17. SCALING-UP HUMAN CONTROL FOR LARGE UAV TEAMS

Michael Lewis, Jumpol Polvichai, Katia Sycara and Paul Scerri

SCALING-UP HUMAN CONTROL FOR LARGE UAV TEAMS

Wide area search munitions (WASMs) are a cross between an unmanned aerial vehicle and a munition. With an impressive array of onboard sensors and autonomous flight capabilities WASMs might play a variety of roles on the modern battle field including reconnaissance, search, battle damage assessment, or communications relay.

The first of these high concept munitions, the low cost autonomous attack system (LOCAAS), is a miniature, autonomous powered munition capable of broad area search, identification, and destruction of a range of mobile ground targets. The LOCAAS uses a small turbojet engine capable of powering the vehicle for up to 30 min and laser radar (LADAR) with automatic target recognition (ATR) to identify potential targets.

While the LOCAASs were originally designed to operate individually, flying preprogrammed search patterns, the WASM concept envisions artificially intelligent munitions that communicate and coordinate to perform their tasks. Were multiple independent LOCAASs to fly in close proximity, a variety of problems including fratricide, strikes against already

---

Human Factors of Remotely Operated Vehicles
Advances in Human Performance and Cognitive Engineering Research, Volume 7, 237–250

ISSN: 1479-3601/doi:10.1016/S1479-3601(05)07017-7

dead targets, suboptimal coverage of the search region, and absence of battle damage assessment might arise. These problems could all be resolved by cooperation among the munitions. The next generation of WASMs are posited to have reliable communication with each other and with manned aircraft and ground forces in their area to allow cooperation and control. These communication channels will be required to transmit data, perhaps including video streams, to human controllers, as well as for inter-WASM coordination. We are developing and testing prototype interfaces for interacting with small WASM teams and developing new approaches to allow human control and coordination to be scaled to large (100–1000) WASM teams.

Human control of teams of autonomous machines presents a variety of new human factors problems discussed in Roth, Hanson, Hopkins, Mancuso, and Zacharias (2004). Fully autonomous teams must be programmed in detail before their mission begins. This is typically accomplished using a graphical interface on which a sequence of waypoints are specified (Endo, MacKenzie, & Arkin, 2004; Miller & Parasuraman, 2003) and changes in mission phase and reactive behaviors are associated with some of these waypoints. Programmed behaviors may involve either individual robots (Endo et al., 2004) or a cooperating team (Scerri, Sycara, & Tambe, 2004a; Scerri, Xu, Liao, Lai, Lewis, & Sycara, 2004b). Once the mission is started the human operator may have no further input. Interacting with an executing team offers more possibilities for control. These interactions may redirect the team by changing waypoints, search regions, targets (Cummings, 2004), or otherwise manipulating robot goals.

Other avenues to control include altering selected behaviors such as the selection of plays in Playbook (Miller & Parasuraman, 2003), or altering behavioral parameters such as changing the value of a robot's *wanderlust* (deviations from a direct path between waypoints) in MissionLab (Endo, et al., 2004). Anticipating the effects of actions and exerting effective control becomes progressively more difficult as the locus of control shifts from observables such as targets to algorithmic parameters. We are currently exploring approaches to controlling teams that combine specifying human roles in team plans, selection among plans, and control of algorithmic parameters as well as manipulation of goals (Scerri et al, 2004a,b). In this chapter we describe a prototype interface for controlling small (4–8) teams of WASMs that has been evaluated for an AC-130 flank patrol task and will be used in an upcoming P-LOCAAS[1] flight test. We then present preliminary results for techniques that may allow operators to control very large UAV teams through reconfiguring coordination algorithm parameters and developing transfer of control policies that allow UAVs to adjust their level of autonomy to compensate for variations in operator workload.

SMALL TEAM WASM CONTROL INTERFACE

Our user interface (shown in Fig. 1) for controlling small WASM teams was constructed by adding a toolbar, taking advantage of drawing and other display functions of the FalconView (FalconView, 2005) personal flight planning system, a popular flight planning system used by military pilots. The user controls individuals or teams of WASMs by sketching ingress paths, search or jettison regions and other spatially meaningful instructions known as tactical areas of interest (TAIs). When a target is detected the user may be alerted and requested to authorize or abort the attack depending on the rules of engagement.

Through a series of dialogs and menu selections the user can select individual WASMs for a task or allow the team to make its own allocation. The FalconView interface communicates with a Lockheed-Martin simulation of LOCAASs and the OneSAF testbed baseline (OTB) (OneSAF, 2004) platoon to brigade level simulation to provide a realistic simulation of the

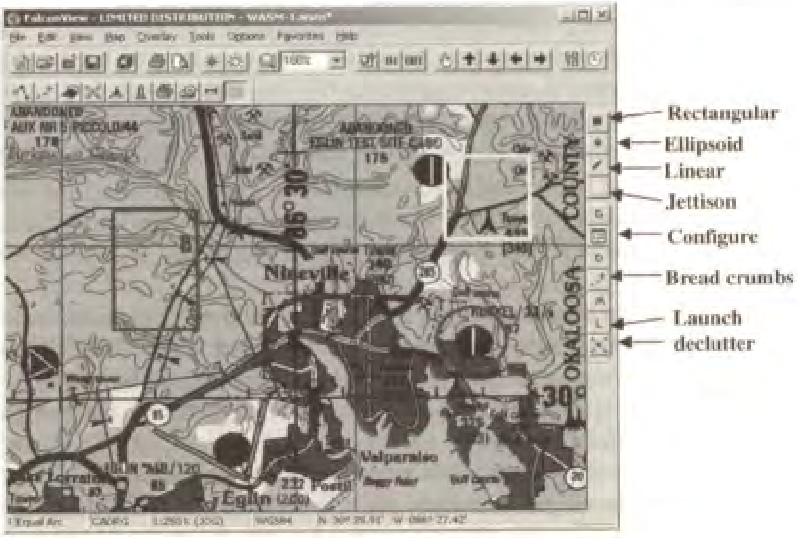

Fig. 1. FalconView based WASM control interface.

interface's capabilities for controlling teams of WASMs. Because many platform simulators such as the AC-130 used in our test also use the Distributed Interactive Simulation (DIS) protocol, OTB provides a *ground truth* server for linking our WASM simulations with other platforms on the simulated battlefield. The simulated WASM broadcasts protocol data units (PDUs) defined by DIS to update its position and pose while listening for PDUs with locations within its sensor cone to detect targets. The laptop presenting the user interface uses custom defined supervisor and weapon state PDUs to convey instructions to the WASMs and monitors WASM PDUs for newly found targets to be added to its display.

CONOPS TEST AND DEVELOPMENT

An initial evaluation of the FalconView tasking interface was conducted for WASM conops for flank patrol for an AC-130 aircraft supporting special operations forces on the ground. The AC-130 is a large, lumbering aircraft, vulnerable to attack from the ground. While it has an impressive array of sensors, those sensors are focused directly on the small area of ground to be attacked. In the test scenarios the WASMs were launched as the AC-130 entered the battlespace.

The WASMs were intended to protect the flight path of the AC-130 into the area of operations, destroying ground-based threats as required. Once the AC-130 entered a circling pattern around its targets, the WASMs were to set up a perimeter defense, destroying targets of opportunity both to protect the AC-130 and to support the soldiers on the ground.

Even under ideal conditions there will be only one human operator on board of the AC-130 responsible for monitoring and controlling the group of WASMs. Hence, high levels of autonomous operation and coordination are required of the WASMs themselves. Fig. 2 shows the configuration of the simulators.

Instructors at the Hurlburt Field Special Operations Command training facility flew three scenarios, one training and two with active data collection in an AC-130 simulator. Flight paths and ground situations were played back from previous training missions with instructors filling the navigator and fire control officer positions.

In each scenario the gunship flew to an engagement area where it circled attacking a ground target. WASMs were launched and tasked using the FalconView interface which showed tracks for the AC-130, WASMs, and targets detected by either the AC-130 or the WASMs. Depending on the task

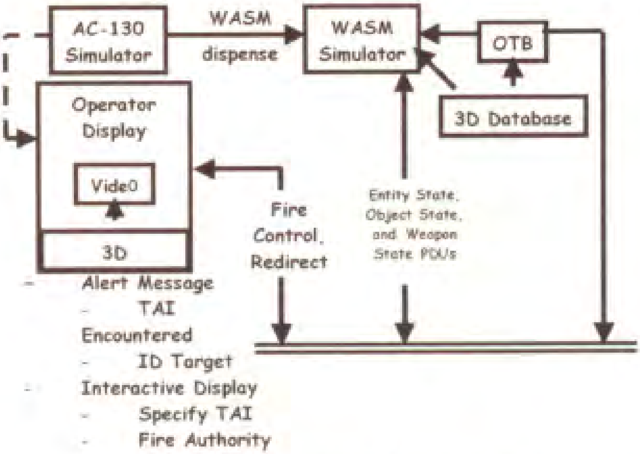

Fig. 2. Configuration of simulations.

configuration selected by the controller, WASMs either reported, returned a video image with request to authorize attack, or attacked when a target was detected by ATR. Nine targets were present in the first test scenario. The controllers launched eight WASMs in pairs to kill six of these nine targets. Two of the WASMs ran out of fuel before finding and eliminating their targets. The scenario ran for 27 min. The second scenario was similar. Nine targets were again arrayed in the region of attack and eight WASMs were launched. In scenario 2 seven of the targets were killed with one WASM running out of fuel. The second scenario ran for 30 min. Controllers were debriefed after each mission leading to observations we have classified as dealing with interface improvements, heuristic evaluation, and workload.

INTERFACE IMPROVEMENTS

Controllers found the sketch-based targeting interface (Fig. 1) easy to understand and use. Although no accidental launch was observed, the controllers felt a confirm dialog was needed to guard against that eventuality. On several occasions the controllers had wanted to redirect sub groups of munitions rather than targeting them individually or as a team. They expressed a desire for some mechanism such as the control-select convention used in Microsoft products to allow them to designate subteams to perform actions. There was general agreement that the 40 meter bounding box for target-centric commands became too small when the map was zoomed out to observe a large area making it very difficult to select targets. One controller observed that the term "OK" was misleading on the attack dialog that gave the alternatives, "Attack" or "OK". We have renamed this option "Close".

HEURISTIC EVALUATION

The controllers commented that the WASMs allowed them to search a much wider area than that covered by the standard AC-130 sensors. On several occasions they launched WASMs for general reconnaissance rather than directing them at potential targets. They felt the WASMs could provide badly needed complementary ISR (intelligence, signals, reconnaissance). Because AC-130s are primarily flown on night missions due to their vulnerability during the day they generally lack the range provided by EO (electro-optical) sensors. The forward looking infra red (FLIR) they use at night has a more limited range and could be supplemented by using WASMs as forward eyes. It was pointed out that camera video would be needed if the WASMs were to be used for battle damage assessment. The controllers liked the feature of showing tracks picked up by the AC-130s sensors on the FalconView interface but felt that tracks picked up by WASMs should be shared with other onboard targeting and navigation systems as well.

WORKLOAD

A variety of comments pertaining to workload were recorded during the debriefing. WASM control was felt to impose very little additional workload during the scenarios that were run. Despite this low workload, there were occasions in each scenario when the responsibility for controlling the WASMs was handed off between the navigation and fire control officers. All thought that the workload associated with controlling WASMs would be much higher in scenarios involving interaction with friendly forces on the ground. There was a consensus among the instructors that the electronic warfare officer was the least loaded of the crew and the best candidate for handling an additional task such as WASM control.

This test demonstrated that up to eight WASMs could be controlled by a single operator provided the vehicles are granted sufficient autonomy and control consisted of supervision and direction rather than moment-to-moment operation. Our sketch-based interface was shown to be easy to learn and use at least for those already familiar with FalconView. A more advanced version of this interface will be used to launch and direct a live P-LOCAAS prototype that will fly a mission with three simulated teammates in summer 2005. In this follow on work simple heuristics and standard search patterns have been replaced by Machinetta (Scerri et al., 2004a,b), a multiagent teamwork infrastructure that provides the ability to instantiate and execute team-oriented plans. This will allow WASMs to perform battle damage assessment for one another, stage simultaneous attacks on a target, and perform other coordinated activities that could multiply the effectiveness of such munitions. Although the P-LOCAAS test flight will use the current sketch-based FalconView interface new interface techniques and control concepts will be needed to control the larger teams of WASMs envisioned by military planners (Vick, Moore, Pirnie, & Stillion, 2001).

HUMAN FACTORS FOR LARGE-SCALE UAV TEAMS

While it is feasible for a human to direct and monitor a relatively small number of UAVs using an interface such as ours, the operator rapidly becomes saturated as the number of platforms increases. Miller (2004) analyzed the workload involved in target confirmation requests to authorize UCAV weapons release and concluded that under anticipated detection rates an operator would likely become overloaded controlling as few as 13 UCAVs doing nothing but target confirmation.

We have been investigating approaches that might help human commanders control much larger UAV teams. We assume that control through mission planning, redirection or redefinition of tactical regions, changes in plan libraries, or changes in rules of engagement as exercised through the FalconView interface do not pose a threat to operator workload because they are independent of number of UAVs. The tasks of monitoring team performance and intervening when trouble is detected, by contrast, are expected to increase rapidly in difficulty with the size of the team. We do not expect human operators to be able to effectively monitor teams of hundreds of UAVs. Instead, we believe some annunciation scheme is needed to allow the UAVs to draw the operator's attention to potential problems. To do this, the team must identify situations where human input might be needed and explicitly transfer responsibility for making that decision to a human. These decisions will typically require projections into the future or global judgments that are not considered by the reactive teamwork algorithms. We have identified three types of potential coordination problems likely to be susceptible to detection and resolution:

1. *Unfilled task allocations.* Role allocation can be allowed to continue, be suspended for some time, or its associated plan can be cancelled. If a human is not available to make the decision, the agent will autonomously suspend allocation for a period.

2. *Untasked team members* may be symptomatic of the team not effectively positioning resources to achieve current and future objectives. There are two things that can be done when a team member does not have a task for an extended period: do nothing or move the agent to some other physical location. If a human is not available to make a decision, the agent will autonomously decide to do nothing.

3. *Unusual plan performance.* Team plans and sub-plans, executed by team members to achieve goals and sub-goals will typically have logical conditions indicating when the plan has become unachievable or irrelevant. We currently allow a plan designer to specify an expected length of time and bring to the attention of the human plans that exceed this expected time. When a plan execution does not meet normal performance metrics, there are two things that can be done: cancel the plan or allow it to continue.

Because a human may or may not be available and meta-reasoning decisions must be made in a timely manner or the value of the decision is lessened, responsibility for this decision is determined through a transfer-of-control strategy, a pre-planned sequence of actions either transferring control of a meta-reasoning decision to some entity or taking an action to buy time. Mathematical models of transfer-of-control strategies are presented in (Scerri, Pynadath, & Tambe, 2002; Scerri et al, 2004a,b) and capture intuitions such as the increasing appropriateness of terminating a long running plan as it continues to run.

We have conducted preliminary experiments to evaluate how the underlying algorithms work in finding potential team problems and dealing with the possibility that a human is not available to make these decisions when they arise. The interfaces were augmented for this experiment with code that made decisions at various lags to simulate human performance.

These "human" decisions were made between 5 s and 2 min after control was transferred provided the "human" is not occupied with another task. The experiments involved a team of 80 WASMs operating in a large environment. The primary task of the team was to protect a manned aircraft by finding and destroying surface-to-air missile sites spread around the environment. Half the team spread out across the environment searching for targets while the other half stayed near the manned aircraft destroying surface-to-air sites as they were found near the aircraft. Plans were simple, requiring a single WASM to hit each found target. If a target was not hit within 12 min of being found, this was considered abnormal plan execution and meta-reasoning was invoked. Meta-reasoning was also invoked when a WASM was not allocated to hit any target for 20 min. Finally, meta-reasoning was invoked when no WASM was available to hit a found target. Two simulated commanders were available to make meta-reasoning decisions.

Six different scenarios were used, each differing the number of surface-to-air missile sites. Each configuration was run ten times. As the number of missile sites increases, the team will have more to do with the same number of WASMs, thus we can expect more meta-reasoning decisions. Fig. 3 shows that the total number of meta-reasoning decisions does increase slightly with the number of targets. Over the course of a simulation, there were around 100 meta-reasoning decisions or about one per agent and slightly less than one per minute. However only about 20% of these were transferred to a simulated human. The large number of decisions that were made autonomously was primarily because simulated humans were busy and not available to make those decisions, precisely the eventuality the transfer-of-control strategy was designed to address.

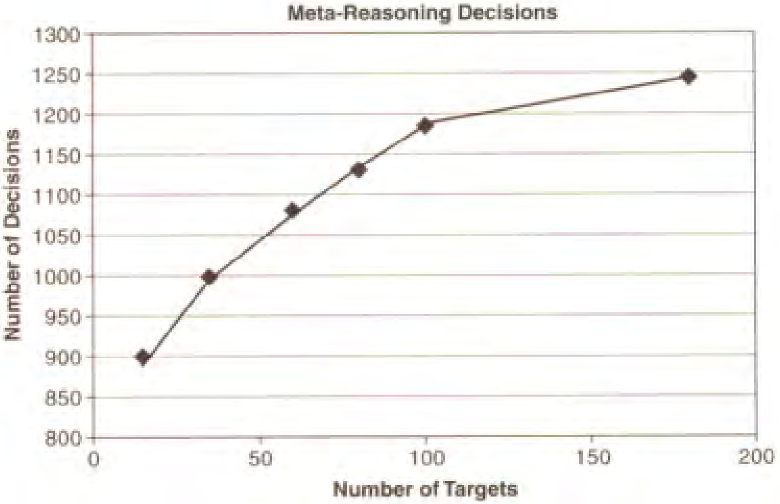

Fig. 3. Meta-reasoning decisions and number of targets (total over 10 trials).

PARAMETRIC CONTROL FOR LARGE TEAMS

Due to the high computational complexity of coordination, critical coordination algorithms typically use heuristics that are parameterized and need to be tuned for specific domains for best performance. For example, different coordination configurations might be required for different rates of change in the world, individual failure rates or communication bandwidth availability. A coordination configuration specifies parameter values for a team's coordination algorithms such as the time-to-live for tokens being passed between team members or the expected utility threshold that must be exceeded before sending a message. When several coordination algorithms are used together, e.g., algorithms for task allocation, communication and planning, the performance of one algorithm will likely affect the performance of the other algorithms, thus tuning parameters of the individual algorithms must be performed together. Due to the non-determinism of environments and coordination algorithms and the sensitivity of performance to circumstances these relationships are highly non-linear. They are also highly variable even for the same configuration. In order for operators to configure and control teams effectively we are developing methods to create a *team performance model* to capture the relation between the environment, team configuration parameters, and measures of performance. To create this concise model from data we are using genetic algorithms to learn a dynamic neural network (Polvichai & Khosla, 2002).

Searching the team performance model to find the combinations of input parameters that result in the desired output allows operators to specify performance tradeoffs and rapidly find a configuration that best meets those constraints. Since not all parameters are configurable, e.g., the observability of the domain cannot be changed during execution, we cannot simply use back propagation of the neural network to find input parameters that meet our output requirements. Instead we perform a search over the changeable configuration parameters to find a configuration that best meets the required performance tradeoffs.

In initial experiments we have demonstrated the ability of an operator to control the global behavior of a large team using a team performance model to guide actions.

The user configures the team at the start of the mission. Performance measures from the simulation are graphically displayed on the user interface at every time step. When performance changes are requested the offline features of the team performance model are used to find suitable reconfigurations. The user interface and reconfiguration assistance were evaluated over 10 scenarios. Scenarios were selected to provide situations that would require users to reconfigure their team in order to meet performance targets. For example, in a mission involving a very large team of 300 agents the user might be requested at some point in the mission to reduce the number of messages per agent or increase the number of plans instantiated. Performance measures are recorded throughout the execution. Each scenario was run for 250 time steps, with each step taking 5 s. The data presented here represent 4 h of runtime with a user in the loop. One scenario with a team of 200 agents is shown in Fig. 4. For the first intervention, the user is asked to

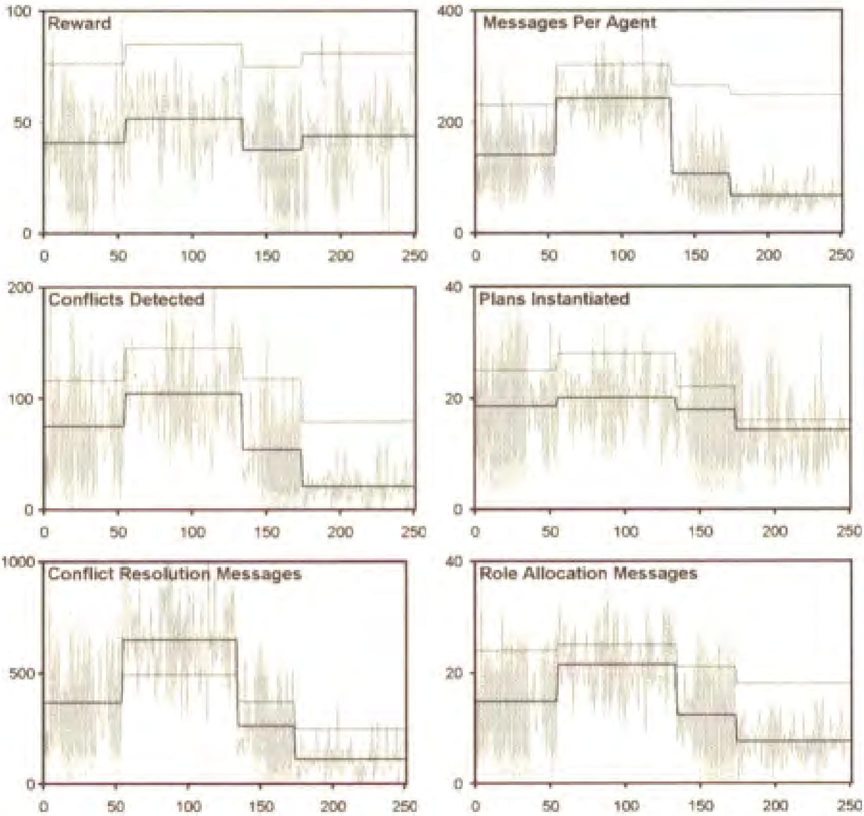

Fig. 4. Six performance measures are shown with an initial configuration and three reconfigurations during execution. The light lines show the values predicted by the model. The dark lines show the average values obtained. The jagged lines show the observed values.

increase level of rewards (goal linked outcomes such as target identifications) obtained by the team disregarding other performance measures. Using the output-to-input feature of the team performance model the user finds a new coordination configuration that increases reward performance and reconfigures the team. In the second intervention the operator must reduce network communication bandwidth by limiting the time-to-live for information tokens to 2 hops requiring further team reconfiguration to lessen the degradation in performance. For the final intervention, the user must reconfigure the team to increase the accumulation of "rewards". Results for six of the performance measures are shown in Fig. 4. The bold lines show average values for the configured system while the lighter lines indicate the values predicted by the output-to-input mappings from the team performance model. The jagged lines show the moment-to-moment variation in the actual performance measures. Despite the high variability of team performance measures the model accurately predicts the direction of effects of reconfiguration on average performance values across all six measures. By demonstrating the team performance model's effectiveness for predicting the effects of team configurations these tests demonstrate the potential of our approach for both the initial configuration of UAV teams and supervisory control over executing teams.

CONCLUSION

In this chapter we have examined difficulties involved in controlling large teams of WASMs. Some forms of interaction, particularly those that specify goals such as waypoints or tactical areas of interest, appear largely immune to scaling problems. Others, such as changes in the plans to be executed whether directly controlled as in Playbook (Miller & Parasuraman, 2003) or indirectly through mission phases (Endo et al., 2004) are design time problems that if properly engineered should impose minimal workload at runtime. There appears to be a consensus among researchers (Miller, 2004; Nickerson & Skiena, 2005; Scerri et al., 2004a,b; Crandall, Nielsen, & Goodrich, 2003) that monitoring should be directed by annunciation from the platforms and that strategies that require servicing of individual robots are most likely to limit the size of teams. One alternative proposed by Nickerson and Skiena (2005) is to control UAVs through call centers. By sharing service requests among operators rather than assigning them fixed teams, the call center could balance the load as subteams of UAVs move between areas of low and high target densities. This solution, however, could aggravate problems with situation awareness that are likely to arise as operators are forced to shift attention among a large number of platforms. This is especially significant because the advantages human control is supposed to bring the team are largely associated with providing context whether it is considering the potential for collateral damage in authorizing an attack or the decision to abandon a target because there are unlikely to be sufficient forces remaining after current attacks are completed.

NOTES

1. The P-LOCAAS or powered low cost autonomous attack system is the prototypical wide area search munition.

ACKNOWLEDGMENT

This work was supported by AFRL/MN and AFOSR grant F49620-01-1-0542. We would like to acknowledge the important contributions of our collaborators, Rob Murphy and Kevin O'Neal of AFRL/MN, Doug Zimmerer and Rolan Tapia of Lockheed Martin, and University of Pittsburgh research programmers, Joe Manojlovich and Paul Arezina.

REFERENCES

Crandall, J., Nielsen, C., and Goodrich, M. (2003). Towards predicting robot team performance. *Proceedings of IEEE international conference on systems, man, and cybernetics*, 906–911.

Cummings, M. (2004). Automation bias in intelligent time critical decision support systems. *AIAA intelligent systems conference*, AIAA, 2004-6313.

Endo, Y., MacKenzie, D., & Arkin, R. (2004). Usability evaluation of high-level user assistance for robot mission specification. *IEEE Transactions on Systems, Man, and Cybernetics*, *34*(2), 168–180.

FalconView Website. Retrieved March 14, 2005 from http://www.falconview.org

Miller, C. (2004). Modeling human workload limitations on multiple UAV control. *Proceedings of the 48th annual meeting of the Human Factors and Ergonomics Society*, Santa Monica, CA: HFES, 526.

Miller, C., and Parasuraman, R. (2003). Beyond levels of automation: An architecture for more flexible human-automation collaboration. *Proceedings of the 47th annual meeting of the Human Factors and Ergonomics Society*, Santa Monica, CA: HFES, 182–186.

Nickerson, J. V., and Skiena, S. S. (2005, January 3–6). Attention and Communication: Decision scenarios for teleoperating robots. *Proceedings of the 38th annual Hawaii international conference on system sciences.*

OneSAF Website. Retrieved January 18, 2004 from http://www.onesaf.org

Polvichai, J., and Khosla, P. (2002). An evolutionary behavior programming system with dynamic networks for mobile robots in dynamic environments. In *Proceedings of 2002 IEEE/RSJ international conference on intelligent robots and system*, *1*, 978–983.

Roth, E., Hanson, M., Hopkins, C., Mancuso, V., and Zacharias, G. (2004). Human in the loop evaluation of a mixed-initiative system for planning and control of multiple UAV teams. *Proceedings of the Human Factors and Ergonomics Society 48th annual meeting*, 280–284.

Scerri, P., Xu, Y., Liao, E., Lai, G., Lewis, M., & Sycara, K. (2004). Coordinating large groups of wide area search munitions. In: D. Grundel, R. Murphey & P. Pandalos (Eds), *Recent developments in cooperative control and optimization*. Singapore: World Scientific.

Scerri, P., Sycara, K., and Tambe, M. (2004). Adjustable Autonomy in the Context of Coordination. In *AIAA 3rd unmanned unlimited technical conference, workshop and exhibit*.

Scerri, P., Pynadath, D., & Tambe, M. (2002). Towards adjustable autonomy for the real world. *Journal of Artificial Intelligence Research*, *17*, 171–228.

Vick, A., Moore, R., Pirnie, B., & Stillion, J. (2001). *Aerospace operations against elusive ground target*. Santa Monica, CA: Rand Corp.