A Design Methodology for Controlling, Monitoring, and Allocating Unmanned Vehicles

Ewart de Visser, Marvin S. Cohen, Melanie LeGoullon, Onur Sert, Amos Freedy, Elan Freedy, Gershon Weltman and Raja Parasuraman

***Abstract*—Supervisory control of unmanned systems presents challenges to users at several levels in the military hierarchy. The UV operator has to control and supervise multiple assets, commanders need to quickly extract information, and allocation of vehicles to different clients involved in the mission is an additional demand. Such tasks may have an impact on task load, task switching, selective attention, and trust in the system. At each level therefore, the interface is a crucial tool for effectively controlling, monitoring, and allocating these assets. We present a methodology for designing such a supervisory control interface. We illustrate this methodology with 3 use cases.**

***Index Terms*— Unmanned vehicles, supervisory control, interface design, human-robot interaction, automation.**

I. INTRODUCTION

MILITARY forces of the future will use mixed manned and unmanned forces for a broad variety of functions: reconnaissance and surveillance, logistics and support, communications, forward-deployed offensive operations, and as tactical decoys to conceal maneuvers by manned assets. Such robot-human teams extend both manned and unmanned capabilities are intended “force multipliers”, as in the US Army Future Combat System [1]. To accomplish this, the unmanned vehicles (UV) must be managed as an integrated total resource that supports higher-level objectives while operating at an increased operational tempo under the coherent guidance of distributed human operators.

At the same time, due to manning reductions, the management of multiple heterogeneous unmanned vehicles is being assigned to a decreasing number of human team members. Thus automation support, even if imperfect, is mandated to achieve the goal of having few operators control many UVs simultaneously [2]. Recent estimates of an operator’s capacity to control multiple UAVs range from 1 to 16 [3, 4].

Human-robot teams introduce unique challenges to the planning and coordination of team performance. Foremost among these is the ability of the human team to predict, collaborate and coordinate its actions with complex systems that are acting in unstructured and unpredictable environment with varied levels of autonomy. As already noted, the availability of trained manpower to operate the systems effectively is a key additional factor, especially in military deployment of unmanned systems. Advanced approaches are necessary in order to increase the cognitive efficiency of remaining human elements.

II. USERS IN MULTIPLE AGENT SYSTEMS

The military is a highly structured hierarchical system with multiple ‘agents’. The introduction of UV technology presents challenges at several levels in this military hierarchy as shown in Figure 1. Users at higher levels in the hierarchy may be more interested in the information provide by the UVs, whereas the operator is primarily interested in control and monitoring of the vehicles. Users at each level have unique demands, but also share similar needs.

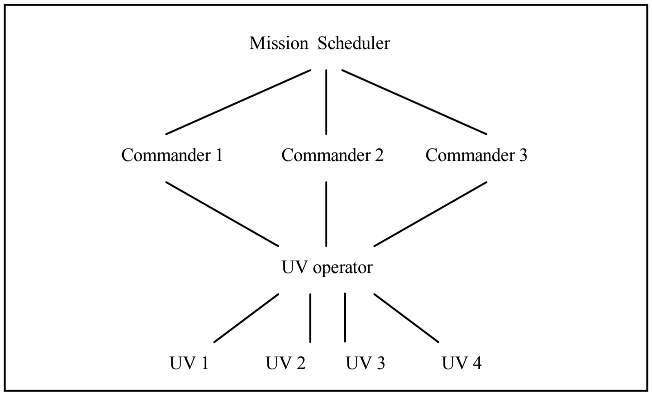


Fig. 1. Overview of future military multiple agent system.

A. The UV Operator

Currently, many single UV systems are manned by multiple operators often using tele-operation or partial automation to control the vehicle. If operators are to effectively control more than one UV at a time, supervisory control is a necessary requirement. Unique requirements for the operator include balancing mental workload, maintaining situation awareness, flexible control and increase sensitivity to robotic failures.

Manuscript received April 1, 2008. Ewart de Visser, Marvin S. Cohen, Melanie LeGoullon, Onur Sert, Amos Freedy, and Gershon Weltman are with Perceptronics Solutions Inc, Sherman Oaks, CA 91423, USA. Ewart de Visser and Raja Parasuraman are at George Mason University, Fairfax, VA 22030, USA. (Contact Ewart de Visser, phone: 910-200-8596; e-mail: edevisse@ gmu.edu).

B. *The Commander*

Commanders need timely, accurate, and easy-to-understand presentation of relevant information collected by UV systems. These systems have capabilities including live monitoring of battlefield information, reviewing reconnaissance imagery, and re-directing assets to areas of interest. If the commanders have multiple vehicles at their disposal, supervisory monitoring control is needed to switch between the different information feeds.

C. *The Mission Scheduler*

In some cases, UV systems may require shared and collaborative control between different clients in the military system. Challenges in successful collaboration in these situations include sorting out conflicting requests for UV resources, delays with mission prioritization, sharing of plans, and missed opportunities to share available resources. In this case, a combination of supervisory control design and collaborative decision-making are needed to successfully support optimal allocation of UVs.

III. AUTOMATION IN SUPERVISORY CONTROL

A. *The need for automation*

The human agents in these systems will be involved in supervisory control of UVs 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 work together or interact with the automation [6]. 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 [7-9]. For successful implementation of supervisory control systems these cognitive factors will have to be considered.

B. *Adaptable and Adaptive automation*

One solution to potential negative side effects of automation in supervisory systems 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 [10]. An example of adaptable automation, automation, in the context of UVs, is a system that allows users to send a set of high level orders to a group of vehicles that can interpret and execute these commands. *Adaptive* automation, on the other hand, is automation invoked by the system itself based on some set criteria. An example of this type of automation is a system that takes complete control of some UVs if the operator's workload is above a pre-defined threshold. 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 [9, 11]. The challenge for designers is to strike a balance between these extremes.

Empirical support for the benefits of both adaptable and adaptive automation for supervisory control is still small, but growing. Previous research has pointed to the general utility of adaptable delegation interfaces in allowing for effective supervisory control of multiple numbers of UVs [2]. The system allowed the user to call 'plays' to a team of robots thereby off-loading some of the responsibility of specifically instructing each vehicle individually. Studies examining the efficacy of adaptive automation while supervising a UGV and UAV have shown benefits in situation awareness and workload [12, 13], and trust [14] compared to manual and static automation (automation that is always on) conditions. The results point to the efficacy of adaptable and adaptive automation as it is *tailored* to unique human user needs.

IV. DESIGN METHODOLOGY

This section of the paper is meant to give a general guide to the design of automation for designers and developers of supervisory control systems. Often a gap exists between those who create and validate theoretical models and those who need to implement the actual design. We provide an approach that is targeted to assist in the early stages of interface design, when no experimental data are available yet. Our methodology comprises five steps: 1) collect observational data of a system; 2) conduct task analyses; 3) construct a quantitative model 4) create preliminary design; 5) validate design. In the next section we provide guidance for each step.

A. *Data collection & Task Analysis*

Dozens of techniques are available to model the tasks of a user [15, 16]. Typically, the first step in this process involves gathering data about the tasks of a system using such 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. 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 interface design. There are a number of different CTA models and associated software available [17].

B. *Constructing a Qualitative or Quantitative Model*

Parasuraman has [18] proposed the use of quantitative models in combination with qualitative models [19, 20] to offer direction for automation design. A variety of cognitive modeling techniques exist that can be useful for 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 [21]. Other options include using cognitive architectures to model human cognition such as ACT-R, Soar, and EPIC. To our knowledge, only one attempt has been made to apply GOMS to the human-robot interaction domain [22].

Developing a single quantitative method for implementing automation may be a daunting task. The trick is to select the most appropriate model given the needs of a particular design.

C. *Create Preliminary Interface Design*

A number of specific interface considerations have been identified for adaptive and adaptable automation systems. In particular, the need to balance a person's workload while maintaining situation awareness has been concluded as an important objective in the design of automation [6, 23] as well as high *task switching* times [24]. Some other 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 23].

D. *Validating the design*

Preliminary designs can be validated in a number of ways. One way method is to attach the interface to a UV simulator and conduct experimental studies. When no such simulator is available, designs can also be validated by conducting usability studies.

V. THREE USE CASES

The design methodology described above was applied to three separate cases of supervisory control interface design. In particular, there is a need to develop interfaces to build acceptable plans to allocate UVs where needed. In the next section, we discuss useful and common interface components interfaces to facilitate supervisory control.

A. *Use case I: Adaptive delegation interface*

Miller and Parasuraman [11] have proposed a concept of adaptive human-automation interaction based on the concept of *delegation* – the concept is called Adaptive Delegation Interfaces, or ADI. Adaptive delegation interfaces are *adaptive* because they are responsive to context and user needs, and involve *delegation* in the same sense that a supervisor works with a human subordinate—the difference being that the human uses an interface that allows for high-level communication with the automation in a common language.

To successfully develop such an interface, a number of requirements have to be satisfied including defining a common language, developing automation that can critique the plans of the operator, and showing plans in multi-modal fashion.

An example of the adaptive delegation interface is shown in Figure 2. The interface has three modes: mission wizard, mission compose, and mission execution. The mission wizard facilitates selecting a set of high level mission goals that can be send to the DCOP automation (see Use Case III) for plan generation. The mission compose view shows the structure of the plans with the common command language that both the automation and the user can use to plan the mission. The language is comprised of tasks, plays, and superplays that give the user flexibility to assign tasks at varying degrees of granularity. There are multiple planning views available to enable a user to plan the mission in a variety of ways including by asset, by time (using a Gantt chart), and by phase. Finally, the mission execution screen is activated when the plan is put into action. The operator can then actively monitor the mission and intervene with 'plays' if necessary.

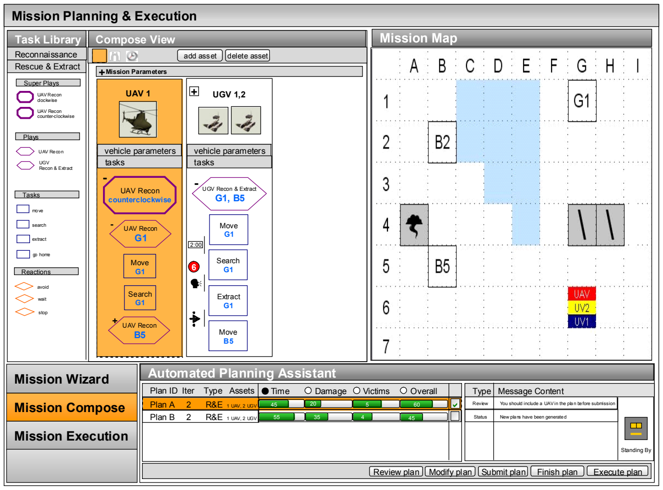


Fig. 2. Screenshot of the adaptive delegation interface.

B. *Use Case II: The commander's interface*

This use case demonstrated the development of an adaptive interface for commanders that facilitates information flow between robots and humans and enhanced commander-vehicle operator coordination. We conducted field observations from Raven operators at both a training school and at a training exercise. 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 (see Figure 3). We refer to these categories collectively as *mission phases*. 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. The modes are briefly described below.

The monitor mission mode is the default view and will allow the commander to watch the Raven video feed in real-time (see Figure 4). 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.

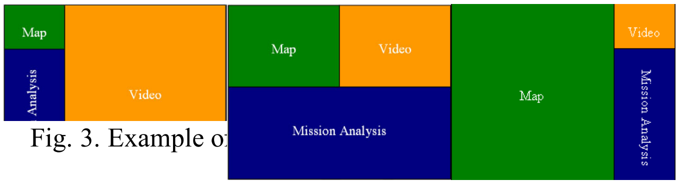


Fig. 3. Example o

The review mission mode allows the commander to examine stored imagery of targets & landmarks that have been obtained by the Raven. 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.

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

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.

Fig. 4. Example of commander interface.

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 an object of interest. The system may offer to take the picture for commanders if they have a high probability of deciding to do so.

C. Use case III: The automated mission scheduler

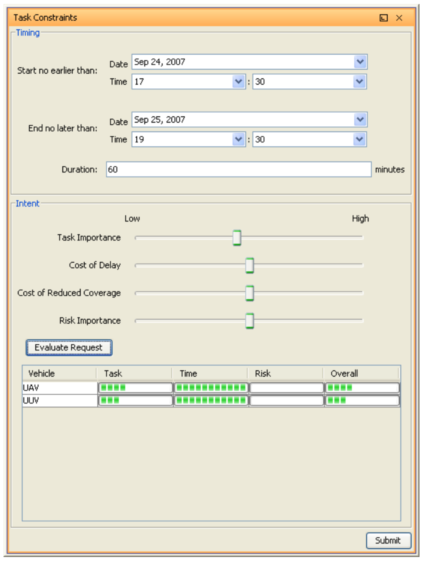
The Automated Mission Scheduler is a net-centric overlay software suite that interfaces with, monitors, and helps optimize the activities of disparate UV command and control elements in a Joint force. It combines multi-agent algorithms for scheduling and task allocation with decision and communication support for mission management and collaborative conflict resolution among clients and operators. AMS has two main components:

- The DCOP Mission Scheduler deploys highly efficient multiagent algorithms that automatically create mission plans for heterogeneous, multiple unmanned vehicles, taking into account multi-level mission objectives, battle space information, resource capabilities and availabilities, client requests, and operator inputs.
- The Collaboration Management Interface supports decision making by both operators and clients, and facilitates problem-solving, communication, and negotiation when necessary to resolve conflicts and optimize with respect to multiple criteria or collective goals. It permits clients, such as tactical commanders in need of information collection resources, to make requests, review automatically generated plans, evaluate alternative possibilities, and negotiate with other clients or operators when appropriate. It supports operator tasks of reviewing, evaluating, adjusting, and responding to multiple simultaneous client requests.

Clients specify objectives or tasks, which include target types, the kind of information required, time windows for collecting it, and geographic regions. The collaboration management tool gives clients the flexibility to formulate requests at different levels of granularity and specificity. Requests for specific vehicles or sensor types is optional.

While specifying their objectives, clients provide several simple evaluative assessments, such as the following for surveillance or reconnaissance missions as shown in Figure 5:

- Target importance is the value of timely, non-degraded information of the type requested (1 = low importance, 5 = average, 10 = critical)
- Delay cost is the proportion of the total information value lost per time interval delay after or before the requested time window (0-100%).
- Reduced Coverage cost is the proportion of the total information value lost per unit time reduction in surveillance duration (0-100%).



Start no earlier than: Date Sep 24, 2007 Time 17 : 30

End no later than: Date Sep 25, 2007 Time 19 : 30

Duration: 60 minutes

Vehicle	Task	Time	Risk	Overall
UAV	🟩🟩🟩🟩	🟩🟩🟩🟩🟩🟩🟩🟩🟩🟩🟩🟩		🟩🟩🟩🟩
UUV	🟩🟩🟩	🟩🟩🟩🟩🟩🟩🟩🟩🟩🟩🟩🟩		🟩🟩🟩

Fig 5. Sliders permit clients to input weights on task objectives. Green bars shown an automatically generated preliminary estimate of how fully the request can be satisfied.

- Risk importance is the cost (negative utility) of tactically undesirable events, such as a UV route's exposing a client's position (0 = no cost, 10 = maximum cost)

Each of these assessments serves as an importance weight for variations on the specified dimension.

VI. CONCLUSION

We have presented a design methodology for implementing automation for supervisory control in UV systems. Three use cases were presented that illustrate this methodology. Key design challenges that emerged were identification of display modes, identification of invocation methods for proper adaptive automation implementation, and a method for delegating high level objectives and tasks to automation for plan generation. Ideas for each of these challenges have been proposed.

VII. REFERENCES

[1] K. Cosenzo, R. Parasuraman, A. Novak, and M. Barnes, "Adaptive automation for robotic military systems," 2006.

[2] R. Parasuraman, S. Galster, P. Squire, H. Furukawa, and C. Miller, "A Flexible Delegation-Type Interface Enhances System Performance in Human Supervision of Multiple Robots: Empirical Studies With RoboFlag," *IEEE Transactions on systems, man, and cybernetics - part A: Systems and Humans,* vol. 35, pp. 481-493, 2005.

[3] M. L. Cummings and S. Guerlain, "Developing operator capacity estimates for supervisory control of autonomous vehicles," *Human Factors,* vol. 49, pp. 1-15, 2007.

[4] S. R. Dixon and C. D. Wickens, "Automation Reliability in Unmanned Aerial Vehicle Control: A Reliance-Compliance Model of Automation Dependence in High Workload," *Human Factors: The Journal of the Human Factors and Ergonomics Society,* vol. 48, pp. 474-486, 2006.

[6] R. Parasuraman and V. Riley, "Humans and automation: Use, misuse, disuse, abuse," *Human Factors,* vol. 39, pp. 230-253, 1997.

[7] J. Lee and K. See, "Trust in automation: Designing for appropriate reliance," *Human Factors,* vol. 46, pp. 50-80, 2004.

[8] T. B. Sheridan and R. Parasuraman, "Human-Automation Interaction," *Reviews of Human Factors and Ergonomics,* vol. 1, pp. 89-129, 2006.

[9] N. B. Sarter, D. D. Woods, and C. E. Billings, "Automation surprises," *Handbook of human factors and ergonomics,* vol. 2, pp. 1926-1943, 1997.

[10] R. Opperman, *Adaptive user support*. Hillsdale, NJ: Erlbaum, 1994.

[11] C. Miller and R. Parasuraman, "Designing for Flexible Interaction Between Humans and Automation: Delegation Interfaces for Supervisory Control," *Human Factors,* vol. 49, pp. 57-75, 2007.

[12] R. Parasuraman, M. Barnes, and K. Cosenzo, "Adaptive automation for human robot teaming in future command and control systems," *International Journal of Command and Control,* vol. 1, pp. 43-68, 2007.

[13] R. Parasuraman, K. A. Cosenzo, and E. De Visser, "Adaptive automationfor human supervision of multiple uninhabited vehicles: Effects on change detection, situation awareness, and workload," *Military Psychology,* p. revision under review, 2007.

[14] E. De Visser, D. Horvath, R. Parasuraman, and K. Cosenzo, "Adaptive automation for human interaction with uninhabited vehicles," in *Proceedings of the Midyear meeting of Division 21 of the American Psychological Association*, Fairfax, VA: George Mason University, 2008.

[15] D. Diaper and N. Stanton, *The Handbook of Task Analysis for Human-Computer Interaction*: Lawrence Erlbaum Associates, 2003.

[16] B. Kirwan and L. K. Ainsworth, *A Guide to Task Analysis*: Taylor & Francis, 1992.

[17] B. John, K. Prevas, D. Salvucci, and K. Koedinger, "Predictive Human Performance Modeling Made Easy," in *Proceedings of CHI* Vienna, Austria: ACM, New York, 2004.

[18] R. Parasuraman, "Designing automation for human use: empirical studies and quantitative models," *Ergonomics,* vol. 43, pp. 931 - 951, 2000.

[19] R. Parasuraman, T. B. Sheridan, and C. D. Wickens, "A model for types and levels of human interaction with automation," *IEEE Transactions on Systems, Man, and Cybernetics. Part A: Systems and Humans,* vol. 30, pp. 286-297, 2000.

[20] T. B. Sheridan and W. L. Verplank, "Human and Computer Control of Undersea Teleoperators," in *The 14th Annual Conference on Manual Control*, USA, 1978, pp. 343-357.

[21] S. K. Card, T. P. Moran, and A. Newell, *The psychology of human-computer interaction*. Hillsdale, NJ: Erlbaum, 1983.

[22] J. Drury, L. , J. Scholtz, and D. Kieras, "Adapting GOMS to model human-robot interaction," in *Proceeding of the ACM/IEEE international conference on Human-robot interaction* Arlington, Virginia, USA: ACM, 2007.

[23] D. B. Kaber, J. M. Riley, K. W. Tan, and M. R. Endsley, "On the Design of Adaptive Automation for Complex Systems," *International Journal of Cognitive Ergonomics,* vol. 5, pp. 37-57, 2001.

[24] P. Squire, G. Trafton, and R. Parasuraman, "Human Control of Multiple Unmanned Vehicles: Effects of Interface Type on Execution and Task Switching Times," in *1st Annual Conference on Human-Robot Interaction*, Salt Lake City, UT, 2006.