However, is it always the best for the human operator to bear the final decision authority at all times and in 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 or she would probably be able to perform the task with little or no aiding." 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). As a matter of fact, Hancock & Scallen (1996) investigates an invocation scheme that machine intelligence may have authority for automation invocation; "A human operator would perform a continuous control task until some performance criteria was violated as might be expected after an extended period of performance. After criterion violation the machine would assume control of the task and return control to the operator after a rest period. If the performance criteria were never violated, automation would never assume control."

As has been mentioned earlier, a better solution to an optimization problem may be found if the constraints for the problem could be made looser. Therefore, theoretically speaking, an ideal configuration of adaptive automation is the one in which either human or computer may make a decision on automation invocation. If we assume either that only human may decide when to change allocation and how, or that only computer may do so, then the resulting automation invocation scheme may be too stiff and fail to attain an optimal performance.

There are experimental and mathematical evidences that show that whether human-initiated invocation of automation is beneficial or not heavily depends on the situation encountered. Harris, Hancock, Arthur, and Card (1991) have conducted an experiment in which subjects are given multiple tasks (a resource management task, a system monitoring task, and a compensatory manual tracking task). The investigators compared the following three conditions: (1) The subjects must perform all tasks manually; (2) The tracking task is performed by the automation; (3) The subjects can decide whether to invoke the automation for tracking task. The investigators have obtained the result that the subjects were more efficient at the resource management task in the third condition in which automation invocation is human-initiated. Hilburn, Molloy, Wong, and Parasuraman (1993) have obtained a similar result through a series of experiment under the multitask environment as in the case of Harris, *et al.* (1991). In one of the experiment, the investigators have compared executive and emergency logic in the critical-event logic category, and have found a slight automation cost under either automation invocation logic.

On the other hand, Harris, Goernert, Hancock, and Arthur (1994) have found a condition in which computer-initiated invocation of automation is beneficial. The investigators conducted an experiment with the multiple task environments as described in the above. Performances of the tasks are compared between operator-initiated and computer-initiated invocation of automation. The investigators have found the following insights: (1) When subjects received written warnings that workload increases were likely to occur, performance during human- and computer-initiated invocation of automation did not differ; (2) When subjects were not warned before workload increases, resource management error was greater during periods of human-initiated invocation of automation. The observations suggest that human-initiated invocation of automation may be less beneficial than computer-initiated invocation when changes in workload may be abrupt or unexpected for the human operator. Harris, Hancock, and Arthur (1993) have also shown that, when subjects became fatigued under multiple-task environment, they became less likely to engage automation even when it is supposed to be used. That means benefits of automation may not be fully appreciated if human-initiated invocation of automation is adopted.

There is also a mathematical argument proving that which agent (human or computer) must be in charge of deciding invocation of automation cannot be fixed but strongly depends on the situation. Inagaki (1997, 2000a) has discusses a rejected takeoff (RTO) problem when an engine failure may be suspected. A mathematical analysis with a probabilistic model has proven that Go/NoGo decision should neither be fully automated nor be left always to a human; i.e., the decision authority of automation invocation must be traded dynamically between human and computer. More concretely, (1) Human pilot must be in authority when the aircraft speed is far below V1 (the takeoff decision speed); (2) Computer must be in authority, if the aircraft is almost at V1 and if there is possibility that human pilot may hesitate to make decisions when he or she failed to decide whether the engine became faulty or not; (3) When the aircraft speed locates between (1) and (2), which agent must be in authority depends on the situation.

Benefits and costs of adaptive automation

One of major motivations for introducing adaptive automation was 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, Molloy, Wong, & Parasuraman, 1993). Another major benefit of adaptive automation lies in its ability to keep the operator in the control loop, which is done by altering levels of automation (LOA). The characteristic contrasts with the static allocation. When the LOA for a task is always positioned at high levels, the operator is likely to suffer from the out of the control loop phenomena that lead to degradation of manual skill, vigilance decrements, and loss of situation awareness for the task (Wiener, 1988; Parasuraman, Bahri, Deaton, Morris, & Barnes, 1992; Gluckman, Carmody, Morrison, Hitchcock, & Warm, 1993; Endsley & Kiris, 1995; Endsley & Kaber, 1997; Kaber, Omal, & Endsley, 1999). For instance, Hilburn, Molloy, Wong, & Parasuraman (1993) conducted a series of experiment under multiple task environment with compensatory tracking, system monitoring, and fuel management tasks, and have found the phenomenon in which humans exhibited inefficient monitoring performance after only 20 minutes of automated monitoring control. When automation or the system is perceived as being "highly reliable," automation-induced "complacency" may arise (Parasuraman, Molloy, & Singh, 1993), where the word complacency has been introduced to mean, "self-satisfaction which may result in non-vigilance based on an unjustified assumption of satisfactory system state" (Moray and Inagaki, 2000).

In case of adaptive automation, even if it may adopt high LOA at one time point, the LOA will be altered at other time point, which may be useful to avoid the out of the control loop phenomena. However, what happens if LOA is altered too frequently? We have seen in section 4.2 some types of logic that alter LOA. If the algorithm were highly sensitive, 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 have investigated the effects of short cycling on task performance. By using a Multi-Attribute Task battery (Comstock & Arnegard, 1990) that includes tracking, monitoring, fuel task management, and ATC communications, Parasuraman, Bhari, Molloy, & Singh (1991) have demonstrated both benefits and costs of short-cycle automation on the manual performance of tasks and on the monitoring for automation failure. Glen, Barba, Wherry, Morrison, & Hitchcock (1994) have investigated adaptive automation effects on flight management task performance under multiple task environment with compensatory tracking, tactical assessment, and communication tasks. The investigators have demonstrated no automation deficits, and have found automation benefits for reaction time in the tactical assessment task. Scallen, Hancock, & Duley (1995) have investigated the rapid cycling of automation under multiple task environment with tracking, fuel management, and system monitoring tasks, where tracking task cycled between manual and automated control at fixed intervals of either 15, 30, or 60 seconds. The investigators have found that excessively short cycles of automation prove disruptive to performance in multi-task conditions.

Another major concern in adaptive automation would be possible conflict between the human and the computer, where conflict can be classified into two groups. The conflict of the first type refers to the case in which human and computer share the same goal but do not share courses of action to achieve the goal. The conflict of this kind occurs when human fails to maintain appropriate mode awareness. An example can be seen in the crash of an Airbus 320 aircraft at Strasbourg, France, in 1991. The pilots thought that the aircraft was making an approach using a flight path angle mode of –3.3 degrees. However the computer, that was an active agent at that time moment, was actually making an approach by using a vertical speed mode of –3,300 ft/min (Sparaco, 1994; Billings, 1997). There are many incidents and accidents due to mode confusion (see, e.g., Dornheim, 1995; Hughes, 1995a; Sarter & Woods, 1995).

The conflict of the second type refers to the case in which the human and the computer do not share the same goal. An example can be seen in the crash of an Airbus 300-600R aircraft at Nagoya, Japan, occurred in 1994. At some point during the final approach, the pilot flying 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 intentions between the pilot and the computer became completely contradictory. Once the computer was ordered by the human to go around, it tried to achieve the go-around at any cost. For the computer, the human's input to descend was simply a *disturbance* that must be cancelled out by applying a stronger control input. 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.

One way to avoid conflict is communication between the human and the computer. Revesman & Greenstein (1986) distinguish two types of communication: (1) dialogue-based communication and (2) model-based communication. In the dialogue-based communication, the human provides the computer with information regarding his or her action plans. The dialogue-based communication is advantageous in attaining precise understanding and high level of situation awareness. However, the dialogue may increase human workload, or may "detract from the human's ability to perform the primary task" (Revesman & Greenstein, 1986). The model-based communication assumes no explicit dialogue between the human and the computer. In the model-based communication, "the computer employs a model of human performance to predict the actions of the human. Using these predictions, the computer selects its own actions so as to minimize some measure of overall system cost" (Revesman & Greenstein, 1986). The investigators have proposed a model and have proven validity of the model-based communication through a simulation study by showing that "the model is a consistent and accurate predictor of human performance, correctly predicting over 80 percent of the subjects' actions over a range of situations" (Greenstein & Revesman, 1986).

Interface design may contribute in reducing conflict between human and computer. While LOA is positioned at a high level, human may often fail to recognize what the computer's intention is. If the computer's intention or its goal is explicitly shown on a display, possibility of conflict may be reduced. Suppose the human did not like the computer's goal or intention. Only he or she has to do is give a new directive to the computer or disengage the automation to seize back the control to him or her. An example of interface design that displays computer's intention by means-ends relation can be found in Furukawa & Inagaki (2001). Appropriate interface design to support situation awareness or recognition is important, especially when it may not be possible to assume that every operator has been trained substantially.

Trust in adaptive automation

It has been observed that a dynamic function allocation suffers from various types of automation surprises (Wickens, 1994; Hughes, 1995b; Sarter, Woods, & Billings, 1997). Since adaptive function allocation adopts a sophisticated algorithm for triggering automation, it may have to face with difficulties that are severer than a standard dynamic function allocation. Lee & Moray (1992) distinguishes four dimensions of *trust*: (1) *foundation* that represents the "fundamental assumption of natural and social order that makes the other levels of trust possible," (2) *performance* that rests on the "expectation of consistent, stable, and desirable performance or behavior," (3) *process* that depends on "an understanding of the underlying qualities or characteristics that govern behavior," and (4) *purpose* that rests on the "underlying motives or intents."

It would not be hard for a human operator to recognize that the designer's purpose or intention in creating adaptive automation lies in regulating operator workload at some optimal level. Respecting the second and the third dimensions may not be straightforward. Since adaptive automation is designed to behave in a situation-dependent manner, its behavior may look inconsistent. Suppose there are two circumstances, A and A*, that differ only slightly. The automation invocation algorithm may detect the difference of A and A*. Then it will behave differently. However, the human may fail to recognize the difference. Then the human would be surprised when he or she saw the automation behaved differently. Human's understanding of the automation invocation algorithm may be imperfect if the algorithm is sophisticated or complicated. When the human failed to be certain of the second and the third dimension of trust, he or she would fail to establish trust in the adaptive automation. Human's distrust or mistrust in automation can cause inappropriate use of automation, as has been pointed out by Parasuraman and Riley (1996).

Concluding Remarks

This chapter has described necessity, efficacy, and significance of the design framework in which functions are allocated dynamically and adaptively between human and machine (or, computer). Adaptive function allocation offers wider and more flexible design decisions on human-automation cooperation than static function allocation does. Therefore, from an optimization theoretic viewpoint, a design decision obtained in the framework of adaptive function allocation can never be worse than that obtained in the framework of static function allocation. However, reality is not so simple. Various difficulties and inconveniences may arise in adaptive automation, which stem from highly dynamic and flexible characteristics of adaptive automation. It is not sensible to claim that adaptive automation is always more effective and useful than any other form of conventional automation in any application area. Adaptive automation determines its behavior by reflecting various factors, such as human workload, human performance, state of the controlled process, time-criticality, risk associated with action or non-action in dynamically changing environment. There are enormous numbers of possible combinations of states or values that the above factors can take. An important point to note is that adaptive automation inherently requires *custom tailored* design decisions. Through careful examination of the possibilities, we can decide which function allocation scheme must be adopted, static, standard dynamic, or adaptive.

Although various concerns have been pointed out on adaptive automation, significance and efficacy of adaptive automation have also been proven theoretically or experimentally. Human-centered automation usually assumes that the human must be at the locus of control (Woods, 1989; Billings, 1991, 1997), while some type of adaptive automation may violate the principle. However, it is proven that such adaptive automation can play an essential role in attaining system safety under time-critical situations (Inagaki, 2000a, 2000b). Research of adaptive automation has its origin in aviation and various studies have been conducted in the context of aviation applications. Rouse (1988), Parasuraman *et al.* (1992), Scerbo (1996), Wickens & Hollands (2000), and Scallen & Hancock (2001) give good survey of those efforts. However, real systems with adaptive automation concept are now being implemented or investigated also in other fields, including military (Bonner, Taylor, & Fletcher, 2000; Corbridge, Simpson, Cook, & Turpin, 2000), passenger cars in the Advanced Safety Vehicle (ASV) project in Japan. Although it is sometimes said that adaptive automation is still in its infancy, the concept is applied explicitly or implicitly in the design of various human-machine systems.

References

Archer, R.D., Lockett, J.F. (1997). WinCrew – A tool for analyzing performance, mental workload and function allocation among operators. *Proc. 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. *Proc. 40th Annual Meeting of HFES.*

Barnes, M. & Grossman, J. (1985). *The Intelligent Assistant Concept for Electronic Warfare Systems.* NWC TP 5585.

Billings, C. E. (1991). *Human-Centered Aircraft Automation: A Concept and Guidelines.* NASA TM-103885.

Billings, C.E. & Woods, D.D. (1994). Concerns about adaptive automation in aviation systems. In Mouloua, M. and Parasuraman, R. (Eds.), *Human Performance in Automated Systems: Current Research and Trends.* Erlbaum, 264-269.

Billings, C. E. (1997). *Aviation Automation – The Search for a Human-Centered Approach.* LEA.

Bresley, B. & Egilsrud, J. (1997). Enhanced Ground Proximity Warning System, *Safety Bird,* 33, 12-22.

Chernoff, H. (1973). The use of faces to represent points in k-dimensional space graphically. *Journal of American Statistical Association*, 68, 361-368.

Chu, Y-Y. & Rouse, W.B. (1979). Adaptive allocation of decisionmaking responsibility between human and computer in multitask situations. *IEEE Trans. Systems, Man, and Cybernetics*, 9(12), 769-778.

Comstock, J. & Arnegard, R. (1990). *Multi-Attribute Task Battery.* Technical Memorandum 104174. Hamptom, VA: NASA Langley Research Center.

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

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

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), 3181-394.

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. *Proc. 13th Triennial Congress of the International Ergonomics Association,* 168-170.

Fitts, P.M. (Ed.).(1951). *Human Engineering for an Effective Air-Navigation and Traffic-Control System.* Columbus, OH: Ohio State University Research Foundation.

Furukawa, H. & Inagaki, T. (2001). A graphical interface design for supporting human-machine cooperation: Representing automation's intention by means-ends relations. *Preprints of the 8th IFAC Human-Machine Systems*, 651-656.

Glenn, F., Barba, C., Wherry, R.J., Morrison, J., Hitchcock, E., & Gluckman, J.P. (1994). In Mouloua, M. and Parasuraman, R. (Eds.), *Human Performance in Automated Systems: Current Research and Trends.* Erlbaum, 33-39.

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. *Proc. 7th International Symposium on Aviation Psychology*, 150-155.

Greenstein, J.S., & Revesman, M.E. (1986). Development and validity of a mathematical model of human decisionmaking for human-computer communication. *IEEE Trans. Systems, Man, and Cybernetics*, 16(1), 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.

Hancock, P.A. & Chignell, M.H. (1988). Mental workload dynamics in adaptive interface design. *IEEE Trans. Systems, Man, & Cybernetics*, 18(4), 647-658.

Hancock, P.A. & Scallen, S.F. (1996). The future of function allocation. *Ergonomics in Design*, October 1996, 24-29.

Hancock, P.A. & Scallen, S.F. (1998). Allocating functions in human-machine systems. In R.R. Hoffman, M.F. Sherrick, & J.S. Warm (Eds.). *Viewing Psychology as a Whole.* Americal Psychology Association, 509-539.

Harris, W.C., Hancock, P.A., Arthur, E.J., & Caird, J.K. (1991). Automation influences on performance, workload, and fatigue. *Proc. 35th Annual Meeting of HFES.*

Harris, W.C., Hancock, P.A., & Arthur, E.J. (1993). The effect of taskload projection on automation use, performance, and workload. *Proc. 7th International Symposium on Aviation Psychology*, 178-184.

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: Current Research and Trends*, 40-44.

Hilburn, B., Molloy, R., Wong, D., & Parasuraman, R. (1993). Operator versus computer control of adaptive automation. *Proc. 7th International Symposium on Aviation Psychology*, 161-166.

Hollnagel, E. & Bye, A. (2000). Principles for modeling function allocation. *International Journal of Human-Computer Studies*, 52, 253-265.

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

Hughes, D. (1995b). Studies highlight automation 'surprises'. *Aviation Week and Space Technology*, 48-49, Feb 6.

Inagaki, T. (1991). Interdependence between safety-control policy and multiple-sensor schemes via Dempster-Shafer theory. *IEEE Trans. Reliability*, (40)2, 182-188.

Inagaki, T. & Johannsen, G. (1992). Human-computer interaction and cooperation for supervisory control of large-complex systems. *Proc. EUROCAST '91: Second International Workshop on Computer Aided Systems Theory*, 281-294.

Inagaki, T. (1993). Situation-adaptive degree of automation for system safety. *Proc. 2nd IEEE Robot and Human Communication*, 231-236.

Inagaki, T. (1997). To go no not to go: Decision under time-criticality and situation-adaptive autonomy for takeoff safety. *Proc. IASTED International Conference on Applied Modelling and Simulation*,144-147.

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*, 154-159, Erlbaum.
Inagaki, T., Takae, Y., & Moray, N. (1999). Automation and human interface for takeoff safety. *Proc. 10th International Symposium on Aviation Psychology*, 402-407.
Inagaki, T. (2000a). Situation-adaptive autonomy for time-critical takeoff decisions. *International Journal of Modelling and Simulation*, 20/2, 175-180.
Inagaki, T. (2000b). Situation-adaptive autonomy: Dynamic trading of authority between human and automation. *Proc. IEA2000/HFES2000 Congress*, III.13-16.
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*, 129-133, Erlbaum.
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*. Erlbaum, 165-169.
Kawazoe, H., Murakami, T., Sadano, O., Suda, K., & Ono, H. (2001). Development of a lane-keeping support system. *Proc. Intelligent Vehicle Initiative (IVI) Technology and Navigation Systems*, 29-35.
Klein, G. (1998). *The Source of Power*. MIT Press.
Klein, G., Orasanu, J., Calderwood, R, & Zsambok, C.E. (Eds.) (1993). *Decision Making in Action: Models and Methods*. Ablex.
Laughery, R. (1999). Using discrete-event simulation to model human performance in complex systems, *Proc. 1999 Winter Simulation Conference*, Phoenix Arizona.
Lee, J.D. & Moray, N. (1992). Trust, control strategies and allocation of function in human machine systems. *Ergonomics*, 35(10), 1243-1270.
Lockhart, J.M., Strub, M.H., Hawley, J.K., & Tapia, L.A. (1993). Automation and supervisory control: A perspective on human performance, training, and performance aiding. *Proc. HFES 37th Annual Meeting*, 1211-1215.
Millot, P. & Willaeys, D. (1985). An approach of dynamical allocation of supervision tasks between man and computer in control rooms of automatized production systems. *Proc. IFAC Man-Machine Systems*, 159-165.
Millot, P. , Taborin, V., Kamoun, A, & Willaeys, D. (1988). Effects of the dynamic allocation of supervision tasks between man and computer on the performance of automated processes. In J. Patrick and K.D. Duncan (Eds.). *Training, Human Decision Making and Control*, Elsevier Science Publishers, 175-187.
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*, 134-138, 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.
Moray, N. & Inagaki, T. (2000). Attention and complacency. Theoretical Issues in Ergonomics Science. 1(4). 354-365.
Morris, N.M., Rouse, W.B., Ward, S.L. (1988). Studies of dynamic task allocation in an aerial search environment. *IEEE Trans. Systems, Man, & Cybern.*, 18(3), 376-389.

Neisser, U. (1976). *Cognition and Reality: Principles and Implications of Cognitive Psychology*, Freedman.

Parasuraman, R., Bhari, T., Molloy, R., & Singh, I. (1991). Effects of shifts in the level of automation on operator performance. *Proc. 6th International Symposium on Aviation Psychology*, 102-107.

Parasuraman, R., Bhari, T., Deaton, J.E., Morrison, J.G., & Barnes, M. (1992). *Theory and Design of Adaptive Automation in Aviation Systems.* Progress Report No. NAWCADWAR-92033-60, Naval Air Development Center Aircraft Division.

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 Trans. 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 Trans. 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*, 295-306, Plenum Press.

Rouse, W.B. (1977). Human-computer interaction in multitask situations. *IEEE Trans. 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., 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.

Rouse, W.B. (1991). *Design for Success: A Human Centered Approach to Designing Successful Products and Systems*, 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*, 28-32.

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 Edition, Wiley, 1926-1943.

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. (1997). *Performance and Workload Effects for Full versus Partial Automation in a High Fidelity Multi-Task System.* (Unpublished PhD Thesis). University of Minnesota.

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*, LEA, 37-63.

Sharit, J. (1997). Allocation of functions. In G. Salvendy (Ed.). *Handbook of Human Factors and Ergonomics*, Second Edition, Wiley, 301-339.

Sheridan, T.B. & Verplank, W.L. (1978). *Human and Computer Control of Undersea Teleoperations.* MIT Man-Machine Systems Laboratory, Technical Report.

Sheridan, T. B. (1992). *Telerobotics, Automation, and Human Supervisory Control.* MIT Press.
Sheridan, T.B. (2001). *Humans and Automation: A Tutorial on System Design and Research Issues.* Human Factors and Ergonomics Society & Wiley.
Shortliffe, E.H. (1976). *Computer-Based Medical Consultations: MYCIN.* American Elsevier.
Sparaco, P. (1994). Human factors cited in French A320 crash. *Aviation Week and Space Technology*, 30-31, Jan 3.
Vicente, K.J. & Rasmussen, J. (1991). Ecological interface design: Theoretical foundations. *IEEE Trans. Systems, Man, and Cybernetics*, 22 (4), 589-607.
Walden & Rouse, W.B. (1978). A queueing model of pilot decisionmaking in amultutask flight management situation, *IEEE Trans. Systems, Man, & Cybern*, 8(12), 867-875.
Wei, Z-G. (1997). *Mental Load and Performance at Different Automation Levels.* Delft University of Technology.
Wickens, C.D. (1984). Processing resources in attention. In R. Parasuraman & D.R. Davies (Eds.) *Varieties of Attention*, Academic Press.
Wickens, C.D. (1994). Designing for situation awareness and trust in automation. *Proc. IFAC Integrated Systems Engineering*, 77-82.
Wickens, C.D., Mavor, A., Parasuraman, R., & McGee, J. (1998). *The Future of Air Traffic Control: Human Operators and Automation.* National Academy Press.
Wickens, C.D., Gordon, S.E., & Liu, Y. (1998). *An Introduction to Human Factors Engineering.* Addison-Wesley Educational Publishers.
Wickens, C.D. & Hollands, J.G. (2000). *Engineering Psychology and Human Performance*, Third Edition, Prentice Hall.
Wiener, E.L. (1988). Cockpit automation. In E.L. Wiener & D.C. Nagel (Eds.). *Human Factors in Aviation.* Academic Press.
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 CP-10036, 61-85.
Zsambok, C.E. (1997). Naturalistic decision making: Where are we now? In C.E. Zsambok & G. Klein (Eds.), *Naturalistic Decision Making.* LEA. 3-16.