Abstract

Abstract

Adaptive automation refers to systems in which both the user and the system can initiate changes in the level of automation. The first adaptive automation systems were implemented in associate systems based on models of operator behavior and workload. Recently, however, systems have been developed that follow the neuroergonomics approach and use psychophysiological measures to trigger changes in the state of automation. Studies have shown that this approach can facilitate operator performance. Further, evidence is beginning to show that people not only think of adaptive systems as coworkers, they may even *expect* them to behave like humans. Consequently, adaptive automation creates new challenges for both users and designers that go beyond traditional ideas of human-computer interaction and system design.

*We humans have always been adept at dovetailing our minds and skills to the shape of our current tools and aids.  But when those tools and aids start dovetailing back – when our technologies actively, automatically, and continually tailor themselves to us just as we do to them – then the line between tool and human becomes flimsy indeed.*

 -- Andy Clark, *Natural-Born Cyborgs: Minds, Technologies and the Future of Human Intelligence* (p. 7)

Introduction

            Neuroergonomics has been described as the study of brain and behavior at work (Parasuraman, 2003). This emerging area focuses on current research and developments in the neuroscience of information processing and how that knowledge can be used to improve performance in real-world environments.  Parasuraman argues that an understanding of how the brain processes perceptual and cognitive information can lead to better designs for equipment, systems, and tasks by enabling a tighter match between task demands and the underlying brain processes. Ultimately, research in neuroergonomics can lead to safer and more efficient working conditions.

            Ironically, interest in neuroergonomics evolved from research surrounding how operators interact with a form of technology designed to make work and our lives easier – automation.  In general, automation can be thought of as a machine agent capable of carrying out functions normally performed by a human (Parasuraman & Riley, 1997). For example, the automatic transmission in an automobile allocates the tasks of depressing the clutch, shifting gears, and releasing the clutch to the vehicle. Automated machines and systems are intended and designed to reduce task demands and workload. Further, they allow individuals to increase their span of operation or control, perform functions that are beyond their normal abilities, maintain performance for longer periods of time, and perform fewer mundane activities. Automation can also help reduce human error and increase safety. The irony behind automation arises from a growing body of research demonstrating that automated systems often *increase*workload and create *unsafe* working conditions.

In his book, *Taming HAL: Designing interfaces beyond 2001*, Degani (2004) relates the story of an airline captain and crew performing the last test flight with a new aircraft. This was to be the second such test that day and the captain, feeling rather tired, requested that the copilot fly the aircraft. The test plan required a rapid take-off, followed by engaging the autopilot, simulating an engine failure by reducing power to the left engine, and then turning off the left hydraulic system. The test flight started out just fine. Four seconds into the flight, however, the aircraft was pitched about 4 degrees higher than normal, but the captain continued with the test plan and attempted to engage the autopilot. Unfortunately, the autopilot did not engage. After a few more presses of the autopilot button, the control panel display indicated that the system had engaged (although in reality, the autopilot had not assumed control). The aircraft was still pitched too high and was beginning to lose speed. The captain apparently did not notice these conditions and continued with the next steps requiring power reduction to the left engine and shutting down the hydraulic system.

            The aircraft was now flying on one engine with increasing attitude and decreasing speed. Moreover, the attitude was so steep that the system intentionally withdrew autopilot mode information from its display.  Suddenly, the autopilot engaged and assumed the attitude capture mode to take the aircraft to the preprogrammed setting of 2,000 ft., but this information was not presented on the autopilot display.  The autopilot initially began lowering the nose, but then reversed course. The attitude began to pitch up again and airspeed continued to fall. When the captain finally turned his attention from the hydraulic system back to the instrument panel, the aircraft was less than 1,500 ft above ground, pitched up 30 degrees, with airspeed dropping to about 100 knots. The captain then had to compete with the envelope protection system for control of the aircraft.  He attempted to bring the nose down and then realized he had to *reduce* power to the right engine in order to undo a worsening corkscrew effect produced by the simulated left engine failure initiated earlier. Although he was able to bring the attitude back down to zero, the loss of airspeed coupled with the simulated left engine failure had the aircraft in a 90-degree roll.  The airspeed soon picked up and the captain managed to raise the left wing, but by this time the aircraft was only 600 ft. above ground.  Four seconds later the aircraft crashed into the ground killing all on board.

            Degani (2004) discusses several factors that contributed to this crash. First, no one knows why the autopilot’s attitude was preprogrammed for 2,000 ft, but it is possible that the pilot never entered the correct value of 10,000 ft. Second, although the pilot tried several times to engage the autopilot, he did not realize that the system’s logic would override his requests because his copilot’s attempts to bring the nose down were canceling his requests. Third, there was a previously undetected flaw in the autopilot’s logic. The autopilot calculated the rate of climb needed to reach 2,000 ft when *both* engines were powered up, but did not recalculate the rate after the left engine had been powered down. Thus, the autopilot continued to demand the power it needed to reach the preprogrammed altitude despite that the aircraft was losing speed. Last, no one knows why the pilot did not disengage the autopilot when the aircraft continued to increase its attitude. Degani suggests that pilots who have substantial experience with autopilot systems may place too much trust in them. Thus, it is possible that assumptions regarding the reliability of the autopilot coupled with the absence of mode information on the display left the captain without any information or reason to question the status of the autopilot.

            This incident clearly highlights the complexity and problems that can be introduced by automation. Unfortunately, it is not a unique occurrence. Degani (2004) describes similar accounts of difficulties encountered with other automated systems including cruise control in automobiles and blood pressure devices.

Research on human interaction with automation has shown that it does not always make the job easier. Instead, it *changes* the nature of work. More specifically, automation changes the way activities are distributed or carried out and can therefore introduce new and different types of problems (Woods, 1996).  Automation can also lead to different types of errors because operator goals may be incongruent with the goals of systems and subsystems (Sarter & Woods, 1995; Wiener, 1989). Woods (1996) argues further that in systems where subcomponents are tightly coupled, problems may propagate more quickly and be more difficult to isolate. In addition, highly automated systems leave fewer activities for individuals to perform. Consequently, the operator becomes a more passive monitor instead of an active participant. Parasuraman, Mouloua, Molloy, and Hilburn (1996) have shown that this shift from performing tasks to monitoring automated systems can actually inhibit one’s ability to detect critical signals or warning conditions. Further, an operator’s manual skills can begin to deteriorate in the presence of long periods of automation (Wickens, 1992).

Adaptive Automation

            Given the problems associated with automation noted above, researchers and developers have begun to turn their attention to alternative methods for implementing automated systems. *Adaptive automation* is one such method that has been proposed to address some of the shortcomings of traditional automation. In adaptive automation, the level of automation or the number of systems operating under automation can be modified in real time. In addition, changes in the state of automation can be initiated by *either* the human or the system (Hancock & Chignell, 1987; Rouse, 1976; Scerbo, 1996). Consequently, adaptive automation enables the level or modes of automation to be tied more closely to operator needs at any given moment (Parasuraman et al., 1992).

            Adaptive automation systems can be described as either *adaptable*or *adaptive.*Scerbo (2001) has described a taxonomy of adaptive technology. One dimension of this taxonomy concerns the underlying source of flexibility in the system, i.e., whether the information displayed or the functions themselves are flexible. A second dimension addresses how the changes are invoked. In adaptable systems, changes among presentation modes or in the allocation of functions are initiated by the user. By contrast, in adaptive systems both the user and the system can initiate changes in the state of the system.

            The distinction between adaptable and adaptive technology can also be described with respect to authority and autonomy. Sheridan and Verplank (1978) have described several levels of automation that range from completely manual, to semiautomatic, to fully automatic.  As the level of automation increases, systems take on more authority and autonomy. At the lower levels of automation, systems may offer suggestions to the user. The user can either veto or accept the suggestions and then implement the action. At moderate levels, the system may have the autonomy to carry out the suggested actions once accepted by the user. At higher levels, the system may decide on a course of action, implement the decision, and merely inform the user. With respect to Scerbo’s (2001) taxonomy, adaptable systems are those in which the operator maintains authority over invoking changes in the state of the automation (i.e., they reflect a superordinate-subordinate relationship between the operator and the system). In adaptive systems, on the other hand, authority over invocation is shared. Both the operator and the system can initiate changes in state of the automation.

            There has been some debate over who should have control over changes among modes of operation.  Some argue that operators should always have authority over the system because they are ultimately responsible for the behavior of the system. In addition, it is possible that operators may be more efficient at managing resources when they can control changes in the state of automation (Billings & Woods, 1994; Malin & Schreckenghost, 1992). Many of these arguments are based on work with life critical systems in which safe operation is of utmost concern.  However, it is not clear that strict operator authority over changes among automation modes is always warranted. There may be times when the operator is not the best judge of when automation is needed. For example, changes in automation may be needed at the precise moment the operator is too busy to make those changes (Weiner, 1989). Further, Inagaki, Takae and Moray (1999) have shown mathematically that the best piloting decisions concerning whether to abort a take-off are not those where either the human or the avionics maintain full control. Instead, the best decisions are made when the pilot and the automation share control.

      Scerbo (1996) has argued that in some hazardous situations where the operator is vulnerable, it would be extremely important for the system to have authority to invoke automation. If lives are at stake or the system is in jeopardy, allowing the system to intervene and circumvent the threat or minimize the potential damage would be paramount. For example, it is not uncommon for many of today's fighter pilots to sustain G forces high enough to render them unconscious for periods of up to 12 seconds. Conditions such as these make a strong case for system-initiated invocation of automation. An example of one such adaptive automation system is the Ground Collision-Avoidance System (GCAS) developed and tested on the F-16D (Scott, 1999). The system assesses both internal and external sources of information and calculates the time it will take until the aircraft breaks through a pilot determined minimum altitude. The system issues a warning to the pilot. If no action is taken, an audio “fly up” warning is then presented and the system takes control of the aircraft. When the system has maneuvered the aircraft out of the way of the terrain, it returns control of the aircraft to the pilot with the message, “You got it”. The intervention is designed to right the aircraft quicker than any human pilot can respond. Indeed, test pilots who were given the authority to override GCAS eventually conceded control to the adaptive system.

Adaptive Strategies

            There are several strategies by which adaptive automation can be implemented (Morrison & Gluckman, 1994; Rouse & Rouse, 1983). One set of strategies addresses system functionality. For instance, entire tasks can be allocated to either the system or the operator, or a specific task can be partitioned so that the system and operator each share responsibility for unique portions of the task.  Alternatively, a task could be transformed to a different format to make it easier (or more challenging) for the operator to perform.

            A second set of strategies concerns the triggering mechanism for shifting among modes or levels of automation (Parasuraman et al., 1992; Scerbo, Freeman, & Mikulka, 2003). One approach relies on goal-based strategies. Specifically, changes among modes or levels of automation are triggered by a set of criteria or external events.  Thus, the system might invoke the automatic mode only during specific tasks or when if it detects an emergency situation. Another approach would be to use real-time measures of operator performance to invoke the changes in automation. A third approach uses models of operator performance or workload to drive the adaptive logic (Hancock & Chignell, 1987; Rouse, Geddes & Curry, 1987, 1988). For example, a system could estimate current and future states of an operator’s activities, intentions, resources, and performance.  Information about the operator, the system, and the outside world could then be interpreted with respect to the operator’s goals and current actions to determine the need for adaptive aiding. Finally, psychophysiological measures that reflect operator workload can also be used to trigger changes among modes.

Examples of Adaptive Automation Systems

            Adaptive automation has its beginnings in artificial intelligence.  In the 1970s, efforts were directed toward developing adaptive aids to help allocate tasks between humans and computers. By the 1980s, researchers began developing adaptive interfaces. For instance, Wilensky, Arens, and Chin (1984) developed the UNIX Consultant (UC) to provide general information about UNIX, procedural information about executing UNIX commands, as well as debugging information. The UC could analyze user queries, deduce the user goals, monitor the user’s interaction history, and present the system’s response.

*Associate systems.*  Adaptive aiding concepts were applied in a more comprehensive manner in the Defense Advanced Research Projects Agency (DARPA) Pilot’s Associate program (Hammer & Small, 1995). The goal of the program was to use intelligent systems to provide pilots with the appropriate information, in the proper format, at the right time. The Pilot’s Associate could monitor and assess the status of its own systems as well as events in the external environment. The information could then be evaluated and presented to the pilot. The Pilot’s Associate could also suggest actions for the pilot to take. Thus, the system was designed to function as an assistant for the pilot.

            In the 1990s, the U.S. Army attempted to take this associate concept further in its Rotorcraft Pilot’s Associate (RPA) program. The goal was to create an associate that could serve as a “junior crew member” (Miller & Hannen, 1999).  A major component of the RPA is the Cognitive Decision Aiding System (CDAS) which is responsible for detecting and organizing incoming data, assessing the internal information regarding the status of the aircraft, assessing external information about target and mission status, and feeding this information into a series of planning and decision-making modules. The Cockpit Information Manager (CIM) is the adaptive automation system for the CDAS. The CIM is designed to make inferences about current and impending activities for the crew, allocate tasks among crew members as well as the aircraft, and reconfigure cockpit displays to support the ability of the “crew-automation team” to execute those activities (see Figure 1). The CIM monitors crew activities and external events and matches them against a database of tasks to generate inferences about crew intentions. The CIM uses this information to make decisions about allocating tasks, prioritizing information to be presented on limited display spaces, locating pop-up windows, adding or removing appropriate symbology from displays, and adjusting the amount of detail to be presented in displays. Perhaps most important, the CIM includes a separate display that allows crew members and the system to coordinate the task allocation process and communicate their intentions (located above the center display in Figure 1). The need for communication among members is important for highly functioning human teams and, as it turned out, was essential for user acceptance of the RPA. Evaluations from a sample of pilots indicated that the RPA often provided the right information at the right time. Miller and Hannen reported that in the initial tests, no pilot chose to turn off the RPA.

            The RPA was an ambitious attempt to create an adaptive automation system that would function as a team member. There are several characteristics of this effort that are particularly noteworthy. First, a great deal of the intelligence inherent in the system was designed to anticipate user needs and be proactive about reconfiguring displays and allocating tasks. Second, both the users and the system could communicate their plans and intentions, thereby reducing the need to decipher what the system was doing and why it was doing it.  Third, unlike many other adaptive automation systems, the RPA was designed to support the simultaneous activities of multiple users.

            Although the RPA is a significant demonstration of adaptive automation, it was not designed from the neuroergonomics perspective. It is true that the a good deal of knowledge about cognitive processing related to decision-making, information representation, task scheduling and task sharing, was needed to create the RPA, but the system was not built around knowledge of brain functioning.

*Brain-based systems.* An example of adaptive automation that follows the neuroergonomics approach can be found in systems that use psychophysiological indices to trigger changes in the automation. There are many psychophysiological indices that reflect underlying cognitive activity, arousal levels, and external task demands. Some of these include cardiovascular measures (e.g., heart rate, heart rate variability), respiration, galvanic skin response (GSR), ocular motor activity, speech, as well as those that reflect cortical activity such the electroencephalogram (EEG), event-related potentials (ERPs) derived from EEG signals to stimulus presentations, functional magnetic resonance imaging (fMRI), and near infrared spectrometry (NIRS) that measures changes in oxygenated and deoxygenated hemoglobin (see Byrne & Parasuraman 1996 for a review).  One of the most important advantages to brain-based systems for adaptive automation is that they provide a continuous measure of activity in the presence or absence of overt behavioral responses (Byrne & Parasuraman 1996; Scerbo et al. 2001).

The first brain-based adaptive system was developed by Pope, Bogart and Bartolome (1995).  Their system uses an index of task engagement based upon ratios of EEG power bands (alpha, beta, theta, etc.). The EEG signals are recorded from several locations on the scalp and are sent to a LabView Virtual Instrument that determines the power in each band for all recording sites and then calculates the engagement index used to change a tracking task between automatic and manual modes. The system recalculates the engagement index every two seconds and changes the task mode if necessary. Pope and his colleagues studied several different engagement indices under both negative and positive feedback contingencies. They argued that under negative feedback the system should switch modes more frequently in order to maintain a stable level of engagement. By contrast, under positive feedback the system should be driven to extreme levels and remain there longer (i.e., fewer switches between modes). Moreover, differences in the frequency of task mode switches obtained under positive and negative feedback conditions should provide information about the sensitivity of various engagement indices. Pope et al. found that the engagement index based on the ratio of beta/(alpha +theta) proved to be the most sensitive to differences between positive and negative feedback.

The study by Pope et al. (1995) showed that their system could be used to evaluate candidate engagement indices. Freeman, Milkulka, Prinzel and Scerbo (1999) expanded upon this work and studied the operation of the system in an adaptive context. They asked individuals to perform the compensatory tracking, resource management, and system monitoring tasks from the Multi-Task Attribute Battery (MAT; Comstock & Arnegard, 1991). Figure 1 shows a participant performing the MAT task while EEG signals are being recorded. In their study, all tasks remained in automatic mode except the tracking task which shifted between automatic and manual modes. They also examined performance under both negative and positive feedback conditions. Under negative feedback, the tracking task was switched to or maintained in automatic mode when the index increased above a pre-established baseline reflecting high engagement. By contrast, the tracking task was switched to or maintained in manual mode when the index decreased below the baseline reflecting low engagement. The opposite schedule of task changes occurred under the positive feedback conditions. Freeman and his colleagues argued that if the system could moderate workload, better tracking performance should be observed under negative as compared to positive feedback conditions. Their results confirmed this prediction. In subsequent studies, similar findings were found when individuals performed the task over much longer intervals and under conditions of high and low task load (see Scerbo et al., 2003).

            More recently, St. John, Kobus, Morrison, and Schmorrow (2004) have described a new DARPA program aimed at enhancing an operator’s effectiveness by managing the presentation of information and cognitive processing capacity through cognitive augmentation derived from psychophysiological measures. The goal of the program is to develop systems that can detect an individual’s cognitive state and then manipulate task parameters to overcome perceptual, attentional, and working memory bottlenecks. Unlike the system described by Pope et al. (1995) that relies on a single psychophysiological measure, EEG, the augmented cognition systems use multiple measures including NIRS, GSR, body posture, as well as EEG. The physiological measures are integrated to form “gauges” that reflect constructs such as effort, arousal, attention and workload.  Performance thresholds are established for each gauge to trigger mitigation strategies for modifying the task. Some of these mitigation strategies include switching between verbal and spatial information formats, reprioritizing or rescheduling tasks, or changing the level of display detail.

Wilson and Russell (2003; 2004) report on their experiences with an augmented cognitive system designed to moderate workload. Operators were asked to perform a target identification task with a simulated uninhabited combat air vehicle under different levels of workload. They recorded EEG from six sites as well as heart, blink, and respiration rate. In their system, the physiological data were analyzed by an artificial neural network (ANN). The ANN was trained to distinguish between high and low levels of operator workload in real time. The output from the ANN was used to trigger changes in the task to moderate workload. Comparisons among adaptive aiding, no aid, or random aiding revealed some performance benefits and lower ratings of subjective workload for the adaptive aiding condition under the more difficult condition.

Taken together, the findings from these studies suggest that it is indeed possible to obtain indices of one’s brain activity and use that information to drive an adaptive automation system to improve performance and moderate workload. There are, however, still many critical conceptual and technical issues (e.g., making the recording equipment less obtrusive and obtaining reliable signals in noisy environments) that must be overcome before systems such as these can move from the laboratory to the field (Scerbo et al., 2001).

Further, many issues still remain surrounding the sensitivity and diagnosticity of psychophysiological measures, in general. There is a fundamental assumption that psychophysiological measures provide a reliable and valid index of underlying constructs such as arousal or attention. In addition, variations in task parameters that affect those constructs must also be reflected in the measures (Scerbo et al. 2001). In fact, Veltman and Jansen (2004) have recently argued that there is no direct relation between information load and physiological measures or state estimators because an increase in task difficulty does not necessarily result in a physiological response. According to their model, perceptions of actual performance are compared to performance requirements. If attempts to eliminate the difference between perceived and required levels of performance are unsuccessful, one may need to increase mental effort or change the task goals. Both actions have consequences.  Investing more effort can be fatiguing and result in poorer performance. Likewise, changing task goals (e.g., slowing down, skipping low priority tasks, etc.) can also result in poorer performance. They suggest that in laboratory experiments, it is not unusual for individuals to compensate for increases in demand by changing task goals because there are no serious consequences to this strategy. However, in operational environments, where the consequences are real and operators are highly motivated, changing task goals may not be an option. Thus, they are much more likely to invest to effort needed to meet the required levels of performance. Consequently, Veltman and Jansen contend that physiological measures can only be valid and reliable in an adaptive automation environment if they are sensitive to information about task difficulty, operator output, the environmental context, and stressors.

Another criticism of current brain-based adaptive automation systems is that they are primarily reactive. Changes in external events or brain activity must be recorded and analyzed before any instructions can be sent to modify the automation. All of this takes time and even with short delays, the system must still wait for a change in events to react. Recently, however, Forsythe (in press) has described a brain-based system that also incorporates a cognitive model of the operator. The system is being developed by DaimlerChrysler through the DARPA Augmented Cognition program to support driver behavior. Information is recorded from the automobile (e.g., steering wheel angle, lateral acceleration) as well as the operator (e.g., head turning, postural adjustments, and vocalizations) and combined with EEG signals to generate inferences about workload levels corresponding to different driving situations.  In this regard, the system is a hybrid of brain-based and operator modeling approaches to adaptive automation and can be more proactive than current adaptive systems that rely solely on psychophysiological measures.

Workload and Situation Awareness

*Workload.* One of the arguments for developing adaptive automation is that this approach can moderate operator workload.  Most of the research to date has assessed workload through primary task performance or physiological indices (see above). Kaber and Riley (1999), however, conducted an experiment using both primary and secondary task measures. They had their participants perform a simulated radar task where the object was to eliminate targets before they reached the center of the display or collided with one another.  During manual control, the participants were required to assess the situation on the display, make decisions about which targets to eliminate, and implement those decisions. During a shared condition, the participant and the computer could each perform the situation assessment task. The computer scheduled and implemented the actions, but the operator had the ability to override the computer’s plans. The participants were also asked to perform a secondary task requiring them to monitor the movements of a pointer and correct any deviations outside of an ideal range.  Performance on the secondary task was used to invoke the automation on the primary task. For half of the participants, the computer suggested changes between automatic or manual operation of the primary task and for the remaining participants, those changes were mandated.

            Kaber and Riley (1999) found that shared control resulted in better performance than manual control on the primary task. However, the results showed that mandating the use of automation also bolstered performance during periods of manual operation. Regarding the secondary task, when use of automation was mandated, workload was lower during periods of automation; however, under periods of manual control, workload levels actually increased and were similar to those seen when its use was suggested.  These results show that authority over invoking changes between modes had differential effects on workload during periods of manual and automated operation. Specifically, Kaber and Riley (1999) found that the requirement to “consider” computer suggestions to invoke automation led to higher levels of workload during periods of shared/automated control than when those decisions were dictated by the computer.

*Situation awareness.* Thus far, there have been few attempts to study the effects of adaptive automation on situation awareness (SA). Endsley (1995) describes SA as the ability to perceive elements in the environment, understand their meaning, and to make projections about their status in the near future.  One might assume that efforts to moderate workload through adaptive automation would lead to enhanced SA; however, that relationship has yet to be demonstrated empirically. In fact, within an adaptive paradigm periods of high automation could lead to poor SA and make returning to manual operations more difficult. The findings of Kaber and Riley (1999) regarding secondary task performance described above support this notion.

            Recently, Bailey and his colleagues (2003) examined the effects of a brain-based adaptive automation system on SA. The participants were given a self-assessment measure of complacency toward automation (i.e., the propensity to become reliant on automation; see Singh, Molloy, & Parasuraman, 1993) and separated into groups who scored either high or low on the measure. The participants performed a modified version of the MAT battery that included a number of digital and analog displays (e.g., vertical speed indicator, GPS heading, oil pressure, and auto pilot on/off) used to assess SA. Participants were asked to perform the compensatory tracking task during manual mode and to monitor that display during automatic mode.  Half of the participants in each complacency potential group were assigned to either an adaptive or yoke control condition. In the adaptive condition, Bailey et al. used the system modified by Freeman et al. (1999) to derive an EEG-based engagement index to control the task mode switches. In the other condition, each participant was yoked to one of the individuals in the adaptive condition and received the same pattern of task mode switches; however, their own EEG had no effect on system operation. All participants performed three 15-minute trials and at the end of each trial, the computer monitor went blank and the experimenter asked the participants to report the current values for a sample of five displays. Participants’ reports for each display were then compared to the actual values to provide a measure of SA (Endsley, 2000).

Bailey and his colleagues (2003) found that the effects of the adaptive and yoke conditions were moderated by complacency potential. Specifically, for individuals in the yoke control conditions, those who were high as compared to low in complacency potential had much lower levels of SA. On the other hand, there was no difference in SA scores for high and low complacency individuals in the adaptive conditions. More important, the SA scores for both high and low complacency individuals were significantly higher than those of the low complacency participants in the yoke control condition. The authors argued that a brain-based adaptive automation system could ameliorate the effects of complacency by increasing available attentional capacity and in turn, improving SA.

Human-Computer Etiquette

            Recently, there has been interest in the merits of an etiquette for human-computer interaction. Miller (2002) describes etiquette as a set of prescribed and proscribed behaviors that permit meaning and intent to be ascribed to actions. Etiquette serves to make social interactions more cooperative and polite. Importantly, rules of etiquette allow one of form expectations regarding the behaviors of others. In fact, Nass, Moon, and Carney (1999) have shown that people adopt many of the same social conventions used in human-human interactions when they interact with computers. Moreover, they also expect computers to adhere to those same conventions when computers interact with users.

            Miller (2004) argues that when humans interact with systems that incorporate intelligent agents they may *expect* those agents to conform to accepted rules of etiquette. However, the norms may be implicit and contextually dependent: what is acceptable for one application may violate expectations in another. Thus, there may be a need to understand the rules under which computers should behave and be more polite.

            Miller (2004) also claims that users ascribe expectations regarding human etiquette to their interactions with adaptive automation. In their work with the RPA, Miller and Hannen (1999) observed that much of the dialog between team members in a two-seat aircraft was focused on communicating plans and intentions. They reasoned that any automated assistant would need to communicate in a similar manner to be accepted as a “team” player. Consequently, the CIM described earlier was designed to allow users and the system to communicate in a conventionally accepted manner.

             The benefits of adopting a human-computer etiquette are described by Parasuraman and Miller (2004) in a study of human-automation interactions. In particular, they focused on interruptions. In their study, participants were asked to perform the tracking and fuel resource management tasks from the MAT battery. A third task required participants to interact with an automated system that monitored engine parameters, detected potential failures, and offered advice on how to diagnose faults. The automation support was implemented in two ways. Under the “patient” condition, the automated system would withhold advice if the user was in the act of diagnosing the engines or provide a warning, wait five seconds, and then offer advice if it determined the user was not interacting with the engines. By contrast, under the “impatient” condition the automated system offered its advice without warning while the user was performing the diagnosis. Parasurman and Miller referred to the patient and impatient automation as examples of good and poor etiquette, respectively. In addition, they examined two levels of system reliability. Under low and high reliability, the advice was correct 60 and 80 percent of the time, respectively.

            As might be expected, performance was better under high as opposed to low reliability. Further, Parasuraman and Miller (2004) found that when the automated system functioned under the good etiquette condition, operators were better able to diagnose engine faults regardless of reliability level. In addition, overall levels of trust in the automated system were much higher under good etiquette within the same reliability conditions. Thus, “rude” behavior made the system seem less trustworthy irrespective of reliability level. Several participants commented that they disliked being interrupted. The authors argued that systems designed to conform to rules of etiquette may enhance performance beyond what might be expected from system reliability and may even compensate for lower levels of reliability.

            Parasuraman and Miller’s (2004) findings were obtained with a high criticality simulated system; however, the rules of etiquette (or interruptions) may be equally important for business or home applications.   In a recent study, Bubb-Lewis and Scerbo (2002) examined the effects of different levels of communication on task performance with a simulated adaptive interface. Specifically, participants worked with a computer “partner” to solve problems (e.g., determining the shortest mileage between two cites or estimating gasoline consumption for a trip) using a commercial travel planning software package.  In their study, the computer partner was actually a confederate in another room who followed a strict set of rules regarding how and when to intervene to help complete a task for the participant. In addition, they studied four different modes of communication that differed in the level of restriction ranging from context sensitive natural language to no communication at all.  The results showed that as restrictions on communication increased, participants were less able to complete their tasks, which in turn, caused the computer intervene more often to complete the tasks. This increase in interventions also led the participants to rate their interactions with the computer partner more negatively.  Thus, these findings suggest that even for less critical systems, poor etiquette makes a poor impression. Apparently, no one likes a show-off even if it is the computer.

Living with Adaptive Automation

            Adaptive automation is also beginning to find its way into commercial and more common technologies. Some examples include adaptive cruise control found on several high-end automobiles and “smart homes” that control electrical and heating systems to conform to user preferences.

Recently, Mozer (2004) described his experiences living in an adaptive home of his own creation. The home was designed to regulate air and water temperature and lighting.  The automation monitors the inhabitant’s activities and makes inferences about the inhabitant’s behavior, predicts future needs, and adjusts the temperature or lighting accordingly. When the automation fails to meet the user’s expectations, the user can set the controls manually.

            The heart of the adaptive home is the adaptive control of home environment (ACHE) and functions to balance two goals: user desires and energy conservation. Because these two goals can conflict with one another, the system uses a reinforcement learning algorithm to establish an optimal control policy. With respect to lighting, the ACHE controls multiple, independent light fixtures, each with multiple levels of intensity (see Figure 3). The ACHE encompasses a learning controller that selects light settings based on current states. The controller receives information about an event change that is moderated by a cost evaluator. A state estimator generates high-level information about inhabitant patterns and integrates it with output from an occupancy model as well as information regarding levels of natural light available to make decisions about changes in the control settings. The state estimator also receives input from an anticipator module that uses neural nets to predict which zones are likely to be inhabited within the next two seconds.  Thus, if the inhabitant is moving within the home, the ACHE can anticipate the route and adjust the lights before he arrives at his destination. Mozer (2004) recorded the energy costs and as well as costs of discomfort (i.e., incorrect predictions and control settings) for a month and found that both decreased and converged within about 24 days.

            Mozer (2004) had some intriguing observations about his experiences living in the adaptive house. First, he found that he generated a mental model of the ACHE’s model of his activities.  Thus, he knew that if he were to work late at the office, the “house” would be expecting him home at the usual time and he often felt compelled to return home! Further, he admitted that he made a conscious effort to be more consistent in his activities. He developed a meta-awareness of his occupancy patterns and recognized that as he made his behavior more regular, it facilitated the operation of the ACHE, which in turn, helped it to save energy and maximize his comfort. In fact, Mozer claimed, “the ACHE trains the inhabitant, just as the inhabitant trains the ACHE” (p. 293).

            Mozer (2004) also discovered the value of communication. At one point, he noticed a bug in the hardware and modified the system to broadcast a warning message throughout the house to reset the system. After the hardware problem had been addressed, however, he retained the warning message because it provided useful information about how his time was being spent. He argued that there were other situations where the user could benefit from being told about consequences of manual overrides.

Conclusion

The development of adaptive automation represents a qualitative leap in the evolution of technology. Users of adaptive automation will be faced with systems that differ significantly from the automated technology of today.  These systems will be much more complex from both the users’ and designers’ perspective. Scerbo (1996) argued that adaptive automation systems will need time to learn about users and users will need time to understand the automation. In the Mozer’s (2004) case, he and his home needed almost a month to adjust to one another.  Further, users may find that adaptive systems are less predictable due to the variability and inconsistencies of their *own*behavior.  Thus, users are less likely to think of these systems as tools, machines, or even traditional computer programs. As Mozer (2004) indicated, he soon began to think about how his adaptive home would respond to his behavior.  Others have suggested that interacting with adaptive systems is more like interacting with a teammate or coworker (Hammer & Small, 1995; Miller & Hannen, 1999; Scerbo, 1994).

The challenges facing designers of adaptive systems are significant. Current methods in system analysis, design, and evaluation fall short of what is needed to create systems that have the authority and autonomy to swap tasks and information with their users. These systems require developers to be knowledgeable about task sharing, methods for communicating goals and intentions, and even assessment of operator states of mind.  In fact, Scerbo (1996) has argued that researchers and designers of adaptive technology need to understand the social, organizational, and personality issues that impact communication and teamwork among humans to create more effective adaptive systems. In this regard, Miller’s (2004) ideas regarding human-computer etiquette may be paramount to the development of successful adaptive systems.

            Thus far, most of the adaptive automation systems that have been developed address life critical activities where the key concerns surround the safety of the operator, the system itself, and recipients of the system’s services. However, the technology has also been applied in other contexts where the consequences of human error are less severe (e.g., Mozer’s adaptive house). Other potential applications might include a personal assistant, butler, tutor, secretary, or receptionist. Moreover, adaptive automation could be particularly useful when incorporated in systems aimed at training and skill development as well as entertainment.

To date, most of the adaptive automation systems that have been developed were designed to maximize the user-system performance of a single user. Thus, they are user independent (i.e., designed to improve the performance of any operator).  However, overall user-system performance is likely to be improved further if the system is capable of learning and adjusting to the behavioral patterns of its user as was shown by Mozer (2004). Although building systems capable of becoming more user-specific might seem like a logical next step, that approach would introduce a new and significant challenge for designers of adaptive automation – addressing the unique needs of multiple users. The ability of Mozer’s house to successfully adapt to his routines is due in large part to his being the only inhabitant. One can imagine the challenge faced by an adaptive system trying to accommodate the wishes of two people who want the temperature set at different levels.

The problem of accommodating multiple users is not unique to adaptive automation. In fact, the challenge arises from a fundamental aspect of humanity. People are social creatures and as such, they work in teams, groups, and organizations. Moreover, they can be co-located or distributed around the world and networked together. Developers of collaborative meeting and engineering software realize that one cannot optimize the individual human-computer interface at the expense of interfaces that support team and collaborative activities.  Consequently, even systems designed to work more efficiently based on knowledge of brain functions must ultimately take into consideration groups of people. Thus, the next great challenge for the neuroergonomics approach may lie with an understanding of how brain activity of multiple operators in social situations can improve the organizational work environment.

Benefits of Run Time Dynamic Adaptive Automation:

* Cross Browser/OS testing independent of a user's machine
* Testing against older versions of browsers
* Real mobile device testing for mobile web and native apps, both iOS & Android
* Visual Validation for softer "look and feel" testing
* Ability to define test scripts in plain English and translate to any Programing Language code
* Test result reports including screenshots of failed test cases
* Execution from Windows & OSX or Linux desktops
* Execution from CI platforms such as Jenkins & Bamboo

Introduction to Testing:

Cross Browser Testing Automation:

Cross Device Testing Automation:

There are many automation tools that allows you to run application on multiple platforms and saves a lot of time.

While these automation tools are life-saving but still manual intervention is required because these tools cannot test the usability and accessibility of applications.

Emulator Software: to perform cross platform and environment testing, emulator need to be virtualization. Virtual machine need to be created with different environmental combinations as well as emulator are used to check behavior of applications.

Introduction to Automation Testing:

Introduction to Storage Testing:

Introduction to Networking:

Introduction to Cloud:

Introduction to Cloud Storage:

Adaptive Software Automation with Networked Cloud Computing.

# **What Is the Adaptive Network?**

[Tweet](https://twitter.com/share?url=https://www.ciena.com/insights/what-is/What-Is-the-Adaptive-Network.html&text=What%20Is%20the%20Adaptive%20Network?)

[Share](https://facebook.com/sharer/sharer.php?u=https%3A%2F%2Fwww.ciena.com%2Finsights%2Fwhat-is%2FWhat-Is-the-Adaptive-Network.html)

[+1](https://plus.google.com/share?url=https%3A%2F%2Fwww.ciena.com%2Finsights%2Fwhat-is%2FWhat-Is-the-Adaptive-Network.html)

[Share](https://www.linkedin.com/shareArticle?mini=true&url=https%3A%2F%2Fwww.ciena.com%2Finsights%2Fwhat-is%2FWhat-Is-the-Adaptive-Network.html)

[E-mail](mailto:?subject=The%20Adaptive%20Network%20is%20a%20new%20approach%20that%20expands%20on%20autonomous%20networking%20concepts%20to%20transform%20the%20static%20network%20into%20a%20dynamic%2C%20programmable%20environment%20driven%20by%20analytics%20and%20intelligence.&body=https%3A%2F%2Fwww.ciena.com%2Finsights%2Fwhat-is%2FWhat-Is-the-Adaptive-Network.html)

[Subscribe](https://www.ciena.com/subscribe)

*The Adaptive Network is a new approach that expands on autonomous networking concepts to transform the static network into a dynamic, programmable environment driven by analytics and intelligence.*

Since the introduction of the first Public Switched Telephone Network, networks have continually evolved. Through the various stages of development—from fixed endpoints in the early Internet to today’s broadband networks that connect mobile users to massive data centers and bandwidth behemoths like Netflix, Amazon, and Facebook—networks have adjusted to accommodate new demands.

The once-static infrastructure is undergoing a more profound transformation than ever before. The latest incarnation is [autonomous networking](https://www.ciena.com/insights/what-is/What-Is-an-Autonomous-Network.html), which is a trend that has been building for some time. The autonomous network runs without much human intervention. It can configure, monitor and maintain itself independently.

But, even though it’s a significant advance, autonomous networking is still too restrictive and too rigid. So Ciena has defined a new approach to the evolution of networking—the [Adaptive Network](https://www.ciena.com/adaptive-network/)—that’s geared toward providing a network that can grow with a company as its business needs and markets change.

**[WHITE PAPER](https://www.ciena.com/insights/white-papers/Introducing-the-Adaptive-Network-Vision.html)**

### [Introducing the Adaptive Network Vision](https://www.ciena.com/insights/white-papers/Introducing-the-Adaptive-Network-Vision.html)

The Adaptive Network is remaking the network into a dynamic, programmable infrastructure built on analytics and automation.

The Adaptive Network allows providers to evolve their current infrastructures into more of a communications loop that relays information from network elements, instrumentation, users, and applications to a software layer for review, analysis, and action—rather than bogging down the network itself.

The Adaptive Network includes three important layers:

* [Programmable infrastructure](https://www.ciena.com/adaptive-network/#programmable-infrastructure)**:** This includes the network’s physical and virtual elements, as well as the telemetry gathered from them. The programmable infrastructure layer is highly intelligent and interprets data so the network can make decisions—whether that means routing traffic around a circuit that's down or investigating and correcting an issue with latency or lower-than-expected capacity on a specific link. Programmable infrastructure requires a flexible grid; a reconfigurable photonic layer to give the ability to reroute channels of variable spectral occupancy across any path, and across any optical spectrum in the network; and telemetry from the IP layer correlated with routing data. In addition, a programmable infrastructure needs tunable coherent transponders to efficiently map a flexible number of client signals to the variable line capacity. In turn, that requires a centralized purpose-built Optical Transport Network (OTN) or packet switching architecture.
* [Analytics and intelligence](https://www.ciena.com/adaptive-network/#analytics-and-intelligence)**:**The programmable infrastructure produces significant amounts of data. Some of it is big data that indicate trends that the network learns and adjusts for over time. Big data can inform the network on how to adjust in the long term, which traffic patterns to look out for, and which parts of the network could be vulnerable. Then there’s small data—things that are happening at a fairly rapid pace. It could be a flicker on a circuit or an immediate request from a customer. Such events require a speedy response from the network, and those moves will be made by the analytics. But once the decisions have been made, a human operator or pre-defined policies could step in and approve or change things as necessary. In a truly autonomous network, there would be no operator influence at this point.
* [Software control and automation](https://www.ciena.com/adaptive-network/#tabbed-component-headline)**:** Research shows the undisputed number one cause of network outages is human error, with estimates as high as 32 percent, according to Dimension Data's 2014 Network Barometer report. Effective automation of network tasks, such as loading access controllers and provisioning routers, or automated calculation and configuration of TE tunnels to optimize traffic and relieve congestion,can eliminate those errors and keep the network running at peak performance. The ability for automation to work across multiple vendors is critical. Some technologies are good at working with one set of devices from a single vendor, but few networks are built on a single vendor’s gear. Networks have to interoperate, using APIs, to function efficiently and move data efficiently and swiftly from point to point.

The development of the Adaptive Network is a watershed moment for the networking world. It’s a cohesive evolution that supports all aspects of intelligent automation—such as intent-based orchestration, analytics, and programmable domain control. It’s a microservices-based architecture that delivers extensibility and scale. Plus, it takes a DevOps integration approach to provide operational and service agility.

The Adaptive Network is a new approach that expands on autonomous networking concepts to transform the static network into a dynamic, programmable environment driven by analytics and intelligence.

# **4 ways an Adaptive Network can overcome today’s challenges and take your network to the next level**

August 14, 2018

#### **Françoise Pouliquen, Carrier Account Director, Ciena EMEA**

[Tweet](https://twitter.com/share?url=https://www.ciena.com/insights/articles/4-ways-an-Adaptive-Network-can-overcome-todays-challenges-and-take-your-network-to-the-next-level.html&text=4%20ways%20an%20Adaptive%20Network%20can%20overcome%20today%E2%80%99s%20challenges%20and%20take%20your%20network%20to%20the%20next%20level)

[Share](https://facebook.com/sharer/sharer.php?u=https%3A%2F%2Fwww.ciena.com%2Finsights%2Farticles%2F4-ways-an-Adaptive-Network-can-overcome-todays-challenges-and-take-your-network-to-the-next-level.html)

[+1](https://plus.google.com/share?url=https%3A%2F%2Fwww.ciena.com%2Finsights%2Farticles%2F4-ways-an-Adaptive-Network-can-overcome-todays-challenges-and-take-your-network-to-the-next-level.html)

[Share](https://www.linkedin.com/shareArticle?mini=true&url=https%3A%2F%2Fwww.ciena.com%2Finsights%2Farticles%2F4-ways-an-Adaptive-Network-can-overcome-todays-challenges-and-take-your-network-to-the-next-level.html)

[E-mail](mailto:?subject=Network%20management%20conditions%20are%20tough%20for%20carriers%2C%20with%20legacy%20network%20limitations%3B%20and%20intense%20competition%20from%20new%20market%20entrants%20and%20evolving%20business%20models.%20An%20agile%20and%20adaptive%20network%20can%20help%20operators%20overcome%20these%20challenges%3B%20and%20exploit&body=https%3A%2F%2Fwww.ciena.com%2Finsights%2Farticles%2F4-ways-an-Adaptive-Network-can-overcome-todays-challenges-and-take-your-network-to-the-next-level.html)

[Subscribe](https://www.ciena.com/subscribe)

*In the*[*first blog in this series*](https://www.ciena.com/insights/articles/the-adaptive-network-why-automation-alone-is-not-enough.html)*, we discussed the key aspects of the Adaptive Network, and why, as traffic increases and becomes less predictable, effective partnerships will become more critical. We all know that network management conditions are tough for carriers, with legacy network limitations; and intense competition from new market entrants and evolving business models. An agile and adaptive network can help operators overcome these challenges; and exploit the emerging opportunities like IoT and 5G use cases, says Françoise Pouliquen, Carrier Account Director, Ciena EMEA.*

There is a relentless push-pull from rapid business and technology change affecting operators today. On one hand, dramatic growth in subscriber demands are driving fronthaul and backhaul traffic and putting networks under intense pressure. While on the other, there’s an industry wide race to develop and commercialize new revenue-generating services, such as IoT use cases and 5G mobile services – and to implement the network technologies and architectures needed to support and deliver them. On top of that, new market entrants, including some of the largest internet companies, are deploying massive-scale network connections that support low-cost data transport between key locations and data centers with unrivalled economies of scale.

The challenges for operators are; how to take exponential traffic growth in stride; how to prepare the network for the next-generation of IoT and 5G use cases; and how to remain competitive on price with large connectivity providers in the market.

### [Light Reading webinar: Intelligent Multi-Layer Automation – Mapping the Way to a Self-Healing, Self-Optimizing Network](https://event.on24.com/wcc/r/1813363/E3200CB52652CE2F6BACCE351766A7CE?partnerref=ANLR5social18)

Here are four key ways an Adaptive Network can help:

**1) Increasing network agility and efficiency** – because the Adaptive Network turns the network into a dynamic, programmable infrastructure built on analytics and automation, it helps meet growing bandwidth needs with on-demand scalability. As well as helping operators to handle incremental traffic growth and unpredictable demand peaks, an Adaptive Network supports real-time scaling and resource-allocation to support differentiated QoS for different applications and use cases, paving the way for commercial 5G services.

The agility of an Adaptive Network also helps operators to maximize efficiency by automating a wide range of manual networking processes, from routine service provisioning and turn-up, to resource discovery and traffic routing over the best available components and paths. In this way, it helps operators compete effectively with even the largest connectivity providers.

**2) Future-proofing the network with industry leading packet-optical solutions -**Ciena has created the industry’s most scalable portfolio of programmable, packet-optical network infrastructure to help operators meet massively growing bandwidth demand up to the edge.  Our packet-optical solutions cover the metro edge, between data centres, the backbone core, and submarine. This packet-optical market leadership, enabled by consistently high R&D investment, is based on our deployed 100G, 200G, and 400G capable in-house modem technology, supported by a unique combination of software intelligence to bring the best of optical innovation to the market.

Our leading technology portfolio also supports seamless convergence of voice and data traffic for mobile operators. This will allow operators to integrate 4G and 5G traffic in the future, and to deliver it extremely cost effectively across a unified infrastructure.

By ensuring that the network can keep pace with exponential increases in bandwidth demands and new services requirements (low latency, high availability) our packet-optical portfolio protects our clients’ business for the long-term.

**3) Helping avoid vendor lock-in with open networking -**Many network providers design their portfolios to work together, however, infrastructure is becoming increasingly complex, needing to integrate components and processes in multi-vendor environments. This approach requires large-scale “lift-and-shift” infrastructure upgrades which are costly and disruptive, as well as reducing ROI on existing network equipment.

To maximize cost efficiency and value for our clients, Ciena’s portfolio of hardware and software are designed on the principle of [openness.](https://www.ciena.com/products/) This allows operators to tie an entire network infrastructure together into a single environment that delivers value for the business and end-customers long-term.

As well as integrating all equipment – both legacy and new – Ciena can help monitor and manage multi-vendor networks with a centralized, integrated set of tools. This capability is delivered by [Ciena’s Blue Planet](https://www.ciena.com/blueplanet/products/) software suite and our [Manage, Control and Plan](https://www.ciena.com/products/manage-control-plan/) (MCP) software, which gives full visibility of resources and services across multi-vendor domains, with tools to troubleshoot and manage diverse infrastructure components remotely.

**4) Driving network innovation in strategic partnership** - Finally, but equally importantly, Ciena is a strategic partner for global operators. Based on our financial and operational stability, we are able to commit to continual innovation of our portfolio, ensuring that our clients can embrace emerging opportunities and take future network challenges in our stride.

One example of how we are investing for the future is our recent acquisition of [Packet Design](https://www.ciena.com/insights/articles/Ciena-Completes-Acquisition-of-Packet-Design.html), with network performance management software that is focused on Layer 3 network optimization, topology and route analytics. By integrating Packet Design into Blue Planet, we will be able to extend our intelligent orchestration and automation capabilities from layer 0, 1 and 2 into the IP layer. As a result, our clients will be able to further optimize service delivery and maximize resource utilization – taking the Adaptive Network to the next level.

Adaptive Interface to Scalable Cloud Storage:

Many of today’s applications are delivered as scalable, multi-tier services deployed in large data centers. These services frequently leverage shared, scale-out, key-value storage layers that can deliver low latency under light workloads, but may exhibit significant queuing delay and even dropped requests under high load.

Stout is a system that helps these applications adapt to variation in storage-layer performance by treating scalable key-value storage as a shared resource requiring congestion control. Under light workloads, applications using Stout send requests to the store immediately, minimizing delay. Under heavy workloads, Stout automatically batches the application’s requests together before sending them to the store, resulting in higher throughput and preventing queuing delay. We show experimentally that Stout’s adaptation algorithm converges to an appropriate batch size for workloads that require the batch size to vary by over two orders of magnitude. Compared to a non-adaptive strategy optimized for throughput, Stout delivers over 34× lower latency under light workloads; compared to a non-adaptive strategy optimized for latency, Stout can scale to over 3× as many requests.