

Designing an Adaptive Automation System for Human Supervision of Unmanned Vehicles: A Bridge from Theory to Practice

Ewart J. De Visser^{1,2}, Melanie LeGoullon^{1,2}, Amos Freedy², Elan Freedy², Gershon Weltman² & Raja Parasuraman¹

¹George Mason University, Fairfax, Virginia

²Perceptronics Solutions Inc., Sherman Oaks, California

Careful consideration must be given to the implementation of automation into complex systems. Much research in adaptive automation has identified challenges for system implementation. A key focus of this research has surrounded the methods of automation invocation including critical events, measurement, and modeling techniques. However, little consideration has been given to selecting and implementing appropriate techniques for a given system as a guide to designers of adaptive automation. This paper proposes such a methodology. We demonstrate the use of this methodology by describing a case study about a system designed to support effective communication and collaboration between the commander and vehicle operator in an unmanned aerial vehicle (UAV) system.

INTRODUCTION

An emerging area of opportunity for exploring innovative automation systems is the growing field of small unmanned vehicles in the military. Introducing robotic assets such as Small Unmanned Aerial Vehicles (SUAVs) into company-level units enhances the unit's overall autonomy and extends both manned and unmanned capabilities. However, unreliable voice communications between commander and robot operators coupled with a lack of a common view of the robot's activities can cause problems for these human-robot teams. More generally, the human operators of these systems will be involved in supervisory control of SUAVs and, in the extreme case, will operate multiple systems while on the move and while under enemy fire. Because of the consequent increase in the cognitive workload demands, automation will be needed to support human-system performance.

Practical experience and much research have shown, however, that 'blindly' automating all possible features may not lead to the best solution, especially when a user has to interact with the automation (Parasuraman & Riley, 1997). Some of the possible negative side effects of automation use are reduced situation awareness, inappropriate trust, unbalanced workload, decision biases, and over-reliance and complacency (Lee & See, 2004; Sarter, Woods, & Billings, 1997; Sheridan & Parasuraman, 2006).

One solution to these side effects of automation is to design a system that is flexible and responsive to user needs, environmental demands, and context. This form of automation is known as *adaptable* or *adaptive* automation (Opperman, 1994). Adaptable automation is automation that can be modified by the user whereas adaptive automation is automation invoked by the system itself based on some set criteria. These types of automation may accommodate the user to a greater extent than rigid, static automation. Conversely, workload may increase if the user is highly involved in modifying the system while the unpredictability of a system may increase if a user's involvement is too small (Miller &

Parasuraman, 2007; Sarter et al., 1997). The challenge for designers is to strike a balance between these extremes.

ADAPTIVE AUTOMATION DESIGN

Qualitative Models

The key challenge in the design of adaptive automation is determining *what*, *when* and *how much* to automate. Some guidance on *how much* to automate has been provided by proposing the levels of automation framework (Sheridan & Verplank, 1978), which describes an automation dimension with 10 levels ranging from the human performing every task (Level 1) to the machine performing all tasks (Level 10) with various gradations in between. This framework can be a useful guide in determining the division of labor between all agents in the system, but it does not provide advice on which tasks to automate. Further guidance on *what* to automate was provided by Parasuraman et al. (2000) who proposed an automation design taxonomy that incorporated both the level of automation and the type of automation. In their model four types of automation are described based on a basic human information processing model: information acquisition, information analysis, decision making, and decision (action) implementation. Both models are useful in making qualitative recommendations for automation design. For example, one can select the type of automation and then determine the appropriate level of automation based on certain criteria such as workload, risk, situation awareness, etc. (Parasuraman, 2000).

Invocation Methodologies

A great deal of adaptive automation research has focused on *when* to automate. Three major categories of invocation have been identified in the past: 1) critical events; 2) measurement based invocations; and 3) model based invocations (Inagaki, 2003; Parasuraman, Bahri, Deaton,

Morrison, & Barnes, 1992; Prinzel, 2003). Critical event strategies involve some sort of meaningful event that warrants the engagement of automation. This can be accomplished by implementing a simple logic: if this event happens, then this action should be implemented. Conversely, measurement invocation strategies measure the operator's cognitive state which is compared to a threshold deemed high enough to warrant engaging the automation. Measurement strategies can include physiological measurement (Byrne & Parasuraman, 1996), performance measurement (Endsley, 1999; Kaber & Endsley, 2004), or a combination of both (Wilson & Russell, 2003). Finally, automation can be engaged based on a pre-defined model. Such models can include models for the operator, or a combination of the critical events and measurement based strategies – the hybrid model. An often neglected aspect of these invocation methods is user preference. At an early stage, Rouse et al. (1987) pointed out that the designers of an adaptive aiding system should consider, along with performance and capabilities, the plans and intentions of the operator. Similar arguments along these lines have been made from an ethical perspective (Hancock, 1997; Hancock & Szalma, 2003), but not much thought has been given to modeling the preference of users in the design of adaptive automation systems.

Quantitative Models

The automation taxonomies discussed so far are primarily qualitative in nature and may not provide enough guidance for a designer to make informed decisions about the final function allocation within a system. Parasuraman (2000) therefore proposed to use quantitative models in combination with these qualitative models to offer additional direction.

A variety of cognitive modeling techniques exist that can be useful for adaptive automation, for both performance and user-preference considerations. A widely used and extensively validated technique is GOMS modeling, a family of methods aimed to create models of expert users by breaking down tasks into their goals, operators, methods, and selection rules (Card, Moran, & Newell, 1983) Other options include using cognitive architectures to model human cognition such as ACT-R, Soar, and EPIC. To our knowledge, these specific models have not been applied formally to adaptive automation, although one GOMS model has been adapted to address issues with Human-Robot Interaction (Drury, Scholtz, & Kieras, 2007).

Developing a single quantitative method for implementing adaptive automation may be a daunting task. The challenge is to select the most appropriate model given the needs of a particular design and this may vary from case to case.

AN ADAPTIVE AUTOMATION DESIGN METHODOLOGY

Even though a large adaptive automation literature exists, designers are challenged with several issues. First, it is not clear which method of invocation is most suitable for a

given system. Perhaps adaptive behaviors are best triggered by critical events in one system and operator's workload in another. Second, little thought has been given to modeling the preference of the user in adaptive systems, although the Rotorcraft Pilot's Associate Program is an exception (Miller & Parasuraman, 2007).

The following section of the paper is meant to give a general guide to the design of adaptive automation for designers and developers based on the theory, experimental data, and models provided in the previous section. Often a gap exists between those who validate and create theoretical models and those who need to implement the actual design. We provide an approach that is targeted to help in the early stages of interface design, when no experimental data are available yet.

Adaptive Automation Architecture

Adaptive systems are generally referred to as knowledge-based systems which typically have a task model and a user model (Miller & Parasuraman, 2007). Task models can be useful for identifying plans and goals by a user's interaction of the system. User models can be helpful in recognizing the different needs of various users. In order to define both the task and the user model, we propose a methodology comprised of five steps: 1) collect observational data of a system; 2) conduct task analyses; 3) create preliminary design; 4) select a method of invocation; and 5) implement and validate method of invocation. In the next section we provide guidance for each step.

Task Observation & Analysis

Dozens of techniques are available to model the tasks of a user (Diaper & Stanton, 2003; Kirwan & Ainsworth, 1992). Typically, the first step in this process involves gathering data about the tasks of a system using techniques as structured interviews, review of documentation, and task observation. The second step is analysis of the obtained system tasks. Often, a combination of techniques provides the best analysis and overview of a system. One general analysis is the cognitive task analysis (CTA), which aims to model all the tasks in a system based on the cognitive requirements of the user including his goals, plans, and actions. This type of analysis is particularly useful for potential adaptive systems because a complete overview of tasks is needed before they can be allocated to either the user or the system in particular situations. The result of such an analysis is an overview of the system that can be used to derive system requirements. These requirements can then guide the adaptive interface design. There are a number of different CTA models and associated software available (John, Prevas, Salvucci, & Koedinger, 2004).

Preliminary design considerations

A number of specific interface considerations have been identified for adaptive automation systems. In particular,

the need to balance a person's workload while maintaining situation awareness has been indicated as an important objective in the design of adaptive automation (Kaber, Riley, Tan, & Endsley, 2001). Some already identified interface issues include displaying the right amount of information to the operator in a comprehensible manner, making clear the shift between manual and automated control, workload increases due to monitoring and management demands, and transparency of the machine's intentions (for a review see Kaber et al., 2001).

Selecting and implementing a method of invocation

Once a preliminary design has been created, the method of invocation should be considered. Candidate tasks for critical events invocations can be found by searching the task analysis for information that may require immediate attention of the user. These tasks can be considered critical events. Rules for critical events can be simple if-then rules and implemented with an appropriate level of automation. To select measurement or model invocations, a designer must consider which data will be available from the system. If data about the user is available in a continuous real-time manner, the designer may be able to derive performance, effort, and intent measurements. Based on these real-time measurements, adaptations can be made to balance an operator's task load. If no data is available about the operator, a pre-defined model may be useful. An intent model could be based on a model of all tasks. Sequences of these tasks can be broken up into plans and goals and compared to actual user input. Adaptations can then be made when specific sequences of user actions occur.

We illustrate the above methodology with a case study in the following section.

A CASE STUDY

Our challenge was to design an adaptive interface for commanders that facilitated information flow between robots and humans and enhanced commander-vehicle operator coordination. We chose the Raven- the U.S. Army's Small Unmanned Aircraft System (SUAS)- as our specific use case and evaluation testbed. In the current Raven system, the commander has a remote viewing terminal (RVT) that serves as the main source of visual imagery from the SUAV. The display on this RVT is identical to that of the vehicle operator's (VO) and consists of several flight parameters superimposed over a camera scene.

Our two-pronged approach was to 1) enhance the current system by adaptively displaying relevant information as the mission progressed, and 2) extend the current capabilities of the system by turning the display into a portal through which the commander could access advanced features and communicate intent to the Raven operators.



Figure 1. Field observations were conducted at two locations to understand the commander's informational needs.

Step 1: Field Observations

In order to understand the commanders' tasks and their information needs, extensive field observations were conducted at two locations: Ft. Benning, GA and Ft. Polk, LA. In the former location, the observations focused on Army doctrine, training, and schoolhouse knowledge of small UAVs by observing a day of training at the Small Unmanned Aircraft System (SUAS) Course. In this course, current Raven operators were being trained to become Raven instructors for their units. In the latter location, we observed company commanders and their infantry units over several days of a week-long full-scale exercise intended to prepare Army units for upcoming deployments (see Figure 1).

The observational data collected from both locations resulted in a wealth of information including 50 hours of field observations, 2 hours of audio transcriptions and over 3 hours of videotape footage from interviews with Raven operators.

Step 2: Cognitive Task Analysis

Using the knowledge gained from the schoolhouse doctrine and instructors' interviews, we developed a comprehensive HTA that highlighted the typical tasks a commander performs during a Raven mission as well as the specific information requirements associated with those tasks. This task analysis was then used to prepare probes for the field observations made during the training exercise. The results from this second trip served to inform a modified version of Klein, Calderwood, and MacGregor's (1989) Critical Decision Method CTA.

In conducting the HTA and CTA, an interesting pattern emerged. It became clear that the commander's tasks could be sorted into three general super-tasks: monitoring the current video feed, reviewing past information, and re-tasking the vehicle in-flight. We refer to these categories collectively as *mission phases*.

In addition to this emergent categorization of tasks, the CTA identified several bottlenecks that might plague a

typical mission. The most common (and arguably the most disruptive) bottleneck was the radio communications between commander and Raven operators as shown by the operational sequence diagram (see Figure 2).

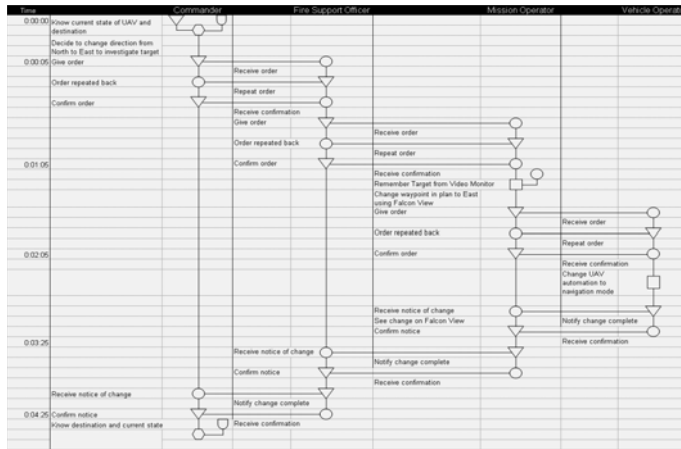


Figure 2. OSD showing communications bottleneck for commander.

Factors contributing to this bottleneck were radio frequency congestion, intermediaries, and the absence of the message recipient. A second bottleneck stemmed from the commander’s incomplete understanding of the Raven system, which sometimes leads to additional radio communications.

A final bottleneck was the slow manner in which information obtained by the Raven team was distributed from the command post to the rest of the company.

Step 3: Preliminary Adaptive Interface Design: Overview

Our proposed design is a departure from the current system in which the commander must make do with raw flight data and spend precious time waiting for queries, re-tasking directives and responses to be relayed over the radio net to the vehicle operators. Re-casting the bottlenecks listed above as critical areas of improvement, we identified three main system requirements to guide our design: First, the new interface must reduce reliance on verbal radio communications. Second, the interface must be intuitive. And third, the interface must facilitate the production of products for later consumption, such as during patrols and AARs.

In short, we envision an interactive display that bypasses the brittle and cumbersome nature of voice radio communications by creating a data link between the interfaces of the commander and the operators.

Mission Modes

Based on the three distinct mission phases identified by the CTA, we proposed three analogous Mission Modes that give the commander the flexibility he needs to manage the UAV asset effectively based on his information needs and time available (see Figure 3). The modes are briefly described below.

Monitor Mission Mode. The monitor mission mode is the default view and will allow the commander to watch the UAV video feed in real-time. The video display is largest in the configuration to allow easy situation assessment by the commander. The current location of the asset is readily available in a thumbnail map while details of what is being tracked appear in the mission analysis panel.

Review Mission Mode. The review mission mode allows the commander to examine stored imagery of targets & landmarks that have been obtained by the UAV. In this mode the commander can flip through captured images, obtain distances for roads and landmarks, read target-specific information entered by the MO, and mark up the images with several annotation tools in the Mission Analysis panel.

Change Mission Mode. The change mission mode allows the commander to quickly signal a course change to the UAV team when needed. The map is largest in this configuration to allow accurate flight plan review and in-flight re-tasking requests.



Figure 3. Monitor, Review, and Change Mission Mode.

Step 4: Selecting Invocation Methods

The proposed interface will switch modes adaptively based on (1) mission type, (2) critical events, and (3) individual preferences of the commander. The critical question is: Which invocation points should trigger the adaptive mode selection and when should they be activated?

Our approach to this problem was to use the goals and methods from the HTA to build a GOMS model for the commander. The methods in this model describe the specific goals a commander can accomplish with the interface. The decisions represent the choices the commander has to make when accomplishing these goals. The selection rules describe which procedural IF-THEN rule the commander can follow when multiple options are available. The model provides a useful framework to support the various invocation methods.

Mission Type. Different missions require different goals to be accomplished. For instance, reconnaissance missions typically have a specific aim in mind such as “verify target X is at location Y” while surveillance involves a more general goal of observing whatever can be seen. Depending on the mission at hand, mission specific goals can be included or excluded from the model.

Critical Events. During a mission, certain events may require an interface adaptation. For example, upon spotting a suspicious car, an operator engages the *loiter* mode and the *left* camera for the Raven. Following such a sequence, the commander’s display could switch to the Monitor Mode:

```

Selection Rule for Goal: MONITOR MODE
  IF Flight mode = Loiter and Camera = Left and
Monitor Mode = off,
  THEN Accomplish Goal: SWITCH TO MONITOR MODE
  IF Flight mode = Loiter and Camera = Left and
Monitor Mode = on,
  THEN Accomplish Goal: MAINTAIN MONITOR MODE

```

Commander Preference. The system can also learn the preferences of commanders by monitoring and analyzing their goals and decisions and incorporating such information into a Bayesian network. For example, one commander may frequently take a picture of the same object of interest. The system may offer to take the picture for commanders if they have a high probability of deciding to do so.

Step 5: Implement and Validate Method of Invocation

One of the well-documented benefits of GOMS is the ability to make *a priori* predictions about performance times of expert behavior. Thus, we tested the feasibility of our design in the very early stages of development by comparing the predicted performance times of our proposed interface against the predicted performance times of the current system for three representative tasks. The results showed a 48% improvement in performance time, shaving almost 5 minutes off a nearly 10 minute series of tasks. Based on these promising results, we can further justify continued development of this new design.

DISCUSSION

The adaptive automation concept has a long history (Parasuraman et al., 1992; Rouse, 1987), but empirical studies are more recent (see Inagaki, 2003; and Parasuraman, 2000; for reviews). In general, empirical evaluations have shown that adaptive systems lead to more balanced workload, improved situation awareness, and reduced automation over-reliance (Parasuraman, 2000). However, only a few prototype adaptive systems have been developed. One example is the Rotorcraft Pilot's Associate, which aids helicopter pilots in an adaptive manner depending on mission context and which has successfully passed in-flight tests (Dornheim, 1999). Furthermore, validated design methodologies do not exist for implementing adaptive automation in real systems.

This paper represents a first step towards formulating a framework for adaptive automation design, with a focus on unmanned vehicle systems. We have described a general methodology for implementing adaptive automation design and provided an example case study. Our methodology may also be applied to other domains requiring design of adaptive automation such as in aviation and driving. For example, a critical challenge in the near future for both aviation pilots and drivers will be how to share and adaptively switch control of the plane or car based on the needs of the operator. If the goal is to create safe and reliable systems a sound methodology for designing adaptive automation is critical. Further efforts to validate this methodology may eventually lead to a practical guide for designers creating systems with adaptive automation.

REFERENCES

- Byrne, E. A., & Parasuraman, R. (1996). Psychophysiology and adaptive automation. *Biological Psychology*, 42(3), 249-268.
- Card, S. K., Moran, T. P., & Newell, A. (1983). *The psychology of human-computer interaction*. Hillsdale, NJ: Erlbaum.
- Diaper, D., & Stanton, N. (2003). *The Handbook of Task Analysis for Human-Computer Interaction*: Lawrence Erlbaum Associates.
- Dornheim, M. (1999). Apache tests power of new cockpit tool. *Aviation Week and Space Technology*, 151(16), 46-49.
- Drury, J., L., Scholtz, J., & Kieras, D. (2007). *Adapting GOMS to model human-robot interaction*. Paper presented at the Proceeding of the ACM/IEEE international conference on Human-robot interaction.
- Endsley, M. R. (1999). Level of automation effects on performance, situation awareness and workload in a dynamic control task. *Ergonomics*, 42(3), 462-492.
- Hancock, P. A. (1997). *Essays on the future of human-machine systems*. Eden Prairie, MN: BANTA Information Services Group.
- Hancock, P. A., & Szalma, J. L. (2003). The future of neuroergonomics. *Theoretical Issues in Ergonomics Science*, 44, 238-249.
- Inagaki, T. (2003). Adaptive Automation: Sharing and Trading of Control. In E. Hollnagel (Ed.), *Handbook of Cognitive Task Design*: Lawrence Erlbaum Associates.
- John, B., Prevas, K., Salvucci, D., & Koedinger, K. (2004). *Predictive Human Performance Modeling Made Easy*. Paper presented at the Proceedings of CHI.
- Kaber, D. B., & Endsley, M. R. (2004). The effects of level of automation and adaptive automation on human performance, situation awareness and workload in a dynamic control task. *Theoretical Issues in Ergonomics Science*, 5(2), 113-153.
- Kaber, D. B., Riley, J. M., Tan, K. W., & Endsley, M. R. (2001). On the Design of Adaptive Automation for Complex Systems. *International Journal of Cognitive Ergonomics*, 5(1), 37-57.
- Kirwan, B., & Ainsworth, L. K. (1992). *A Guide to Task Analysis*: Taylor & Francis.
- Klein, G. A., Calderwood, R., & MacGregor, D. (1989). Critical decision method for eliciting knowledge. *IEEE Transactions on Systems, Man, & Cybernetics*, 19(2), 462-472.
- Lee, J., & See, K. (2004). Trust in automation: Designing for appropriate reliance. *Human Factors*, 46(1), 50-80.
- Miller, C., & Parasuraman, R. (2007). Designing for Flexible Interaction Between Humans and Automation: Delegation Interfaces for Supervisory Control. *Human Factors*, 49(1), 57-75.
- Opperman, R. (1994). *Adaptive user support*. Hillsdale, NJ: Erlbaum.
- Parasuraman, R. (2000). Designing automation for human use: empirical studies and quantitative models. *Ergonomics*, 43(7), 931 - 951.
- Parasuraman, R., Bahri, T., Deaton, J. E., Morrison, J. G., & Barnes, M. (1992). *Theory and Design of Adaptive Automation in Aviation Systems*. Warminster, PA: Naval Air Warfare Center. In.
- Parasuraman, R., & Riley, V. (1997). Humans and automation: Use, misuse, disuse, abuse. *Human Factors*, 39(2), 230-253.
- Parasuraman, R., Sheridan, T. B., & Wickens, C. D. (2000). A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, and Cybernetics. Part A: Systems and Humans*, 30, 286-297.
- Prinzel, L. J. (2003). *Team-centered Perspective for Adaptive Automation Design*: National Aeronautics and Space Administration, Langley Research Center.
- Rouse, W. B., Geddes, N. D., & Curry, R. E. (1987). An Architecture for Intelligent Interfaces: Outline of an Approach to Supporting Operators of Complex Systems. *Human-Computer Interaction*, 3(2), 87-122.
- Sarter, N. B., Woods, D. D., & Billings, C. E. (1997). Automation surprises. *Handbook of human factors and ergonomics*, 2, 1926-1943.
- Sheridan, T. B., & Parasuraman, R. (2006). Human-Automation Interaction. *Reviews of Human Factors and Ergonomics*, 1(1), 89-129.
- Sheridan, T. B., & Verplank, W. L. (1978). *Human and Computer Control of Undersea Teleoperators*. Paper presented at the The 14th Annual Conference on Manual Control, USA.
- Wilson, G. F., & Russell, C. A. (2003). Real-Time Assessment of Mental Workload Using Psychophysiological Measures and Artificial Neural Networks. *Human Factors*, 45(4), 635-644.