6. CONCLUSIONS AND TRANSITION

This chapter has addressed a critical limitation of human performance when high levels of stress are imposed, with particular emphasis on the stress of high task demand or mental workload. We have also shown how the response to stress varies between people, in terms of skill automaticity, time-sharing skills, coping strategies, and genetically based working memory and executive function capabilities. In particular we have emphasized how this response is manifest in various aspects of brain function, consistent with the neuroergonomics approach.

As such, we provide two vital links to the next chapter. First, designers of automation to replace or augment human performance are driven heavily (although not exclusively) by the desire to reduce or "offload" human operator workload. Second, because such needs are not constant, but vary from person to person and occasion to occasion, such automation can be applied *adaptively* rather than in the same, static way. It is in this domain that we find that the neuroergonomic methods discussed in this chapter provide some of the most important signals for when to adapt automation to the specific needs of individual human operators.

Key Terms

absolute workload *349*
adaptive automation *352*
arousal *361*
attentional narrowing *364*
automaticity *349*
brain computer interfaces *374*
cognitive appraisal *363*
confirmation bias *366*
data link *356*
embedded secondary tasks *351*
entropy *357*
mental workload *346*
multitasking *348*
neuroergonomics *346*
offline measures *352*
online measures *352*
predictive models *350*
relative predictions *350*
relative workload *349*
strategic control *366*
stress inoculation *370*
time-sharing *371*
transactional appraisal *363*
workload assessment *350*

12 AUTOMATION AND HUMAN PERFORMANCE

1. INTRODUCTION

Since their invention, computers have become smaller, faster, more powerful, cheaper, and—to a degree—more "intelligent." These changes have come about at an exponential rather than a linear rate—an acceleration known as "Moore's Law" (Moore, 1965)—and have fuelled the widespread introduction of computer-based automation, which, from small beginnings in the 1960's, has encroached on all parts of life today. Automated systems are found in all aspects of work—in manufacturing, power generation, health care, transportation, offices, homes, and in many other industries. The growth has been so pervasive that automation is here to stay. Think of life without GPS, Internet search engines, and electronic commerce. In the near future, miniature automated devices may permeate our clothes and perhaps even our bodies. The extent to which automation has pervaded both the workplace and everyday life is well captured by a massive volume on automation published by Nof (2009), which required more than 90 chapters to describe this widespread application!

Many factors are responsible for the widespread implementation of automation, which shows little sign of abating. The factors include economic issues, in particular reducing labor costs, increasing efficiency, improving safety requirements, and remaining competitive in the marketplace (Satchell, 1998). Have such outcomes been realized?—To a large degree, yes.

Automation has yielded many benefits. Consider two domains where automation is common: health care and aviation. In the former, electronic medical records and decision support systems have contributed to a reduction in adverse patient outcomes (Gawande & Bates, 2000; Morrow, Wickens, & North, 2006). Automatic clinical reminders that guide the physician's attention to health issues for a particular patient and recommend follow up have also improved patient care (Karsh, 2010; Vashitz et al., 2009). In surgery, "image-guided navigation" that supports the surgeon during mastectomy operations can improve patient safety (Manzey et al., 2011). In aviation, automation has allowed aircraft to fly more direct routes, thereby reducing fuel costs. The safety record of more automated commercial airplanes also continues to improve on that of earlier generations of aircraft (Billings, 1997; Pritchett, 2009; Wiener, 1988). Similar benefits have been documented in many other domains where automation has been implemented—at work, in transportation, in leisure activities, and at home (Nof, 2009; Sheridan & Parasuraman, 2006).

A principal benefit of automation, irrespective of the area of application, is that it can, if carefully designed, reduce the human user's workload, both mental and physical. Such workload reductions can occur in response execution and muscular exertion (consider the automated can opener, screw driver, or pencil sharpener), in decision choice (remember, as discussed in Chapter 8, the mental effort involved in making high-risk decisions in unfamiliar domains), and in information acquisition and analysis (recall the cost of scanning a cluttered display,

or mentally adding two numbers). More than anything else, the potential for automation to reduce workload is what makes it attractive to the human operator in environments in which time stress is high or in work settings where cognitive effort has to be minimized because of the need to carry out many other concurrent tasks. Yet, as we shall see later in this chapter, this workload-reducing feature can at the same time invite new types of problems when automation is introduced.

Given the widespread benefits that automation has provided, it is not surprising that designers have pushed for greater and more powerful automation when they are charged with developing new systems. This is often done in the belief that human error will be eliminated, or that excessive levels of operator workload will be reduced, so that opportunities for human error will decrease. However, such beliefs have turned out to be fallacious. While automation may reduce some forms of error, it can introduce new ones (Pritchett, 2009; Sarter, 2008), and in some cases automation may paradoxically *increase* rather than decrease human mental workload (Wiener & Curry, 1980). Research on human-automation interaction has shown that automation changes the nature of the cognitive tasks that humans have to do, often in ways that were unexpected or unanticipated by designers (Parasuraman & Riley, 1997). Consequently and ironically, as automation becomes more powerful and assumes more authority, the human role actually becomes *more* rather than less important (Parasuraman & Wickens, 2008).

A technology-centered approach to design has been largely responsible for the human performance issues that have arisen with automated systems. Designers have typically concentrated their energies on the sensors, algorithms, and actuators that go into automated systems, with little or no attention given to the characteristics of the human users of such systems. There is now ample evidence to support the view that rather than focusing simply on the technical features of the automation, designers should also consider human performance, an approach sometimes called **human-centered automation** (Billings, 1997). The challenge, therefore, is to design for *joint* human-automation performance (Lee & Seppelt, 2009).

In this chapter we discuss how that challenge can be met. We consider different aspects of human capabilities and limitations that are brought out when humans interact with automation and which have been extensively described in previous chapters of this book. Because automation can be applied to the entire range of human functioning, from sensing through decision making to action, many of the components of the information-processing model that was introduced in Chapter 1 are relevant to an understanding of human-automation interaction. We begin our examination of issues in human-automation interaction by first discussing examples and purposes of automation.

2. EXAMPLES AND PURPOSES OF AUTOMATION

Automation can be defined as the performance by machines (typically computers) of functions that were previously carried out, whether fully or partially, by humans (Parasuraman & Riley, 1997). In some cases, the term automation has also been applied to describe those tasks that humans are incapable of performing (e.g., sensing beyond the visible or audible spectrum, or robots lifting heavy loads or handling toxic material). Automation may be described in terms of its purposes, the human performance functions it replaces, and the strengths and weaknesses it shows as humans interact with automated devices ranging from simple alarm systems to complex autopilots and decision-aiding systems. The different purposes of automation may be assigned to five general categories.

2.1 Tasks That Humans Cannot Perform

Automation is sometimes necessary because it can carry out functions that the human operator cannot perform. This category describes many of the complex mathematical operations performed by computers (e.g., those involved in statistical analysis). In the realm of dynamic systems, examples include control guidance in a manned booster rocket, in which the time delay of a human operator would cause instability (see Chapter 10); aspects of control in complex nuclear reactions, in which the dynamic processes are too complex for the human operator to respond to online; or robots that operate in hazardous confined spaces, such as their use in searching for victims in the collapsed World Trade Center following the September 11 terrorist attack (Casper & Murphy, 2003). In these and similar circumstances, automation appears to be essential and unavoidable, whatever its costs.

2.2 Human Performance Limitations

This category of automation includes functions that the human operator can do but only poorly or at the cost of high workload because of system complexity and information load. Examples include the autopilots that control many aspects of flight on commercial aircraft (Degani, 2004; Pritchett, 2009; Sarter & Woods, 1995; Sebok et al., 2012), and the automation of certain complex monitoring functions, such as the ground proximity warning system (GPWS), alerting pilots to the possibility of collision with the terrain, or alerts for possible collisions with other aircraft (Wickens, Rice, et al., 2009). Efforts have also been directed toward automating diagnosis and decision processes in such areas as medicine (Garg et al., 2005; Morrow et al., 2006), nuclear process control (Woods & Roth, 1988), ship navigation (Lee & Sanquist, 2000), and coordination of multiple unmanned aerial and ground vehicles (Barnes & Jentsch, 2010; Cummings et al., 2007; Parasuraman et al., 2007). Military command and control operations are also increasingly being carried out in a network-centric manner, where many entities are connected together in large, complex, distributed networks, further mandating the use of automated agents (Cummings et al., 2010). These approaches generally require the implementation of artificial intelligence, in the form of expert systems (Darlington, 2000) or agent-based software (Lewis, 1998).

2.3 Augmenting or Assisting Human Performance

Automation can assist humans in areas where they exhibit limitations. This category is similar to the preceding one, but automation is intended not as a replacement for integral aspects of the task but as an aid to peripheral tasks or mental operations necessary to accomplish the main task. As we have seen in previous chapters, there are major bottlenecks in human performance, in particular, limitations in human working memory and in prediction or anticipation for which automation would be useful. An automated display or visual echo of auditory messages is one such example, as discussed in Chapters 6 and 7. Examples of this might be the phone number retrieved from operator information which appears on a small telephone display; or digitized data link instructions from air traffic control "uplinked" to the aircraft that can appear as a text message on the pilot's console (Helleberg & Wickens, 2003).

Another example is a computer-displayed "scratch pad" of the output of diagnostic tests in fault diagnosis of the chemical, nuclear, or process control industries. As suggested in Chapter 8, this procedure would greatly reduce memory load. As noted several times throughout this book, any sort of predictive display that would off-load the human's cognitive burden of making predictions would be of great use. Yet another example of an automated aid is the display "decluttering" option, which can remove unnecessary detail from an electronic display when it is not needed,

thereby facilitating the process of focused and selective attention (St. John et al., 2005; Yeh & Wickens, 2001).

2.4 Economics

Automation is often introduced because it is less expensive than paying people to do equivalent jobs or to be trained for those jobs. Thus, we see robots replacing workers in many manufacturing plants and automated phone menus replacing the human voice on the other end of the line. Unmanned air vehicles are far cheaper to both manufacture and fly than are manned airplanes (Cooke et al., 2006). But as the phone menu example suggests, the economy achieved by such automation does not necessarily make the service "user friendly" to the human who must interact with it (Landauer, 1995; St. Amant et al., 2004).

2.5 Productivity

There are many instances in which increased demands for productivity are imposed when there is limited manpower. For example, increased demands for air travel put more planes in the sky, but the work force of skilled air traffic controllers is limited. Doctors may need to see more patients when their number is limited. The military is often seeking to fly more unmanned air vehicles with a limited number of pilots to increase the productivity of surveillance, and hence push for more UAVs to be supervised by a single pilot. In such cases, workload is rapidly exceeded unless layers of automation are introduced (Cummings & Nehme, 2010; Dixon et al., 2005).

3. AUTOMATED-RELATED INCIDENTS AND ACCIDENTS

Although automation has yielded many benefits, at the same time it has introduced new problems that have occasionally led to accidents. Several highly publicized incidents and accidents have underscored the need for designing automated systems by taking human factors into account early in the systems requirements phase. Many such incidents have involved commercial automated aircraft (Billings, 1997; Parasuraman & Byrne, 2003). Analyses of these accidents have not only revealed that automation can introduce new vulnerabilities in system performance, but have also illustrated how human capabilities and limitations are brought to the forefront when designers introduce automation from a purely technology-centered perspective. We describe a few of the many such automation-related incidents and accidents.

A caveat must be noted before describing these examples. Most accidents are the result of multiple precipitating occurrences and conditions ultimately leading to the event (e.g., Reason, 1990, 2008). Consequently, attributing an accident exclusively to poor automation design can be difficult. Nevertheless, analysis of several incidents has pointed to a leading role for automation (Funk et al., 1999).

An early example was the 1972 crash of an L-1011 aircraft in the Florida Everglades while on descent to Miami. The crew became preoccupied with troubleshooting a problem with a landing gear indication light, and they did not recognize that the "altitude-hold" function of the autopilot had been inadvertently disconnected. A major factor contributing to this accident was the poor feedback on automation state provided by the system (Norman, 1990). In their report on the accident, the National Transportation Safety Board (NTSB) stated that disengagement of automation should be clearly signaled so that the pilot can validate whether it was intended or unintended (NTSB, 1973). In the L-1011 accident, the principle that automation states and state changes should be made salient to the human operator was violated. Most current autopilots now

provide an aural and/or a visual alert upon disconnect. The alert remains active for a few seconds or requires a second disconnect command input by the pilot before it is silenced.

At sea, an accident in which low saliency of alerts and high operator trust in automation (complacency) were major contributing factors in the grounding of the cruise ship *Royal Majesty* off the coast of Nantucket, Massachusetts, which resulted in several million dollars worth of damage to the vessel (Parasuraman & Riley, 1997). This ship was fitted with an automatic radar plotting aid (ARPA) for navigation that was based on GPS receiver output. The bridge crew had to monitor the ARPA while engaged in other duties. Because of a loss in the GPS signal due to a frayed cable from the antenna, the ARPA system reverted to "dead reckoning" mode and did not correct for the prevailing tides and winds, so that the ship was gradually steered toward a sand bank in shallow waters. The change in automation mode was signaled by a hard-to-see change in a single letter on a small, liquid crystal display (see change blindness in Chapter 3). At the same time, the crew continued to follow the ARPA display for over a day and failed to notice other indicators that the ship was in dangerously shallow waters, such as communications from small fishing vessels in the area and lights on the shore. The NTSB (1997) report on the incident cited poor interface design, crew over-reliance on the ARPA system, and complacency associated with insufficient monitoring of other sources of navigational information (such as another radar and visual lookout).

The upheavals in Wall Street over the past few years provide a third example illustrating the role of automation in catastrophic incidents. The financial crises in 2008 and 2010 were directly related to the use of computerized derivatives trading and other forms of automated transactions in the stock market. Automated trading has long been touted for its economic benefits (Domowitz, 1993; Steil, 2001), but an unintended consequence was the development of so-called high-frequency trading, where millions of shares were traded automatically without human intervention, creating extreme volatility that lead to the market meltdown of 2008 and again in 2010. The complexity and opacity of the algorithms underlying automated trading, coupled with human users (including those at regulatory agencies such as the Securities and Exchange Commission) who had limited understanding of the automation algorithms, were major reasons for the crises (McTeague, 2011). Furthermore, as noted by Taleb (2007), the problem with many of the algorithms in the financial models that went awry was that they assumed that human decision making was optimal. As discussed previously in Chapter 8, a large body of research has shown, however, that human decision making is dominated by heuristics and other cognitive "short cuts," which work most but not all of the time (Tversky & Kahneman, 1974). Unfortunately, these decision heuristics were never incorporated in the automation algorithms.

4. LEVELS AND STAGES OF AUTOMATION

Analyses of automation-related incidents and accidents reveal that the functionality of the automation has a major influence on how well human operators interact with automation in meeting their system performance goals. The different functions that automation can take have been described in a number of ways. Automation is not all or none, but can vary across a continuum of levels, from the lowest level of fully manual performance (no automation) to the highest level of full automation. Sheridan and Verplanck (1978), in proposing the concept of *supervisory control*, first suggested a taxonomy of 10 such **levels of automation.** Supervisory control refers to a system in which a human operator does not directly operate on the physical plant being controlled but does so through an intermediary, usually a computer, that has effectors to act on the environment based on information obtained from sensors (Sheridan, 2002; Sheridan & Parasuraman, 2006).

HIGH	10. The computer decides everything, acts autonomously, ignoring the human
	9. informs the human only if it, the computer, decides to
	8. informs the human only if asked, or
	7. executes automatically, then necessarily informs the human, and
	6. allows the human a restricted time to veto before automatic execution, or
	5. executes that suggestion if the human approves, or
	4. suggests one alternative
	3. narrows the selection down to a few, or
	2. the computer offers a complete set of decision/action alternatives, or
Low	1. The computer offers no assistance: human must take all decisions and action

FIGURE 12.1 Levels of automation scale (after Sheridan & Verplanck, 1978).

Figure 12.1 shows the 10-point Sheridan-Verplanck scale, with higher levels representing increased autonomy of computer over human action. For example, at a low level 2, several options are provided to the human, but the system has no further say in which decision is chosen. An example of level 4 automation would be a conflict detection and resolution system that notifies an air traffic controller of a conflict in the flight paths of two aircraft and suggests a resolution, but the controller retains authority for executing that alternative or choosing another one. At a higher level 6, the system gives the human only a limited time for a veto before carrying out the decision choice. Sheridan further refined this scale in subsequent published work (Sheridan, 2002; Sheridan & Parasuraman, 2006) and others have proposed related taxonomies (Endsley & Kaber, 1999).

It should be noted that the concept of levels of automation does not require that there be 10 levels; there is no "magic number" 10. What is most important is that the levels are defined such that higher levels define more responsibility for automation and reduced cognitive work for the human. The levels of automation concept also does not imply that humans and automation work as independent agents. As Sheridan and Verplanck (1978) first noted in their description of the supervisory control concept, the human and machine components are *inter*-dependent, with the human making plans to execute via the machine, monitoring its actions, and "teaching" it what to do next. The relative degree to which the human is engaged in these activities, however, varies with the level of automation. For example, as the automation takes on more responsibility, the human requirement for monitoring increases (Parasuraman, 1987).

The Sheridan-Verplanck scale is based on different levels of human versus automation involvement and control, but one can also think of automation also applied to different information-processing *stages*, from sensing through decision making to action. This book has been structured within a framework emphasizing stages of information processing, and automation too can be conceptualized in terms of how it augments or assists those different processing stages. Parasuraman et al. (2000, 2008) extended the levels of automation concept to cover **stages of automation** in human-machine systems. In this expanded version, a simpler form of the human information processing model described in Chapter 1 was adopted, a four-stage model consisting of information acquisition, information analysis, decision making, and action implementation (see Figure 12.2).

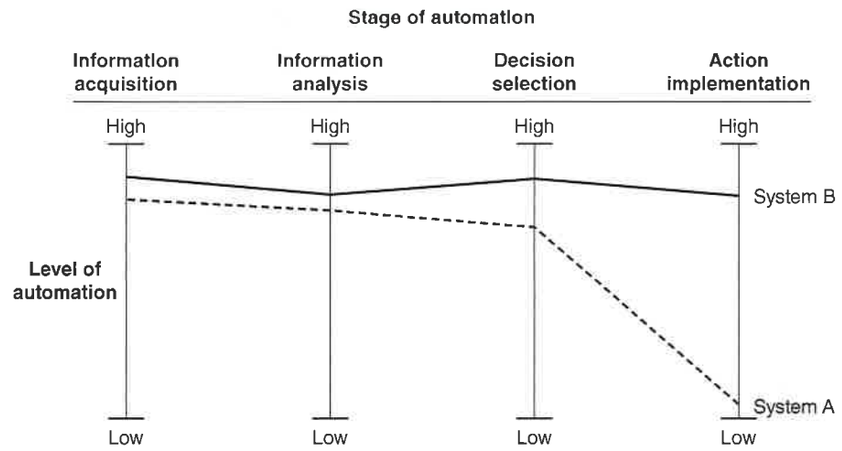

FIGURE 12.2 Model of levels automation for different information-processing stages (after Parasuraman, Sheridan, & Wickens, 2000).

The first stage in the model of Parasuraman et al. (2000) refers to the acquisition and registration of multiple sources of information. This stage includes sensory processing, initial pre-processing of data prior to full perception, and selective attention. For example, the alarms discussed in Chapter 2 are a form of automation designed to direct the user's attention to a problem. The second stage involves manipulation and integration of processed and retrieved information in working memory. This stage can also be conceptualized to include cognitive operations such as rehearsal, integration, and inference, such as situation assessment and automatic diagnosis. However, such operations are proposed to occur *before* decision making and action selection, which is the third stage where automation assists in making a choice. The fourth and final stage involves the implementation of a response or action consistent with the decision choice. The model proposes that automation can be applied at different levels to each of these four stages from completely manual operation to full automation.

While the four-stage model simplifies to some extent the complexities of the human information-processing model discussed throughout this book, with its many feedback loops and availing of parallel processing, it has proven useful as a framework with far-reaching implications for automation design. Furthermore, a model of *automation support* need not be as complex as the human it is meant to aid.

Figure 12.2 provides a schematic of the model of levels and stages of automation. A particular system can involve automation of all four dimensions at different levels. Thus, for example, a given system (A) could be designed to have moderate to high levels of automation of information acquisition, information analysis, and decision making, but a low level of action automation. Another system (B), on the other hand, might have high levels of automation across all four dimensions. An example of type A is the Theater High Altitude Area Defense (THAAD) system. THAAD, which is used to intercept ballistic missiles (Department of the Army, 2003), has relatively high levels of automation across information and decision stages; however, action implementation automation is low, giving the human full control over the firing of missiles. On the other hand, *Robonaut*, a robot used in extra-vehicular tasks during deep space missions,

represents an example of type B, with high automation across all stages (Bluethmann et al., 2003). We describe each stage within the taxonomy as follows.

4.1 Information Acquisition

Automation of information acquisition (stage 1 automation) applies to the sensing and registration of input data. These operations are equivalent to the first human information processing stage, supporting human sensory and selective attention processes. A low level of information acquisition automation could involve manipulation of sensors in order to scan and observe. For example, modern unmanned air vehicles (UAVs) typically have cameras that can provide a remotely located operator a video feed of a scene and are capable of features such as tilt or zoom (Cooke et al., 2006). In the area of health care, automation such as electronic medical records (EMR) can assist a physician by directing selective attention to sources of information about the patient or the medications they may be using. A somewhat higher level of automation at this stage could involve organization of incoming information according to some criteria (e.g., a priority list and highlighting of some part of the information). For example, modern air traffic control facilities use "electronic flight strips" that have the capability of listing aircraft in terms of priority for handling by the controller. In human-computer interaction, the "ping" of a newly arriving e-mail, or the highlighting of a misspelled word provide attentional guidance.

As noted in Chapter 2 in the section on signal detection theory, human operators can sometimes fail to detect critical events in the environment, and such failures to notice critical targets can be more prevalent if the work period is prolonged (vigilance). Information acquisition automation can mitigate both problems by providing alarms that direct operator attention to such events. When such alarms are simply triggered by a sensor, they can be characterized as relatively "dumb" and associated with a low level of information acquisition (stage 1) automation. However, when alarms integrate information from several sensors to make an inference regarding the identity or severity of a critical event, then such "smart" alarms qualify as information analysis (stage 2) automation. A fire alarm that integrates temperature and particulate concentration might be a simple example of such integration. Pritchett (2009) provides examples of both types as they are used in cockpit alerting systems.

4.2 Information Analysis

Automation of information analysis involves support of cognitive functions such as working memory and inferential processes. A low level of stage 2 automation could involve the processing of incoming data and presentation on the operator's display of the projected future course of that data, or so-called trend or predictor displays (Yin et al., 2011, see Chapter 5). For example, nuclear power plant control rooms have displays that show both the current and the anticipated future state of the plant (Moray, 1997). Such predictor displays were also discussed in Chapter 5 in relation to their use in process control environments. A higher level of automation at this stage involves *integration* of information values rather than only prediction. In such cases, the systems combine several input variables into a single value or object, as in the integrated polygon displays that are used in process control or surgical settings (Smith et al., 2006). In both these examples, information integration assists the human operator by reducing the demand on working memory and the need for effortful inferential processing. Diagnostic aids in medicine are prototypical examples of stage 2 automation (Garg et al., 2005). So too is the output of a computer statistics package that makes an inference that two means differ with a certain likelihood. A lower automation level may signal confidence intervals. A higher level will simply signal "significant" or "not significant."

The distinction between stage 1 and 2 *automation* corresponds closely to the distinction between level 1 situation awareness (noticing) on the one hand, and levels 2 and 3 SA (inference and prediction) on the other, as discussed in Chapter 7. Stage 1 automation assists (or replaces) the first of these, stage 2 assists the second.

4.3 Decision Making and Action Selection

The third stage, decision and action selection, involves selection of one from among alternative decision choices. Stage 3 automation involves providing the human decision maker either with an entire list of alternatives, a prioritized list, or a single best choice. Sheridan's original 10-level taxonomy (Figure 12.1) is applicable to this stage of automation. We discussed such types of decision automation in the context of "command displays" in Chapter 6. Important are the distinctions at the highest levels of stage 3 automation in which (a) the human may be offered a single option, but can choose to ignore it; (b) the human cannot ignore the option since it will be chosen (and executed) unless the human vetoes it (within some time limit); (c) the human cannot even veto. Levels (b) and (c) will also mandate the highest level of action implementation automation.

Examples of stage 3 automation can be found in many work domains. An example from aviation is the airborne traffic warning system, which provides a resolution advisory that tells the pilot to fly one particular maneuver (e.g., "climb climb") to avoid a collision with another aircraft (Pritchett, 2009). In health care, decision-aiding systems have been developed to support physicians in making diagnostic decisions about patients or treatments (Garg et al., 2005; Morrow et al., 2006). An example is the appearance on the display screen of computerized patient record systems of specific recommendations regarding treatment of a patient with HIV (Patterson et al., 2004).

It is important to note the distinction between stage 3 automation, which specifies which course of action the human operator should follow, from the previously discussed stage 2 automation, which only supports inferential processing that leads to a decision. In the context of the statistics package, one that goes beyond providing a p value, and tells the user whether to "accept" or "reject" the null hypothesis is invoking stage 3 automation.

The contrast between stage 2 and stage 3 automation is directly analogous to the contrast between what Mosier and Fischer (2010) describe as *front end* and *back end* decision making, respectively. The distinction is critical here, as it was in Chapter 8, because at stage 2, automation need not impose any *values* in making an inference on what is the likely diagnostic state or assessment of a situation. However at stage 3, automation *must* either explicitly or implicitly assume values for the different decision outcomes it is advising (or mandating), and these added assumptions leave room for greater departure between a human's choice and the recommendations of automation.

4.4 Action Implementation

The final stage of action implementation refers to the physical accomplishment of the action choice. Stage 4 automation involves machine execution of the choice of action, replacing human motor response (e.g., hand or limb movements or voice commands).

Different levels of action automation may be defined by the relative amount of manual versus automatic activity in executing the response. For example, in a photocopier, manual sorting, automatic sorting, automatic collation, and automatic stapling represent different levels of action automation that can be chosen by the user. A somewhat more complex example from air traffic control is the automated "handoff," in which transfer of control of an aircraft from one airspace

sector to another is carried out automatically via a single key press, once the decision has been made by the controller (Wickens et al., 1998). Robotic telesurgery, in which a surgeon guides a remote robot that carries out surgical actions on a patient, provides another example of a high level of stage 4 automation. Marescaux et al. (2001) reported successful use of such a system to allow a surgeon in New York to perform a gall-bladder removal operation on a patient 3,500 miles away in France.

The implications of the stages and levels model for automation design are discussed in Section 9.2 of this chapter. In the following sections we consider a number of different aspects of automated systems that can contribute to difficulties in their use by human operators.

5. AUTOMATION COMPLEXITY

Automation, by its very nature, replaces functions that were originally performed by humans, by mechanical or computer components. Thus, while eliminating human error, discussed in Chapter 9, the increased number of non-human components will increase the probability of a system error or fault. Furthermore, the greater the levels or complexity of an automation function, the more components it will contain and, using the reliability equation of Chapter 9, the greater is the possibility that something, somewhere, sometime, will fail. Thus, it is almost inevitable that automation in such complex systems will be imperfect. Automation imperfection can lead to problems of over- or under-reliance on automation, as discussed further in succeeding sections of this chapter.

An assumption that is often made is that computer-based automation can improve the reliability and safety of systems compared to analog or electro-mechanical devices because the hardware failure modes of these older technologies are reduced by using software. However software is also not free of potential failure. The increasing sophistication and complexity of software has lead to many more lines of code in automated systems. Often, new software developed by a company incorporates "legacy" code that was written by programmers long gone from the company and unavailable to provide information on the old code.

As an example of software size and complexity, the new Boeing 787 "Dreamliner" aircraft requires several million lines of code to run its automated systems. With such large systems, there is a significant probability that insidious "bugs" hiding within the software can lead to unforeseen problems (Landauer, 1995). Leveson (2005) has written extensively on the problem of "software safety" and the difficulty of software verification. She has also analyzed the role of software in many accidents involving aircraft, space vehicles, and other complex systems. In her analysis of the SOHO spacecraft accident in 1996, for example, she pointed out that **overconfidence** and complacency lead to inadequate testing and review of changes to ground-issued software commands to the spacecraft (Leveson, 2005). The human organizational response to software failures thus represents another type of automation-related accident, in addition to those discussed previously in this chapter.

Automation complexity brings with it the issue of *observability* to the human user. When complex algorithms are embedded within an automated system, the operator is likely not to understand why the automation performs a certain action because the algorithms are not observable, as was the case with many of the algorithms involved in computer trading that led to the stock market crash in 2008. In some instances the automation is so complex that it functions as an independent "agent" through which the human operator acts on the environment (Lewis, 1998). As a result, mutual intelligibility between the human and machine agent can be lost (Woods, 1996). Consequently, agent-based systems might be best served for relatively, simple, low-risk tasks. For more complex tasks involving contextual decision-making, however, such systems must provide feedback to the human operator so that agent intentions are understood (Olson & Sarter, 2000).

Increased automation complexity brings with it a second concern. If algorithms are so complex as to do things a different way from how humans normally (or previously) accomplished the same task, then the human operator may become surprised, and sometimes suspicious of automated functioning. An example is the flight management system (FMS), a collection of sophisticated autopilots that guide an aircraft through flight efficient routes, using algorithms and logic considerably more sophisticated than a pilot would use to fly the same routes (Pritchett, 2009; Sarter & Woods, 1995; Sarter, 2008; Sebok et al., 2012). Because of these complex, non-human (and therefore non-intuitive) algorithms, such systems will on occasion do things (legitimately) that pilots do not expect, and hence lead them to ask "why is it doing this?," a concept in aviation described as "**automation surprises**" (Degani, 2004; Sarter, 2008; Sarter et al., 1997). In general, such surprises do not have major implications *unless* they lead the human to assume that the automation has failed, and hence, intervene, perhaps inappropriately, situations that have led to fatal accidents (Degani, 2004).

6. FEEDBACK ON AUTOMATION STATES AND BEHAVIORS

If automation is not carefully introduced, it can have the characteristics that Sarter and Woods (1996) have labeled as "not a team player." Much of this deficiency may result from the absence of effective feedback to the human monitor of the automation's functioning, regarding what it is doing and why (Norman, 1990). This issue has long concerned pilots as they supervise their powerful, but complex and often uncommunicative FMS (Wiener, 1988; Sarter & Woods; 1995; Sarter, 2008). Deficiencies in automation feedback can be of several types: it can be completely absent, a state Sarter and Woods (1995) characterized as automation that is "silent;" it can be poor, in the sense of not being salient enough to draw the operator's attention to state changes; it can be ambiguous, so that the operator is confused; and finally, it can be inflexible, lacking detail, and not providing information specific to the situation.

When automation provides *no* feedback on its state, human operators can be left in the dark. As noted in Chapter 2, humans have difficulty in detecting subtle changes in the environment because of limitations in signal detection and vigilance capabilities. Even if feedback is presented, however, its saliency may be so low that operators do not notice it, particularly if their attention is focused elsewhere on their other tasks. As noted in Chapter 3, even apparently compelling changes in the environment (e.g., a man in a gorilla suit strolling through a group of people playing a pass-the-ball game; Simons & Chabris, 1999) can be missed if attention is directed elsewhere—the phenomenon of **change blindness**. On the flight deck, flight mode state annunciators appeared to go unnoticed if they were unexpected (Sarter, Mumaw, & Wickens, 2007). In the *Royal Majesty* ship accident that was discussed earlier in this chapter, the failure of the GPS signal to the automated radar system *was* signaled, but it was small (a change in one character of a small, liquid crystal display) that it was virtually unnoticeable.

Even when salient feedback is provided, additional communication deficiencies can result from the inherent inflexibility in the dialogue with most automated systems. Such systems must, after all, be preprogrammed with a fixed set of rules that limits their "conversational flexibility." The increasingly prevalent phone menu is the perfect example of such inflexibility, where a simple question one might have, that does not meet the pre-specified set of menu categories, cannot be easily handled. Often one must wait till the final option: "if you need to speak to an operator, press eight." Also, as we noted in Chapter 6, there are a number of non-linguistic features of

human-human communications that cannot be readily captured by computer mediated (i.e., automated) communications. We examine the issue of communication between automated systems and human users in more detail later in this chapter when we discuss the concept of "human-computer etiquette."

7. TRUST IN AND DEPENDENCE ON AUTOMATION

There is probably no variable more important in human-automation interaction than that of **trust**. Classic studies by Bainbridge (1983), Muir (1988), Wiener and Curry (1980), and by Lee and Moray (1992) introduced the concept, and early papers by Parasuraman et al. (1993) and Sorkin (1989) introduced concepts of **complacency (over-trust)** and the "**cry wolf effect**" (**under trust**), respectively, concepts that will be described in depth below. There has subsequently emerged a large literature on trust and its relation to automation usage and human-system performance. Lee and See (2004) provided an overview of this work and a process model of trust in automation. Madhavan and Wiegmann (2007) extended this review and also compared human-human and human-automation trust, observing that the two had features in common but could also be distinguished. Hancock et al. (2011) reported a meta-analysis of trust studies in the specific context of human-robot interaction.

At the outset, it is essential to distinguish between automation trust and **automation dependence**. The former is a cognitive/affective state of the user that is typically assessed with subjective ratings (e.g., Jian et al., 2000; Singh et al., 1993); the latter is an objective behavior that can be measured from the user's interaction with the automation (e.g., Lee & Moray, 1992). It may for example be measured by the extent to which the user "turns automation on," follows its advice, or cross checks automation's recommendations against the raw data (Bahner et al., 2008).

Automation trust and dependence are usually correlated: if we trust an agent, whether a machine or a human, we will tend to depend on that agent. For example, "I trust my teenage daughter not to text while driving since I have lectured her many times that it is an unsafe practice;" or "I trust my automatic teller machine to give me the correct amount of cash in a transaction without my having to count the money, because I have never been short changed." This correlation between trust and dependence is often considerably less than 1.0. We may be forced to depend on the automation when our workload is high, but may not always fully trust it; sometimes we may "look over its shoulder." We may also fully trust automation, but may not depend on it at all, if we simply prefer to do the task manually because of the excitement and challenges of the latter.

Several variables are known to affect both trust and dependence in the same way. For example, the more complex the algorithms of the **process** of automation, the lower is the trust (Lee & See, 2004). Certainly this was a major source of pilot *mistrust* of the hugely complex FMS described in Section 5 above. Closely related is the loss of trust caused by the lack of transparency or feedback of what automation is doing described in 6 above. What goes on inside the "black box?" Madhavan et al. (2006) found that the kinds of mistakes made by automation also affect trust. Really "bad" automation errors degrade trust more than plausible errors (like the kinds the human user would make). There are also individual differences in trust/dependence (Krueger et al., 2012; Merritt & Ilgen, 2008).

Of all the variables to affect trust/dependence, probably the most critical is **automation reliability**. Perfectly (100 percent) reliable automated systems are rare except when they are extremely simple. This necessarily means that human operators may sometimes choose not to trust the output of an automated system. Of course, for complex systems operating in an uncertain

world, perfect performance is virtually impossible, whether the task is executed by automation *or* by a human expert, because of the inherent uncertainties involved in the information which automation must process, in domains such as weather forecasting, economics, disease progression, or prediction of the behavior of individual humans (such as in terrorism or mental health). Though imperfect, automation can provide useful assistance to the human in such areas (Wickens & Dixon, 2007). As other causes that degrade reliability, we have identified above the role of software bugs in causing automation imperfections; and we can consider the role of power failures (the calculator giving out in the middle of the exam, forcing reliance upon mental long division), and improper human "set up" or programming of the automation (Wiener & Curry, 1980). The latter two may not be considered failures *of the automation* itself, but they can have similar consequences for trust and dependence.

Whatever the sources of unreliability, a critical concept in the relation between automation trust/dependence and reliability is the **calibration curve**, shown in Figure 12.3. Here reliability is scaled on the X-axis (and can often be expressed numerically on a 0–1.0 scale, by dividing the number of automation errors by the opportunity for errors). Either trust or dependence is represented on the Y-axis, trust by a minimum-maximum subjective rating scale, while for dependence, any number of objective measures can quantify the proportion of times automation is used (e.g., the proportion of times a human decision agrees with an automated recommendation). The diagonal line represents the line of *perfect calibration*. Importantly, the curve bisects the space into two regions, elaborated in the sections below: over-trust to the upper left, and under trust to the lower right. It often happens that these two sections are linked in time via the *dynamics of trust*. (Lee & Moray, 1992; Yeh, Merlo, et al., 2003). In a typical scenario the operator works with an automation system of high reliability. It may operate for many "trials" (or a long time) without failure, and during this time the operator builds up trust in and dependence on it, often to the point of being *complacent* far to the upper left of Figure 12.3. Then it fails, in what we describe as the first failure, an event that has particular significance in the study of human-automation interaction (Rovira et al., 2007). The human response (or non-response) to these first failures are often dramatic (Yeh, Merlo, et al., 2003) and represent the source of many automation-based accidents, such as the *Royal Majesty* grounding described above (see also Dornheim, 2000, for examples of first failure experience in aircraft automation).

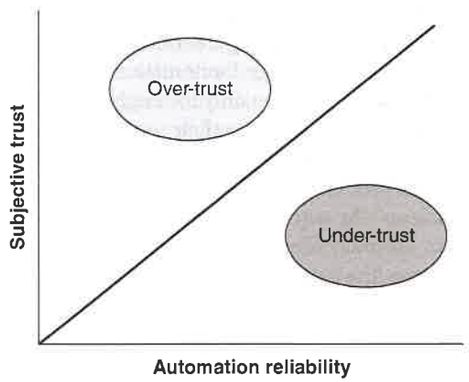

FIGURE 12.3 The relation between subjective trust and automation reliability.

Following the first failure the operator will then typically leap across the calibration curve of Figure 12.3 to the far bottom right, showing a great amount of mistrust ("burned once, never again" or "Fool me once, shame on me. Fool me twice, shame on you"). Then, over time, trust and dependence will gradually recover toward the calibration line at a level approximating long range reliability (Yeh, Merlo, et al., 2003). In the following discussion, we present these two regions in their typical sequence, from over-trust to first failure to under-trust.

7.1 Over-trust

7.1.1 COMPLACENCY Automated systems that are highly reliable but not perfectly so can invite the tendency not to monitor automation or the information sources that the automated system uses. The term complacency has long been used in aviation (Wiener, 1981) and other domains (Casey, 1988). As in the case of the cruise ship *Royal Majesty* that was described earlier, complacency has been implicated as a contributing factor in many accidents.

While automation complacency will provide ample resources for concurrent tasks prior to the first failure, it can have at least two behavioral consequences upon its failure. On the one hand, the infrequent and therefore unexpected automation failures, when they do occur, are hard to detect, as we learned in Chapter 2 (expectancy and signal detection), Chapter 3 (expectancy and visual scanning), and in Chapter 9 (expectancy and reaction time). On the other hand, an operator who expects that automation is doing its job will be less likely to monitor the job it is doing—losing awareness of the evolving state of, or surrounding, the automated system (Endsley & Kiris, 1995; Kaber et al., 1999; see situation awareness: Chapter 7). Hence, if the failure *does* occur, the monitor will be less able to deal with it appropriately. An example would be a pilot who has to jump into the control loop to fly the aircraft manually, should the autopilot unexpectedly fail. Furthermore, research reveals that it is easier to remember an action if you have chosen it yourself than if you have witnessed another agent (another person, or automation) choose that action—the **generation effect** (Farrell & Lewandowsky, 2000; Slamecka & Graf, 1978). Thus, automation leaves the operator less aware of the chosen actions in the system. For example, what is the mode of automation currently in effect (Sarter & Woods, 1997)? Of course complacency does not reveal a problem *until* automation fails and such a failure, although often unlikely, is never impossible.

Experimental evidence for automation complacency was provided by Parasuraman et al. (1993), who had participants perform three concurrent tasks from the Multiple Task Battery (MATB), one of which (an engine monitoring task) was supported by an automated system that was not perfectly reliable. Complacency was operationally defined as the operator not detecting or being slow to notice failures of the stage 1 automation to detect engine malfunctions. In a control condition, participants had to perform only the engine-monitoring task with automation support, without the other manual tasks, so that their overall task load was considerably lower. Detection of automation failures was significantly poorer in the multi-task condition as resources were shared between the tasks (see Chapter 10) than in the single-task condition. When participants had simply to "back up" the automation routine without other duties, monitoring was efficient and near perfect in accuracy. Thus, automation complacency represents an active reallocation of attention away from the automation to other manual tasks in cases of high workload (Manzey & Parasuraman, 2010).

As discussed in Chapter 2, operators are less likely to detect signals when they occur infrequently, a finding consistent with the expectancy theory of vigilance (Parasuraman, 1987). Thus, automation complacency should be more severe when automation failures are infrequent, occur for the first time in the operator's experience (*first failure*), and/or occur after long periods of error-free performance. Indeed, one of the ironies of automation is that the more reliable it is, the more it is trusted, and the more complacent the operator becomes (Bainbridge, 1983). Molloy and Parasuraman (1996) confirmed this in a study in which the automation failed on only a *single* occasion, either early or late, in two separate sessions. Only about half the participants detected the early automation failure, and even fewer detected the late failure (see also replications by Bailey and Scerbo (2007) and by Manzey et al. (2012)). In a related study, De Waard and colleagues (1999) had participants in a simulator drive a vehicle in which steering and lateral control were automated but could be overridden by depressing the brake. On a single occasion a vehicle merged suddenly into the same lane as in front of the participant's vehicle but the automation failed to detect the intrusion. Half the drivers did not detect the failure, depress the brake, and retake manual control, while 14 percent did respond but not quickly enough to avoid a collision.

Complacency in stage 1 automation (alerts and alarms) can be reflected in two different forms of automation dependence: reliance and compliance (Meyer, 2001, 2004, 2012). When operators stop monitoring the *raw data* during the long periods while the alert is "silent" (or not activated), they can turn their attention elsewhere to support concurrent tasks. This form of dependence is described as high **reliance** on automation (Dixon & Wickens, 2006; Meyer, 2001). When operators react rapidly when the alert "sounds" (or is activated), this reflects **compliance**. While a change in reliance does not necessary mandate a change in compliance (or vice versa), the two states are often reciprocally coupled via the *alert threshold* discussed in Chapter 2. That is, decreases in the alert threshold (beta in SDT terms, see Chapter 2) typically will cause compliance to decline, as reliance will increase (Dixon & Wickens, 2006; Maltz & Shinar, 2003).

Automation complacency has also been found in studies with skilled workers supervising automation that closely resembles real systems. Galster and Parasuraman (2001), for example, found that experienced general aviation pilots detected fewer engine malfunctions when using an actual cockpit automation system, the Engine Indicator and Crew Alerting System (EICAS), than when performing all flight simulation tasks manually. Yeh et al. (2003) demonstrated the strong **first failure effect** with army personnel using attention-guidance automation as discussed in Chapter 3. Metzger and Parasuraman (2005) tested experienced controllers on a high-fidelity air traffic control simulator with "conflict probe" automation that pointed to a potential conflict between two aircraft several minutes before its occurrence. Significantly fewer controllers detected a conflict when the conflict probe failed than when the same conflict was handled manually in a separate session. Eye movement analysis also showed that controllers who missed the conflict made significantly fewer fixations of the radar display under automation support than under manual control, pointing to a link between the automation complacency effect and reduced visual attention to the raw data information sources feeding automation (see also Bagheri & Jamieson, 2004; Manzey et al., 2012; Wickens, Dixon, Goh, & Hammer, 2005).

The evidence therefore suggests that automation complacency is typically found under conditions of multiple task load, when manual (non-automated) tasks compete with the automated task for the operator's attention. This finding is also consistent with the meta-analysis of stage 1 automation reliability studies by Wickens and Dixon (2007). They found automation dependence, reflecting complacency, was more correlated with automation reliability in dual task conditions, when cognitive resources were scarce, than in single task conditions. Under such multi-tasking conditions, the operator's attention allocation policy appears to favor his or her manual tasks, as opposed to the automated task. This strategy may itself stem from an initial

orientation of trust in the automation, which is then reinforced when the automation performs without failure (Parasuraman & Manzey, 2010).

Moray (2003; Moray & Inagaki, 2000) pointed out that an attention allocation policy devoted primarily to non-automated tasks and only occasionally to the automated task can be considered *rational* (see also Moray, 1984; Sheridan, 2002). Moray also suggested that complacency could only be inferred if the operator's rate of sampling of the automated task was actually *below* that of an optimal or normative observer. After all, if something never fails (in your experience), why do you need to look at it? The reason of course, is that it *could* fail. In terms of the SEEV model of scanning (Chapter 3), the *value* of monitoring automation is extremely high, even if the expectancy is quite low. But people often use their own actuarial experience to guide their expectancies (Hertwig & Erev, 2009; see Chapter 8).

In support, Bahner et al. (2008) conducted a study to examine sampling of raw data processed by automation in which they compared how often participants looked at the optimal number of information sources needed to verify automation diagnosis. They had participants perform a simulated process control task requiring supervisory control of sub-systems of a life support system for a space station. An automated fault management system provided recommended diagnoses regarding system faults. The extent to which participants accepted the automation's diagnosis without verifying it provided a measure of complacency. Participants could access (via mouse click) all relevant system information (e.g., tank flow rates) needed to verify the diagnosis. Bahner and colleagues reasoned, following Moray (2003), that a participant who accessed the correct number of information parameters needed to verify a diagnosis before accepting it was optimal whereas one who sampled less information than that necessary to completely verify the aid's recommendation could be classified as complacent. *All* participants sampled less than the optimal number of information sources and several demonstrated poor detection of the first failure. Thus complacency was a general finding. Manzey, Reichenbach, and Onasch (2012) found that more optimal samplers were less likely to miss the first failure altogether and Bahner et al. (2008) also found that those participants who had been specifically trained with examples of automation failures had higher sampling rates and intervened more appropriately when automation failed. The results provided strong evidence for the existence of automation complacency, but also pointed to one method (training) that can be used to reduce its incidence: pre-exposure to the automation failure.

7.1.2 AUTOMATION BIAS What has been referred to by Mosier and Fischer (2010) as **automation bias** represents another human performance consequence of over-trust. Closely related to complacency, the automation bias has typically been associated with automated decision aids that are meant to support human decision-making in complex environments (Mosier & Fischer, 2010; Mosier et al., 1998). If the users of such systems have strong trust in such automation, they may ascribe it greater power and authority than other sources of information and advice. Mosier and Skitka (1996, p. 205) defined automation bias as "a heuristic replacement for vigilant information seeking and processing." In this view, individuals may not conduct a thorough analysis of all available information but simply follow the automation advisory, even when the advice is incorrect, thereby committing an error of commission (Bahner et al., 2008). An example is a pilot following the advice of a flight planning automated system although its recommendations are wrong (e.g., Layton et al., 1994).

In an early flight simulation experiment, Mosier et al. (1992) found that 75 percent of pilots incorrectly shut down an engine, when the automation also incorrectly recommended such a shutdown (stage 3 automation) based on its incorrect diagnosis of an engine fire in the wrong engine. In contrast, only 25 percent of pilots using a traditional paper checklist committed the same commission error. A later study revealed that commercial pilots were just as prone to follow such incorrect automation advice. The failure to check the "raw data" is what has been previously described as automation complacency, reflecting the allocation of the operator's attention to other non-automated tasks in busy multi-tasking environments.

The automation bias can also create attentional tunneling, discussed in Chapter 10. Thus Wickens and Alexander (2009), summarizing several flight simulator studies with skilled pilots, observed that 52 percent of pilots followed the direction of the automation stage 3 highway-in-the-sky (HITS) display that directly led them into the obstacle or hazardous path, even though the hazard was visible had pilots consulted the raw data, visible through the windshield outside the airplane.

Certainly, at least some aspects of the automation bias are due to the same attentional limitations that also lead to complacency (Parasuraman & Manzey, 2010). However, other aspects of automation bias may reflect decisional rather than attentional factors (Goddard et al., 2012; Mosier & Fischer, 2010). In this view, automation bias, like other decision heuristics and biases, reflects the tendency of humans to choose the road of least cognitive effort in decision-making, the so-called "cognitive miser" hypothesis (see Chapter 8 on decision making). Automation bias may also occur due to users overestimating the capability of automated aids. More specifically they may ascribe to the aid's greater performance and authority than to other humans or themselves (Dzindolet et al., 2002).

Goddard et al. (2012) reviewed studies of automation bias in health care settings, focusing on the use of decision support systems. They found that automation bias was relatively prevalent in many types of medical diagnostic decision making situations, particularly computer-aided detection of radiological images such as mammograms and in computer-based interpretation of electrocardiograms. In each case, participants had reduced diagnostic accuracy when provided with erroneous advice by the automation, compared to performance without automation.

7.1.3 OVERDEPENDENCE: DESKILLING AND "OOTLUF" In addition to complacency and the generation effect (loss of SA), a third negative consequence of high-level automation is that the operator's ability to carry out the automated manually may decline over time, a phenomenon sometimes called "deskilling" (Ferris, Sarter, & Wickens, 2010; Geiselman, Johnson, & Buck, 2012; Lee & Moray, 1994). There is evidence for such skill loss among pilots of highly automated aircraft, which are mitigated by the pilots choosing to "hand fly" the aircraft from time to time (Wiener, 1988). Collectively, the three phenomena of degraded detection through complacency, awareness/diagnosis, and manual skill loss may be referred to as the syndrome of "**out of the loop unfamiliarity**" or "OOTLUF".

Automation-related incidents and accidents were previously described in this chapter. However, a number of recent aviation accidents have specifically pointed to the issue of deskilling as a direct result of automation. A highly publicized accident was the Colgan Air crash near Buffalo, NY in 2009. The co-pilot had input incorrect information into the FMS, causing it to slow to an unsafe speed that triggered a stall ("stick shaker") warning. The loss in aircraft speed was apparently not noticed by the flight crew, but when the stall warning came on, the captain responded by repeatedly pulling back on the control yoke, which further caused the aircraft to stall and crash, resulting in the death of 49 people on board. The accident investigation suggested that the crash could have been avoided if the captain had pushed the yoke forward rather than back. A similar accident was the Air France crash in the Atlantic in 2009, also involving a high altitude stall in which the pilot made a "nose up" yoke input whereas the opposite should have been done

to maintain lift. In these and related accidents, pilots' loss of skill in handling stalls due to extensive use of autoflight systems has been thought to be a major contributing factor.

In many automated systems, OOTLUF concerns are pitted against the very real automation benefit of reduced workload. For the busy vehicle driver, navigating in an unfamiliar freeway environment, it will likely be both preferred and a true benefit to safety, to offload some aspects of the inner loop driving control (lane keeping and headway monitoring) to an intelligent and reliable autopilot, in order that navigational information can be consulted and decisions can be made without diversion of resources. But the implications of this tradeoff should be considered carefully, so that the OOTLUF syndrome does not occur. In Section 9.2, we address whether there is evidence that there might be optimum levels of automation on the tradeoff that do not produce OOTLUF, yet still provide automation at a high enough level so that workload is tempered (Wickens, 2008).

7.2 Mistrust and Alarm False Alarms

As noted at the beginning of Section 7, the first failure will often drive the operator from the state of over-trust to that of under-trust or distrust, just as other factors too, such as complexity and poor feedback can cause mistrust. As a consequence, such automation may be abandoned (Parasuraman & Riley, 1997), even when it is accurate (after all, 10 percent unreliable automation will still be accurate 90 percent of the time). Nowhere is the phenomenon better illustrated than in the "alarm false alarm" problem within automation stages 1 and 2, in which an alarm (a form of automated advice) will sound, even if no actual failure condition exists (Dixon et al., 2007; Parasuraman et al., 1997; Sorkin, 1989). Such circumstances invite the operator to mistrust the alarm system—that is, to be "under-calibrated" as to the true value that the alarm can offer.

Whether because of true unreliability, or complexity (leading to perceived unreliability), automation disuse can have consequences that may be relatively minor—sometimes we are less efficient when we turn off automation than we would be with its assistance. For example, Wickens and Dixon (2007) found that people were better off depending on automated diagnostic systems with as high as a 20 percent error rate, than they would be relying on their own manual diagnostic skills. In contrast, catastrophic incidents may sometimes occur because a true (valid) alarm was ignored or if a critical condition was never announced in the first place because the "annoying" unreliable alarm system had been turned off previously. Sorkin (1989) reported many cases of train engineers taping over the speakers from which auditory alerts emanated in the cab because they were typically false. Seagull and Sanderson (2001) reported that 42 percent of alarms heard by anesthesiology nurses were ignored (no action taken), and Wickens, Rice, et al. (2009) found that 45 percent of the conflict alerts received by air traffic controllers required no action (nor was one taken). In the domain of weather forecasting. Barnes et al. (2007) reported that 76 percent of tornado warnings that were issued were false.

Sometimes the "cry wolf" response can have tragic consequences. In one report it was concluded that 21 percent of the deaths or injuries related to long-term patient ventilator incidents resulted from delayed or no responses to ventilator alarms (Joint Commission, 2002). The 2001 crash of a Korean Airlines flight in Guam in which over 100 people died represents a particularly tragic consequence of the "cry wolf" syndrome. The air traffic controllers monitoring the flight had disabled the terrain collision avoidance system because it had issued too many false alerts and consequently did not notice that the aircraft was descending into a mountain short of the runway.

There are of course many reasons for not responding to a false alert or false warning, unrelated to any loss of trust (Lees and Lee, 2007; Wickens, Rice, et al., 2009; Xiao et al., 2004),

particularly if the operator also has some access to and has awareness of the raw data or information that lead to the alert. As was discussed in Chapter 2, the response threshold (or criterion) of alert systems is often set at a low level to guard against misses, but at the cost of false alarms. But if the false alarm nevertheless warns the operator of a *potentially* dangerous future situation even if it is not a true danger, it can still be useful and may not increase mistrust or lead to the "cry wolf" syndrome. For example, Lees and Lee (2007) argued that automated alerts in cars (e.g., collision warnings) can supplement the driver's judgment as to what safe driving maneuver to execute. Similarly, Wickens, Rice, et al. (2009) conducted an analysis of conflict alerts issued in an air traffic control center and found little evidence that controllers were prone to the "cry wolf" effect because the alerts, even if false because of a low alert threshold, reinforced the controllers own perception of the raw data (that a close passage of the two aircraft was coming).

In conclusion, the relation between trust, dependence, and reliability is complex, but there is no doubt that humans are not always optimal in calibrating their cognition and behavior, with potentially serious consequences whether over-trust or under-trust is manifest. In the following sections, we discuss some potential solutions for harmonizing human-automation interaction, including adaptive automation, finding an optimal balance in the level and stage of automation to balance the tradeoff between workload and OOTLUF that underlies calibrated trust.

8. ADAPTIVE AUTOMATION

Thus far in our discussion of different aspects of human interaction with automation, we have assumed that the functional properties of the automation, once designed and implemented, remain constant or static during system operations. This approach, in which the characteristics of automation are set at the design stage and then executed in the same way during operational use, has been referred to as static automation. In contrast, in **adaptive automation**, the level and/or stage of automation is not fixed but may change during system operations. (Feigh, Dorneich, & Hayes., 2012; Hancock & Chignell, 1989; Inagaki, 2003; Kaber et al., 2005; Kaber & Kim, 2011; Parasuraman et al., 1992, 1996; Rouse, 1988; Scerbo, 2001).

A general schematic for adaptive automation is shown in Figure 12.4. The cognitive state of the operator, in this case illustrated by mental workload or the capacity of the human to perform,

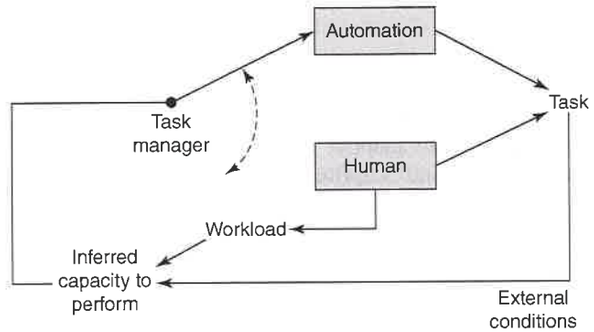

FIGURE 12.4 Adaptive automation. Workload or the capacity of the human to perform is inferred and used by a "task manager" to assign more of a task to automation (if workload is high) or to the human (if workload is reduced). The task manager itself could be automation, human, or a cooperative enterprise.

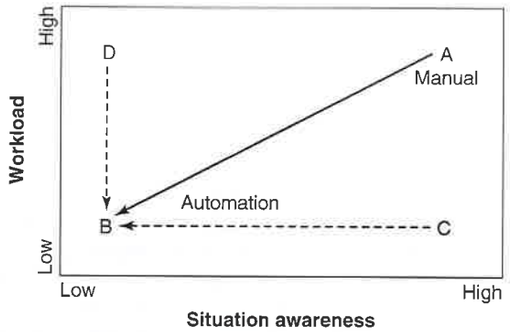

FIGURE 12.5 Three possible strategies of adaptive automation. It is assumed that point B is at a higher level of automation than points A, C, and D.

is inferred and used by a "task manager" to assign more of a task to automation (if workload is high) or to the human (if workload is reduced). The task manager itself could be automation, human, or a cooperative enterprise. Figure 12.5 shows some of the possible ways in which adaptive automation can change workload and situation awareness in order to maintain a balance between the two.

Adaptive automation is akin to dynamic **function allocation** (Lintern, 2012; Winter & Dodou, 2011), in which the division of labor between human and machine agents is not fixed but changeable, flexible, and context dependent. For example, if high human workload is inferred at a particular level of automation and impending performance breakdown is suspected, automation may go to a higher level to support the operator. At other times, if the operator is in danger of losing situation awareness due to working with high-level automation, he or she may be brought back more in the loop through a reduction in the level of automation. In general, adaptive systems seek to limit the potential costs of automation, in particular OOTLUF, and to boost overall system performance by changing automation functionality during system operations. Adaptive automation, in contrast to static automation, allows for restructuring the task environment in terms of (a) what is automated, (b) how to infer, and (c) when changes occur.

8.1 What to Adapt

The first issue concerns what aspect of a task (or task complex) should be adapted. Parasuraman et al. (1999) distinguished between adaptive aiding, in which a certain component of a task is made simpler (by automation), and adaptive task allocation, in which an entire task (from a larger multitask context) is shifted to automation.

Here a reasonable argument can be made that the appropriate choice should be one that reduces workload to the greatest extent, even as it also reduces situation awareness (i.e., moves from point A to point B in Figure 12.5). The rationale for such an argument is that if the adaptive automation moves from C to B, there is no workload savings and hence no reason to invoke automation in the first place; and if it moves from D to B, the task component might as well be fully and inflexibly automated, since this would produce no loss of situation awareness. Such a criterion could be applied independently of whether adaptive aiding or adaptive task allocation is implemented.

8.2 How to Infer

A second, critical issue concerns how to infer when adaptive changes should be made. To do so effectively, both the operator and the automated system must have knowledge of each other's current capabilities, performance, and state. Several different approaches have been proposed to generate criteria for adapting automation to the user (Parasuraman et al., 1992; Rouse, 1988): (1) Environmentally determined, where automation functionality is varied in response to easily measurable environmental changes or external task conditions, e.g., providing descent advisories to air traffic controllers only when traffic load or complexity is high but not otherwise (Hilburn et al., 1997); (2) continuous assessment of operator performance (Kaber & Endsley, 2004; Parasuraman et al., 2009); or (3) continuous assessment of mental workload, through neuroergonomic measures (Wilson & Russell, 2007), so that the operator is aided when a suboptimal state (e.g., high workload) is detected.

As examples of *environmental triggers*, Parasuraman et al. (1999) demonstrated the success of adaptive automation in aviation, that was invoked in takeoff and landing phases (known to be most demanding), but removed during the low-workload midflight cruise portion. In this case the external conditions were the known phase of flight. Inagaki (1999) suggested that different time periods during the acceleration of an airplane for takeoff make it more or less important for automation to assume responsibility for a rejected takeoff decision, should such a decision be required following an engine failure. Here the passage of time and speed of the aircraft is the external condition. In driving, one might consider an automation aid that uses the darkness of night (an external condition) to infer that a driver might be more fatigued and less vigilant, hence adapting an automated alerting device, sensitive to lane deviations.

Measures of performance can also drive adaptation, particularly to the extent that good performance modeling has revealed clear "leading indicators" that preview subsequent breakdowns. Kaber and Riley (1999), for example, demonstrated the benefits of adaptive aiding on a primary task (a dynamic cognitive monitoring and control task) that was based upon degradation of an automation-monitored secondary task. Parasuraman et al. (2009) used performance on a change detection task—detecting icon changes on a situation map—to drive adaptive aiding (automatic target recognition, ATR) on a task requiring supervision of multiple unmanned air and ground vehicles. Compared to performance without the ATR, or to static automation where the ATR was continuously available, the adaptive automation condition was associated with reduced workload and increased situation awareness.

Mental workload or cognitive state can be monitored directly as assessed by physiological measures, as discussed in Chapter 11. Such measures have some advantages over performance measures, primarily their higher bandwidth and the ability to be obtained even in absence of any overt behavioral output which might otherwise pull attentional resources away from the primary task of interest (Kramer & Parasuraman, 2007). In the past decade a number of studies have explored the possibility of using measures such as EEG, heart rate, etc. in adaptive automation (Dorneich, Ververs, et al., 2012; Feigh et al., 2012; Scerbo, 2001).

For example, Wilson and Russell (2007) had participants supervise unmanned air vehicles (UAVs) that provided radar images of critical target locations at which weapons had to be released. EEG, eye movements, and heart rate were monitored and used to train an artificial neural network to recognize low and high mental workload. On detecting high workload, the operator was aided by automation that slowed the speed of the relevant UAV, thus giving the operator additional time to complete the targeting task before the vehicle reached the weapon release point. Compared to a manual condition or to one in which adaptive aiding was provided randomly, adaptive automation led to a significant improvement in targeting performance.

Physiological measures of operator state can be combined with other environmental and operator measures to infer when to aid the operator. Ting et al. (2010), for example, used an artificial neural network to integrate different physiological indexes (heart rate, EEG) with performance measures to trigger adaptive aiding in a simulated process control task. Inagaki (2008) used physiological measures such as ear and nose tip temperature and eye fixation data in combination with performance measures to determine when drivers needed assistance in simulated collision scenarios.

A DARPA research and development program known as Augmented Cognition has been developing neurophysiological-adaptive systems that could potentially be fielded (Schmorrow, 2005). In one of the studies funded by this program, Dorneich et al. (2007) used EEG recorded in mobile individuals to estimate high workload and to drive a communications scheduler that would block incoming messages when high workload was detected. Similar studies using several EEG parameters to estimate operator workload in real time using different machine learning techniques to drive adaptive automation have also been reported (Baldwin & Penaranda, 2012; Christensen et al., 2012; Wang et al., 2012).

How practical are adaptive systems that use physiological measures to assess operator cognitive state? Clearly, despite their advantages, such measures face challenges such as obtaining artifact-free measurements in real work environments and user acceptance (Cummings, 2010). The prominent concerns with both leading indicators of performance and assessments of physiological state is that both of these sources require some time to integrate a sufficient amount of data so that a reliable inference can be made that the capacity to perform is diminishing (or restored). If adequate time and data are *not* allowed, then an inference of capacity change might be wrong, and this could lead to adaptation increasing workload when a decrease is desired, or vice versa. On the other hand if sufficient time to attain a *reliable* estimate is used and dynamic changes in environmental workload are present, then, given the feedback loop shown in Figure 12.4 the resulting lag in inference could produce **closed loop instability**, in the sense described with tracking tasks in Chapter 5. That is, an inference of high (or low) workload could be drawn after a substantial delay, and adaptive aiding implemented (or removed) at the very time that workload has now diminished (or increased). In this regard it is important that advocates of closed loop adaptive automation systems endeavor to establish the time required to make reliable estimates of workload based on EEG or other measures (Christensen et al., 2012).

8.3 Who Decides?

The third issue regarding adaptive automation, and perhaps the most controversial one, is the issue of "who decides" whether to implement or remove automation. That is, in the context of Figure 12.4, who is the "task manager?" In the previous section, it was implicitly assumed that the machine itself was responsible for invoking automation, following the signal of one or more of the three inference sources; external conditions, leading indicators, or workload. We might say that the task manager is itself working at the highest level of stage 3 automation (Figure 12.2). In contrast, an argument could be made that humans themselves are capable of monitoring their own workload (or capacity to perform) and making the appropriate choice to invoke or remove higher automation levels, a concept sometimes referred to as *adaptable* automation (Ferris et al., 2010).

Existing data remain ambiguous as to where the choice should lie. One relevant issue is the accuracy of the humans' own assessment of their capability to perform. To the extent that humans tend to be overconfident or inaccurate in this ability (Horrey et al., 2009; see also Chapter 8), particularly in relation to machine performance at equivalent tasks (Liu et al., 1993), then some caution should be exercised concerning the wisdom of human choice.

Reinforcing this preference for machine over human choice is the results of the adaptive aiding study by Kaber and Riley (1999). Using secondary task performance to implement adaptive aiding on a video game task, Kaber and Riley compared two strategies: a *mandating* strategy directly implemented the aiding when it was assumed, by automation, to be desirable (highest stage 3 automation), whereas an *advising* strategy only provided the corresponding suggestion (a lower level of stage 3 automation). The authors observed a cost for the less automated advising strategy, a cost that they attributed to the added workload demands when operators must monitor their own performance, and then decide whether or not automation was required. Inagaki (2008) also noted that machine authority to implement adaptive changes when an inference is made that the human cannot avoid a hazardous situation effectively, while controversial, could be justified in certain instances. However, such a position is likely to be highly domain specific. Inagaki (2008) suggested that it may be easier to justify computer authority with everyday users of automation such as car drivers, who are likely to vary considerably in abilities and skill, than it would be for skilled, expert users such as aircraft pilots or physicians.

To these formal, data-driven arguments for computer authority as task manager can be added consideration of some compelling hypothetical scenarios. For example, most people would probably agree that automation should be responsible for adapting automated steering and slowing (along with alerting) should a *reliable* inference be drawn that the driver has fallen asleep, or is otherwise incapacitated. But the key factor here is reliability; and it would seem that the less reliable the inference, the lower level or earlier stage that the automation decision should be, on the stage 2 (information analysis) scale of Figure 12.2. For example, in this particular case, consider alerting rather than seizing control. By adapting mid levels of this scale, the designer is thus endorsing a collaborative and cooperative human-machine concept; one well within the spirit of **human centered automation**.

Miller and Parasuraman (2007) also suggested that there exist many situations in which putting the human in charge of changes in automation functionality can be beneficial. They outlined an architecture for such "adaptable" control of automation (see Opperman, 1994) called "Playbook," in which the human can delegate tasks to automation at either a "hands off" high level or by specifying various stipulations and constraints. Parasuraman et al. (2005) reported a study of simulated human-robot interaction in which they found performance benefits for the Playbook approach to adaptive automation.

The concept of adaptive/adaptable automation is a conceptually attractive approach to human-machine system design, capitalizing on the strengths of human and machine in a dynamic and cooperative fashion (Winter & Dodou, 2011). The related concept of **adjustable autonomy** has also been put forward in the field of human-robot interaction, where the relative merits of machine-directed versus human-directed changes in the relative autonomy of the robots have been debated (Cummings et al., 2010; Goodrich et al., 2007; Valero-Gomez et al., 2011).

The adaptive/adaptable automation concept certainly remains in the forefront of the thinking of designers of many highly automated complex systems (Ahlstrom et al., 2005; Inagaki, 2003; Miller & Parasuraman, 2007; Parasuraman et al., 2007; Valero-Gomez et al., 2011). Yet as we have discussed, there are many issues that must be addressed before viable systems can become effective or even feasible. Most importantly, these will depend on a continued and better understanding of the fundamentals of human attention, along with fascinating areas of human performance theory that have only recently received interest in the human factors domain—communication, cooperation, and trust.

9. DESIGNING FOR EFFECTIVE HUMAN-AUTOMATION INTERACTION

In the previous pages, we have identified a number of human performance issues that arise when users interact with automated systems. Many of these issues are prevalent in systems designed purely from a technology-centered perspective. In contrast, the past two decades of research on human-automation interaction have pointed to several solutions to these problems (Degani, 2004; Parasuraman, 2000; Sheridan, 2002; Sheridan & Parasuraman, 2006; Sethumadhavan, 2011). These solutions were identified implicitly in our earlier discussion of human-automation interaction problems. Many of these can be loosely grouped under the rubric of human centered automation (Billings, 1997). It should be noted that these solutions will not necessarily provide the optimal use of automation from the point of productivity or system performance, but should, if followed, provide for greater margins of safety, more satisfaction for the human user, and the least disruptive episodes of "manual recovery" in the instance of system failure.

9.1 Feedback

We saw previously that many cases of accidents and incidents in automated systems have occurred because human operators were provided poor or no feedback on automation states and behaviors (Norman, 1990). Accordingly, designers of automation should make efforts to display critical information regarding the current state of automation, changes in those states (e.g., a switch in automation levels), and the status of the process being monitored or controlled by the automation (e.g., the continuous variable that is sensed by the automated alarm). It should be noted that the type of feedback should carefully thought out; poorly presented or excessive feedback can be as bad as no feedback at all. In Chapter 4, we discussed some case studies of successful displays in the context of ecological interface design (Seppelt & Lee, 2007).

One approach to providing operators feedback to the operator is to use a **multi-modal** display, so as not to overload the main sensory channel that the operator uses, which is typically vision (see Chapter 7). Auditory channels can be considered, and there are examples of the use of auditory feedback to provide information on system state to enhance performance on primarily visual tasks (Ho & Spence, 2008). However, as auditory displays grow in sophistication with the advent of auditory "earcons", speech synthesizers, etc. (Baldwin, 2012, see Chapter 6), even the auditory channel can become crowded. As a result, a number of researchers have explored the utility of haptic or tactile displays as feedback channels (Sarter, 2007). For example, Sklar and Sarter (1999) showed that a tactile display worn of the wrist could provide information on FMS mode changes without disrupting primary flight performance, while improving alert detection.

9.2 Appropriate Levels and Stages of Automation

Integrating our discussion of trust/dependence with that of stages and levels of automation, we understand that "more automation" (e.g., more and later stages of higher levels within a stage; a higher degree of automation) may be a two edged sword. Typically a higher degree of automation will increase routine performance and/or decrease workload. (If neither of these is observed, the automation is clearly faulty from a human performance perspective.) But increasing degree is also likely to increase OOTLUF (by degrading situation awareness) and, as a consequence, degrade failure management. Thus, the improvements in workload and routine performance brought about by increasing degree of automation are offset by the loss of SA and failure response (Wickens, Li, et al., 2010). (This tradeoff is analogous to the changes in signal detection, brought about by increasing beta, on misses and false alarms.) Furthermore, these tradeoffs would appear

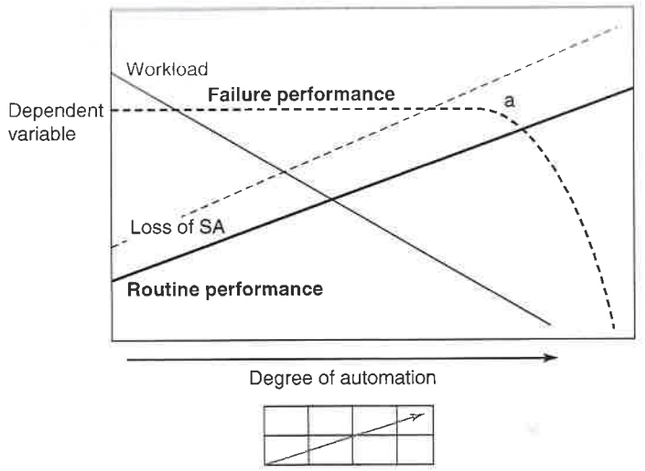

FIGURE 12.6 Hypothetical tradeoff between routine and failure performance, and between workload and the loss of situation awareness, as the degree of imperfect automation (stages and levels) is increased.

to be enhanced as automation reliability increases. The hypothetical tradeoff of these variables with DOA is shown in Figure 12.6.

If these performance, workload, and SA functions of degree of automation were predictable and reliable, they could then be employed to establish the optimum degree of automation to the extent that a designer could establish the relative weight to be assigned to improving performance (both routine and failure) and reducing workload. But this has proven to be a challenging task.

There are of course a few studies that have varied stages and/or levels while measuring some of these critical variables. A classic study was that described in Chapter 6 by Crocoll and Coury (1990), who contrasted a status display (stage 2 automation) with a command display (stage 3) and found that while the latter favored routine performance, the former favored failure-management performance. Analogous findings were obtained by Sarter and Schroeder (2001) when evaluating automation to prevent aircraft icing, contrasting either displays that showed the inference of where ice was building up (stage 2) or a command advisor that recommended maneuvers to recover from icing. A classic study examining the tradeoff across levels of automation was carried out by Endsley and Kiris (1995). They examined the effects of a driving decision aid in this study and observed that the optimal point on the tradeoff was at a mid-high level of automation, but not at the highest level.

Rovira et al. (2007) further examined the effects of different levels of automation reliability (60 percent and 80 percent) and three different levels of decision automation on performance impairment with imperfect decision automation. The performance cost of inaccurate decision advice was most pronounced at the highest level of automation (i.e. when a specific recommendation for an optimum decision was given) and when automation reliability was high.

Wickens, Li, et al. (2010) attempted to integrate in a meta-analysis the collective wisdom of these and several other studies that have varied the degree of imperfect automation, while assessing two or more of the four critical variables (performance under routine and failure mode situations, workload and situation awareness, as shown in figure 12.6) (e.g., Manzey, D., Reichenbach, J., & Onnasch, L., 2012; Sethumadhavan, 2009; Kaber, Onal, & Endsley, 2000). The results of the meta analysis revealed consistent trends for performance across studies: routine performance

improved and failure performance degraded as the degree of automation increased. However, the results for decreasing workload and decreasing situation awareness were less clear cut (in part given the lack of studies that assessed SA across different degrees of imperfect automation). One finding, however, bears particular note. Those studies in which higher degrees of automation were shown to have higher situation awareness were ones in which *both* routine and failure performance improved as degree of automation increased. We might infer that these were studies in which researchers played particular attention to effective display design, and transparency of feedback, a point we addressed in detail in the previous section.

In addition to the fact that good feedback can probably move the optimal point farther to the right of Figure 12.6, a case can also be made that, when the risks of imperfection and human/automation error are high, the optimum point should be moved more to the left (Parasuraman, Sheridan, & Wickens, 2000), but when the time pressure is extremely high, such that a human operator decision may not be able to be made in time (such as the decision to shut down a faulty engine on takeoff; Inagaki, 2003), the optimal point for an automation system to support the pilot in the decision should be moved farther to the right.

9.3 Designing for Human-Automation "Etiquette"

As discussed previously, trust plays an important role in determining the degree to which human operators depend on automated systems. Trust has both cognitive and affective properties. The latter takes on a more prominent role as automated systems increase in their "intelligence" and in their ability to interact with humans in ways that mimic human-human interaction—e.g., through voice and "face to face" communications. Nass and colleagues (Nass et al., 1995; Reeves & Nass, 1996) have shown that people often respond socially to computers in ways similar to how they interact normally with other people. Because some forms of automation can appear to be endowed with greater intelligence and with other human-like properties, therefore, it is important to ask whether they should be designed to act in socially appropriate ways with humans.

Our interaction with others is typically governed by rules that are implicitly understood and adhered to in most settings, whether formal or informal. Such etiquette, or adherence to an accepted but frequently implicit code of behavior between individuals in any social setting, may also be important for effective human-computer relations. Parasuraman and Miller (2004) showed that etiquette can influence the efficiency with which operators make diagnostic decisions when using automation. They tested participants on the MATB simulation with the aircraft automated fault management system that provided advisories on possible engine faults. In one condition the automation followed good etiquette, i.e. provided participants with a pre-warning and then waited for them to complete what they were doing being issuing an advisory. In a second condition, the automation displayed poor etiquette by not warning the participants and not waiting for them to finish what they were doing. Diagnostic accuracy was about 20 percent higher in the good etiquette condition than in the poor etiquette condition. Dorneich, Ververs, et al. (2012) reported similar benefits of good **automation etiquette** on multitasking performance. They designed an adaptive automation system that managed the flow of messages to the user by directing only high priority messages to the user when the user's workload was high and storing low-priority messages for later viewing, in a manner similar to a human assistant who follows good etiquette.

There are also aspects to etiquette other than knowing when and when not to interrupt. Grice (1975) described the behavioral practices that allow for acceptable and efficient interaction between people. For example, in conversation with another person we typically try to avoid being obscure or ambiguous if we want to communicate effectively. Hayes and Miller (2011) suggested

that automated systems should similarly avoid obscurity or ambiguity. In this view, automation that is designed to follow such agreed-upon axioms of etiquette tends to be accepted and liked by human operators.

9.4 Calibrating Operator Trust: Display Design and Training

As we have seen, poorly calibrated trust is a major contributor to inefficient human-automation interaction. Human users exhibit both under- and over-trust. Both can be remediated through attention to automation design and training. We address first the possible solutions to mistrust, and then to the challenges of over-trust and complacency.

9.4.1 MITIGATING MISTRUST The material in Section 7.2, by identifying the sources of mistrust in automation, has implicitly suggested solutions. For example, simplifying the complexity of automation functionality and/or making it more "transparent" to the user via good displays should reduce mistrust. So also should increased training of the human supervisor of the functioning of those algorithms. To guard against the mistrust of false-alarm prone alerts, we refer the reader back to the points made in Chapter 2 (Section 4.3) but wish to elaborate the readers on two of them: the issue of training and that of likelihood alerts.

First, with regard to training, it is necessary for users of alarm systems to realize that, in conditions in which system failures may be subtle yet catastrophic and early warnings are thus desirable, and in which the base rate of failures is quite low; then alarm false alarms are an inevitable consequence to be tolerated (Parasuraman et al., 1997).

Second, regarding displays, there is evidence that the alarm false alarm problem can be mitigated to an extent through the use of **likelihood displays** (Sorkin et al., 1988). Such displays provide two or more graded levels of certainty that a critical condition exists. In essence, such a concept allows the system to say "I'm not sure" rather than just blurting out a full alarm or nothing at all (and setting a risky criterion to avoid misses). As we learned in Chapter 2, allowing human signal detectors to express their confidence in "signal-present" at more than one level improves human detection performance. Similarly, allowing the alarm system a corresponding resolution in confidence provides a corresponding improvement in the sensitivity of the human and system together (Sorkin et al., 1988).

In a field study of a homeland security threat detection system, radiation portals at border crossings, Sanquist et al. (2008) showed that using a likelihood alarm display and a Bayesian analysis could reduce the false alarm problem. Monitors for detecting such radioactive sources (e.g., a "dirty bomb") that are currently deployed at border crossings are plagued by "nuisance alarms"—alarms that occur because of objects that are radioactive but are not true threats, such as fertilizer, pet litter, or irradiated fruit. Sanquist et al. first estimated the (very low) base rate of true threats (e.g., weapons-grade plutonium) and provided design criteria for increasing alarm positive predictive value (PPV)—the probability that, given an alarm, a true threat exists. They also showed that the PPV could be increased (and nuisance alarms reduced) by including cargo manifest information (e.g., whether a truck was carrying fertilizer) in the detection system algorithm. Finally, they showed that the use of a three-level likelihood display (e.g., "Pass: no material of concern"; "Alert: naturally occurring radiation material; and "Alarm: potential threat of radiation material") also substantially reduced false alerts.

9.4.2 MITIGATING OVER-TRUST AND COMPLACENCY In order to mitigate the trust calibration phenomena of over-trust, there is some evidence that providing automation reliability information can help users calibrate their trust and dependence on an automated combat identification system (Neyedli et al., 2011; Wang et al., 2009). As we suggested implicitly in Section 7.1.1, by

comparing first failure responses with subsequent responses to the failures of automation (e.g., Parasuraman & Molloy, 1996; Merlo et al., 2003; Manzey, D., Reichenbach, J., & Onnasch, L., 2012, Wickens et al., 2009), one of the best techniques is to get the "first failure out of the way" with training or practice on automation, before real-time use is undertaken. Hence, the "first failure" in that real-time use is now actually a "subsequent failure" after a certain amount of mistrust in the automation has been allowed to accumulate.

Importantly however, simply informing the automation user that a failure *could* occur (Skitka, Mosier, & Burdick 2000) is far less effective than is actually experiencing that failure, a conclusion echoing that regarding debiasing in decision making, discussed in Chapter 8, Section 9.1 (Larrick, 2006). This conclusion is highlighted by the findings of Bahner, Huper, and Manzey (2008), who examined whether experience with automation failures could reduce complacency. They tested two groups of participants in a process control simulation in which fault management automation provided advisories on system faults. One group was simply informed that the automation would work highly reliably, although not perfectly, and that they should verify each diagnosis before accepting it by checking the information sources pertaining to the diagnosis ("information group"). The other group received the same information but was additionally exposed to a few automation failures (incorrect diagnoses) during training ("experience group"). Following the work of Lee and Moray (1992), Bahner and colleagues found that experience with imperfect automation reduced overall trust and thereby the degree of complacency, which they measured by the number of information parameters checked prior to accepting a diagnosis. Participants in the "information group" sampled fewer information parameters than the "experience group." This finding indicates that training with exposure to automation failures can reduce complacency.

10. CONCLUSIONS

Automated systems, supporting or replacing all stages of human information processing, are found in all aspects of work and life—in manufacturing, power generation, health care, transportation, offices, homes, and in many other industries. In many such environments, automation has improved efficiency, enhanced safety, and reduced operator workload. At the same time, automation has also introduced new problems and changed the nature of cognitive work of human operators, which at times has lead to incidents and accidents. Several human performance issues have arisen because automated systems have often been designed from a technology-centered perspective. These include unbalanced mental workload, reduced situation awareness, and uncalibrated trust, both under-trust and over-trust. A number of approaches to designing for effective human-automation interaction are possible. These include using appropriate levels and stages of automation, reducing automation complexity, providing feedback, and training for calibrated trust. Adaptive/adaptable automation may also help in reducing some of the human performance costs of automation, although further work needs to be conducted on its practical feasibility as a design option.

Key Terms

adaptive automation *395*
adjustable autonomy *399*
automation bias *392*
automation dependence *388*
automation etiquette *402*
automation reliability *388*
automation surprises *387*
calibration curve *389*
change blindness *397*
complacency (over-trust) *388*
compliance *391*
cry wolf effect (under trust) *388*
first failure effect *391*
function allocation *396*
generation effect *390*
human-centered automation *378*
levels of automation *381*
likelihood displays *403*
Mental workload *397*
multi-modal *400*
out of the loop unfamiliarity *393*
reliance *391*
stages of automation *382*
trust *388*

EPILOGUE

Over the course of the 11 chapters that have addressed specific components of performance, a number of themes emerged that characterized findings and principles across more than a single chapter. By virtue of their repeated occurrences, these are themes that we believe are particularly important for understanding human performance strengths and limitations in the workplace. Our list below is not exhaustive, and we would be pleased if readers might like to augment this with thoughts of their own.

1. ***Working memory limitations.*** Repeatedly, we have noted that working memory is a very constraining limit in its own right (as the phone dialing example illustrates), but also such limits drive other processing constraints, and principles, such as the costs of having material to be compared separated in space and time. Effective use of working memory is effortful. Effort is a limited resource, and the human's natural tendency to conserve it can harm processes based on working memory, creating errors, delaying performance, or imposing cognitive workload.
2. ***The 2 C's: compatibility and confusion.*** Both of these **C** concepts made repeated appearances: compatibility in terms of display-control (stimulus-response), ecological, modality, and cognitive aspects. The key general point this raises is the interaction between stages of information processing. No stage can easily be treated in isolation because the mapping *between* them is so important.

   While compatibility has long been a well known principle in engineering psychology, the concept of similarity-based **confusion** does not enjoy such a rich history, yet is every bit as critical in characterizing human performance. If two things look, sound, or feel similar and could both occur in the same context, there is a high likelihood that the wrong one will be inappropriately perceived, its associated response triggered, or that they will be confused in working memory, sometimes with devastating consequences. Thus, where similarity is the bad here, *discriminability* represents the good.
3. ***Tradeoffs.*** As with the two C's, tradeoffs also have shown two manifestations. First, *people* often have a cognitive set that can trade off two variables or kinds of processes in human performance. For example, there are hits versus misses in signal detection (via beta), speed versus accuracy in many tasks, task A versus task B in time sharing, effort conservation versus accuracy in decision making and search, and balancing probability versus value in choice. Human performance theory is critical in helping to understand these strategic tradeoffs, what drives people along the tradeoff function, where they *should* operate versus where they do operate, and how to measure the quality of human performance across the function. In a sense, the receiver operating characteristic (ROC), the speed-accuracy operating characteristic (SAOC), and the performance operating characteristic (POC) offer explicit representations of such tradeoffs.