Teamwork Coordination for Realistically Complex Multi Robot Systems

Michael Lewis and Jijun Wang
School of Information Sciences
University of Pittsburgh
Pittsburgh, PA 15260

ml@sis.pitt.edu jiw1+@pitt.edu

Paul Scerri
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

pscerri@cs.cmu.edu

ABSTRACT

In this paper we examine human tasks in controlling UAV teams by considering their computational complexity in the number of UAVs. This analysis suggests that controlling or following fine grained cooperation among many subteams of UAVs is the most difficult task and probably beyond human capabilities for most sizes of teams. We introduce Machinetta, a teamwork system that automates fine grained level of cooperation through proxies and uses a heuristic approach based on small world networks to allow coordination of large teams. We describe an interface based on the FalconView pfps system that has been successfully used to control small teams of UAVs. We describe likely obstacles to human control of larger teams and introduce a 3 tiered architecture in which a commander performs O(1) control tasks, a group of secondary operators perform O(n) control tasks on a call room basis, and a teamwork system such as Machinetta automates the O(mn) coordination tasks.

1.0 INTRODUCTION

Effective military use of robotics will depend on coordination of large teams. Automating an aircraft squadron, for example, would not only require automating the piloting of planes but also leadership functions, support roles, and incidental forms of cooperation such as battle damage assessment or escort functions. To illustrate anticipated scales one scenario proposed by RAND [1] using LOCAAS (low cost autonomous attack system) calls for the use of up to 1,000 UAVs (uninhabited aerial vehicles) in a massed attack..

Because some functions such as selection of targets or authorization to attack may doctrinally require human input evaluating the operator's span of control as the number of controlled entities scale is critical for designing feasible human-automation control systems. Current human span of control limitations are severe. Miller [2], for example, showed that under expected target densities, a controller who is required to authorize weapon release for a target identified by a UCAV (unmanned combat air vehicle), could control no more than 13 UAVs even in the absence of other tasks. A similar breakpoint of 12 was found by [3] for retargeting Tomahawk missiles. Recent AFRL studies [4] target an even more modest 4 UAVs/operator. Similar numbers (3-9) [5] have been found for UGVs (unmanned ground vehicles). To extend these numbers to large scale teams will require breaking new ground in redefining the role of the operator and taking advantage of new forms of automation. Roles traditionally reserved for a human commander such as verifying each UCAV-found target prior to weapon release [2,6] may become infeasible in rapidly evolving missions as the size of teams increases unless novel control architectures can be developed.

To extend operator span of control to large teams we must consider how control difficulty for different control tasks grows with increases in team size. Authorization for weapon release after operator verification of each

UAV-detected target, for example, is O(n) because demand increases linearly with the number of UAVs to be serviced. Anther form of control such as designation of an attack region by drawing a box on a GUI is independent of the number of UAVs and so is O(1). Practical applications are likely to require some mixture of control regimes. In our prior work with wide area search munitions [6], for example, the operator specified search and jettison areas and ingress and egress routes , O(1), but was also required to authorize attacks and allowed to command UAVs directly, both tasks of O(n) difficulty. Examined from this perspective the most complex tasks faced in controlling large teams are those that involve choosing and coordinating subgroups of UAVs. Simply choosing a subteam to perform a particular task (the iterated role assignment problem), for example, has been shown to be O(mn) [7]. Taking task complexity into consideration can be helpful in designing human-automation control regimes. Even the simple categorization just outlined demonstrates that to scale beyond ~10 UAVs a control architecture must either eliminate, automate, or distribute human tasks that are O(n) or higher.

Our research focuses on human control in the range of 4 to 200 cooperating UAVs. Although various control strategies might be effective at the lowest end of this continuum (4-8) most of the team sizes will depend upon automating O(mn) tasks involving subteam formation and cooperation. These O(mn) tasks are required for the temporary formation of subteams to perform cooperative actions such as flush and ambush, simultaneous attack, or battle damage assessment (BDA). Without this capability UAV teams could fly to target areas and forage independently but could not perform the full range of cooperative missions that might be required.

This problem of automating cooperation and the infeasibility of human control for even moderate values of N is a generic problem that must be solved to realize the "self-synchronization" objective of Network Centric Warfare [8]. Our research is developing a general model for this process rather than isolated solutions such as "ring of fire" automated target assignment demonstrated in Fleet Battle Experiment Alpha [8]. The crucial research problems addressed are: 1- automating cooperation among a large number of agents in a computationally tractable way and 2- making this team of cooperating agents responsive to their commander's intent.

1.1 Coordination Architectures and human controllability

The choice of coordination architecture goes a long way toward determining the robustness, flexibility, and human controllability of a robotic team. Of the commonly used architectures only *teamwork* (STEAM [9]; Machinetta [10]) and market-based (Centibots [11], Traderbots [12]) architectures allocate roles and support subteam coordination. Swarms (Swarms [13], swarm-bot [14]) and behavior-based (MissionLab [15], ALLIANCE [16]) architectures achieve coordination solely through individual behavior while model-based (MultiUAV [17], linear programming [18]) architectures optimize across the whole team.

While large UAV teams need substantial autonomy a human is needed somewhere in the loop if for nothing else than to set goals for the team to accomplish. In some coordination architectures even commanding particular actions may be difficult. Roth et al. [19], for example, describe an "intent matrix" interface for specifying the relative value of target classes to a model-based architecture. Particular targets could only be selected indirectly by changing class values and computing new plans until one including the desired target(s) was found. Simple O(1) forms of control setting goals for the entire team such as designating sets of potential targets or search zones are easy within any of the architectures but O(n) commands such as directing a UAV through a waypoint may become difficult. *Teamwork* architectures such as Machinetta are particularly amenable to human control because their explicit representation of plans, roles, and information offer a variety of levers for influencing behavior. Human intentions can be expressed through goals, waypoints, explicit human roles in team plans, or instantiation of new plans.

1.2 Teamwork and scalability

Our *teamwork* approach is based on *team oriented plans* (TOPs) which describe joint activities to be performed in terms of the individual *roles* to be performed and any constraints between those roles. TOPs are instantiated dynamically from TOP templates at runtime when preconditions associated with the templates are filled. Typically, a large team will be simultaneously executing many TOPs. A team of UCAVs, for example, might execute a variety of attack TOPs. When a UCAV identifies a target in an open area it might instantiate a simple attack TOP and send out a request to fill second attacker and BDA roles. After the roles are filled two UCAVs attack the target and the third follows to record the damage (Figure 1). Another UCAV spotting a convoy of trucks near cover might instantiate a more complex simultaneous attack plan requiring filling multiple attacker roles in order that they might attack together to catch the convoy in the open. Constraints between these roles will specify interactions such as required execution ordering and whether one role can be performed if another is not currently being performed.

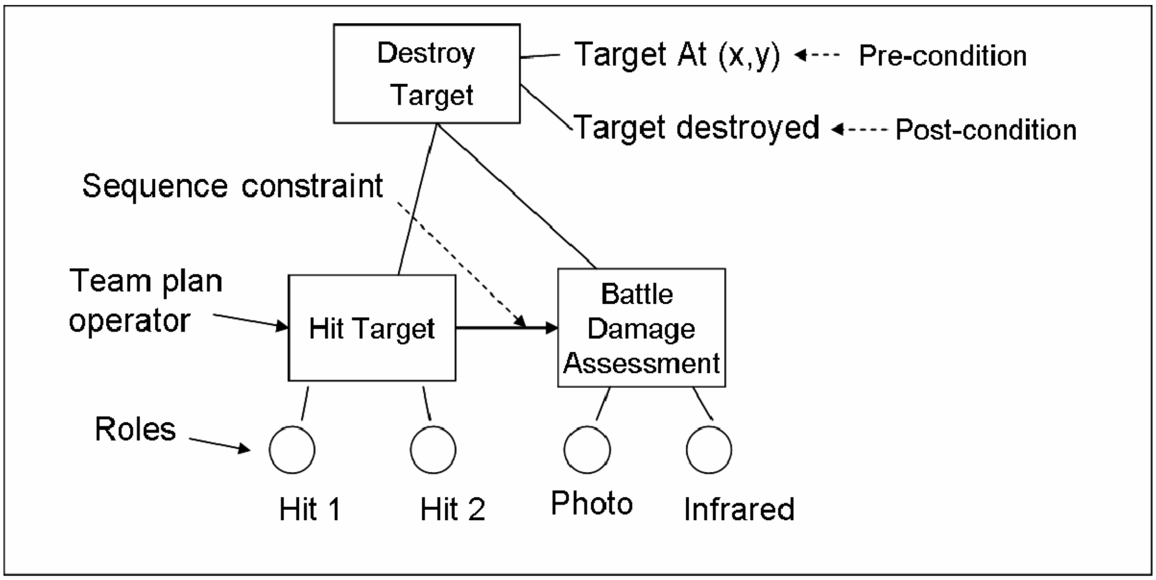

Figure 1. Team Oriented Plan (TOP) for two attackers and BDA

This level of cooperation, however, is no easier for machines than for people. If team members must maintain accurate models of one another as is necessary in conventional *teamwork* implementations the problem becomes NP hard making the coordination of more than ~20 UAVs computationally intractable. In recent work [20], however, we have found a heuristic solution that allows scaling *teamwork* algorithms to very large teams. Our algorithms make use of the *small world* property of large networks. Each member of the network maintains communications with a small number of associates. Connections between associates are used to move information around the network. While executing a plan members filling its roles maintain accurate models of one another but when the plan terminates they revert to exchanging messages only with their permanent associates. While this scheme can no longer guarantee optimal coordination, roles are filled, plans are deconflicted, and other operations performed correctly with high probability. If conflicting plans, to attack the same target were instantiated by different UCAVs, for example, the conflict would likely be discovered when some common acquaintance hears about both plans. The acquaintance would then instantiate a plan for resolving the conflict and execute that plan leading to the termination of one of the competing attacks. In this way communications and modeling effort expended on the larger team can be kept low while subteams executing TOPs are able to maintain accurate models and full communications among themselves. Our work additionally shows that even small biases toward choosing the appropriate acquaintance to pass information to can lead to substantial increases in efficiency. If a request seeking to fill a role in an attack on a convoy for example were passed on to an acquaintance who had previously mentioned a convoy it might be more likely to reach someone to fill the role than if it had been passed off randomly. Figure 2 shows the dramatic decrease in number of messages needed as this bias improves even slightly.

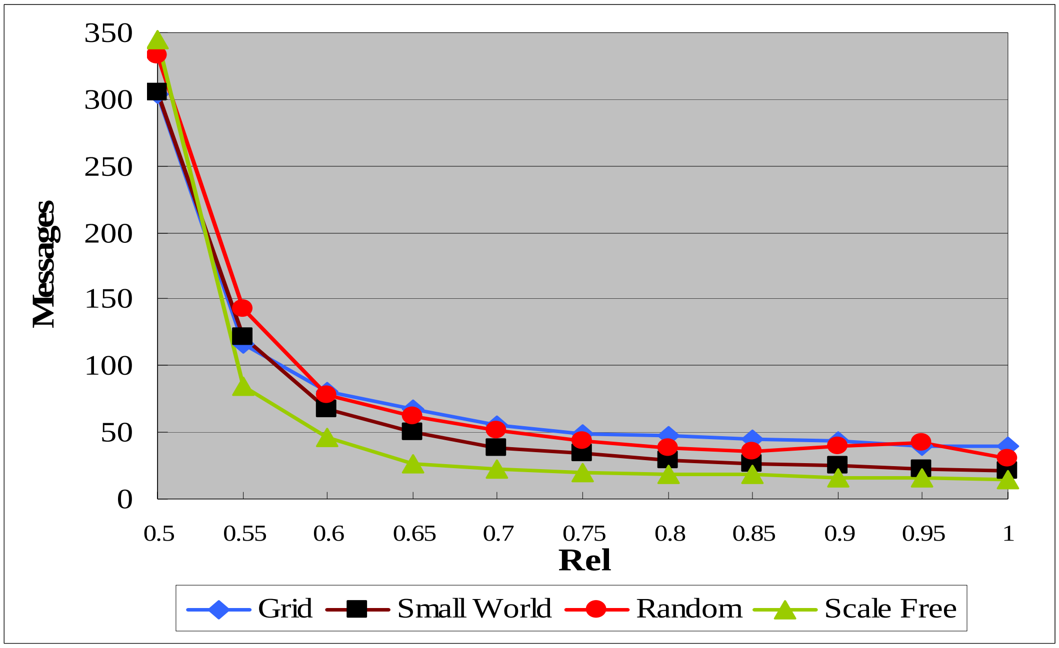

Figure 2. Messages exchanged to reach target at varying levels of relevance and network configurations

Because these *teamwork* algorithms have been shown to be effective in a range of domains they have been implemented in the form of reusable software *proxies*. Each team member has a proxy with which it works closely, while the proxies work together to implement the *teamwork*. The current version of the proxies is called *Machinetta* [10] and extends the successful *Teamcore* proxies [9]. The Machinetta software consists of five main modules, three of which are domain independent and two of which are tailored for specific domains. The three domain independent modules are for coordination reasoning, maintaining local beliefs (state) and adjustable autonomy. The domain specific modules are for communication between proxies and communication between a proxy and a team member. The coordination reasoning is responsible for reasoning about interactions with other proxies, thus implementing the coordination algorithms. The adjustable autonomy algorithms reason about the interaction with the team member, providing the possibility for the team member to make any coordination decision instead of the proxy.

2.0 CONTROLLING UAV TEAMS

Wide Area Search Munitions (WASMs) are a cross between an unmanned aerial vehicle and a munition. The first of these high concept munitions, the 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 minutes and LADAR (laser radar) with ATR (automatic target recognition) 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 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.

Our work began with a prototype interface for controlling small (4-8) teams of WASMs that has been evaluated for a AC-130 flank patrol task and was used in a P-LOCAAS flight test in the fall of 2005. Tests are currently planned to investigate operators' ability to control 32 simulated WASMs using this same interface. In the initial AC-130 tests WASM cooperation was based on simple heuristics such as choosing the nearest WASM or the one with the least fuel for an attack. Subsequent work has used Machinetta proxies although the team sizes and scenarios addressed have not yet made substantial use of the system's potential for coordination.

Our user interface for controlling small WASM teams was constructed by adding a toolbar taking advantage of drawing and other display functions of the FalconView [21], 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) [22] platoon to brigade level simulation to provide a realistic simulation of the 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. In a series of three half hour tests instructors at the Hurlburt Field Special Operations Command training facility experienced minimal difficulty in using the system from a AC-130 simulator. In each test 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. Other than agreement that the 40 meter bounding box for selecting targets needed to enlarge as the display was zoomed out to remain usable the interface was judged easy to use and allowed the operator to deploy WASMs effectively. Although minor changes were made in subsequent development the interface used to control the P-LOCAAS flight test with three simulated companions

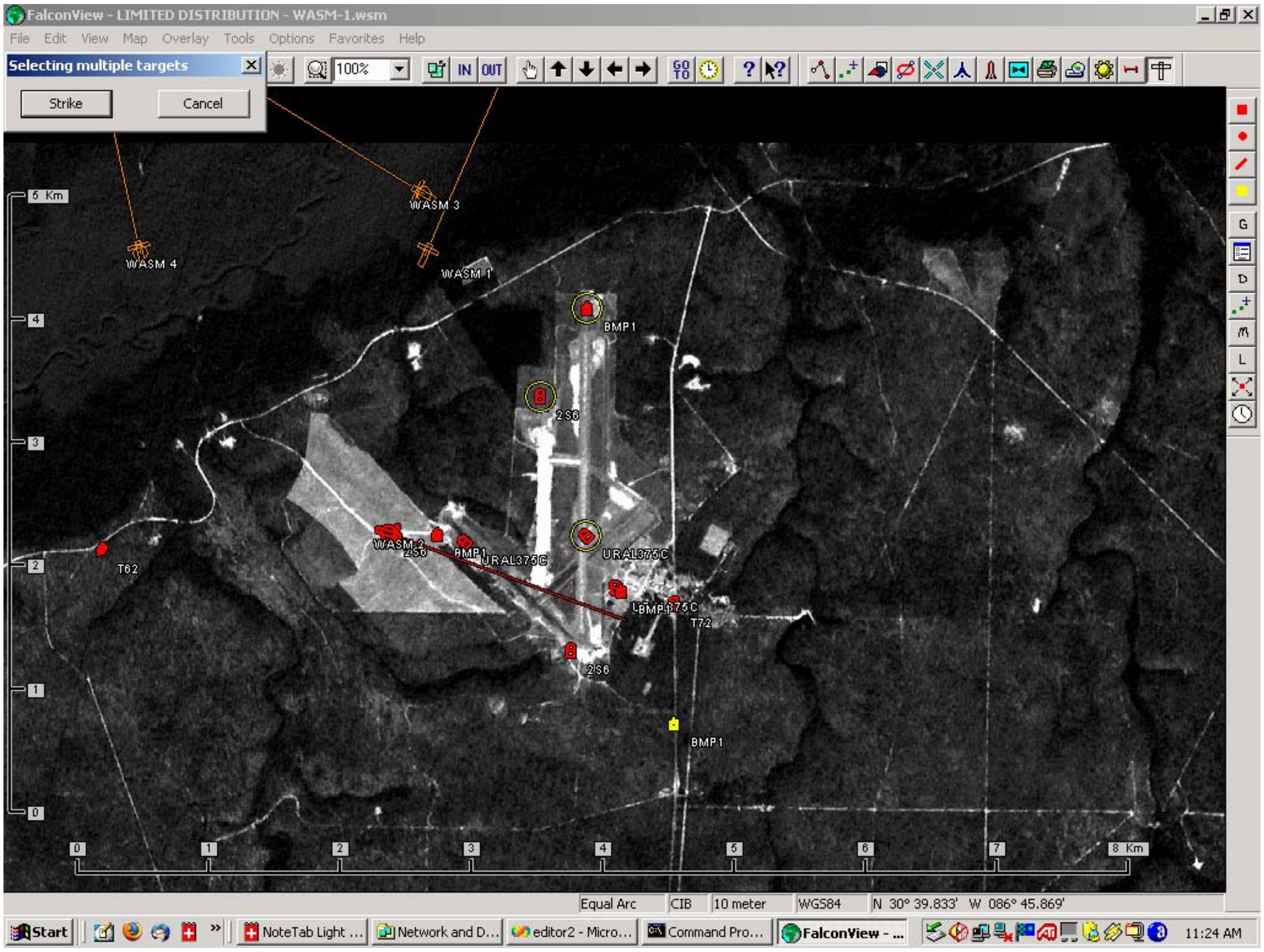

Figure 3. Falconview control interface using photo overlay. Note the labeled WASMs, waypoints, targets, and jettison area drawn onto the picture.

coordinated by Machinetta proxies remained essentially the same. Further tests of the current FalconView interface controlling 32 WASMs through Machinetta proxies are scheduled to be performed at Wright Patterson Air Force Base later this year.

3.0 SCALING TO LARGER TEAMS

We introduced this paper by considering the complexity of tasks that might need to be performed to control large numbers of UAVs. The formation and coordination of subteams emerged as the most computationally challenging activity, followed by tasks that required interacting with individual UAVs. We pointed out through illustration that the types of cooperation needed to meet NCW objectives involving "self-synchronization" typically depended on this sort of cooperation within subteams. Because algorithmic development in related work has found heuristic methods that allow automation of these forms of cooperation in a computationally tractable way, coordination of large UAV teams now appears possible. The issue of making this team of cooperating UAVs responsive to their commander's intent, however, remains unresolved. Investigating a similar fire fighting task using Machinetta to coordinate simulated fire brigades [22] found that human control frequently worsened team performance and that adding additional resources under human control did not necessarily lead to improvement. We have simulated some of the sorts of delays and disturbances that might result from human input in a medium fidelity simulation of UAV control tasks to gauge the system's sensitivity to such variation.

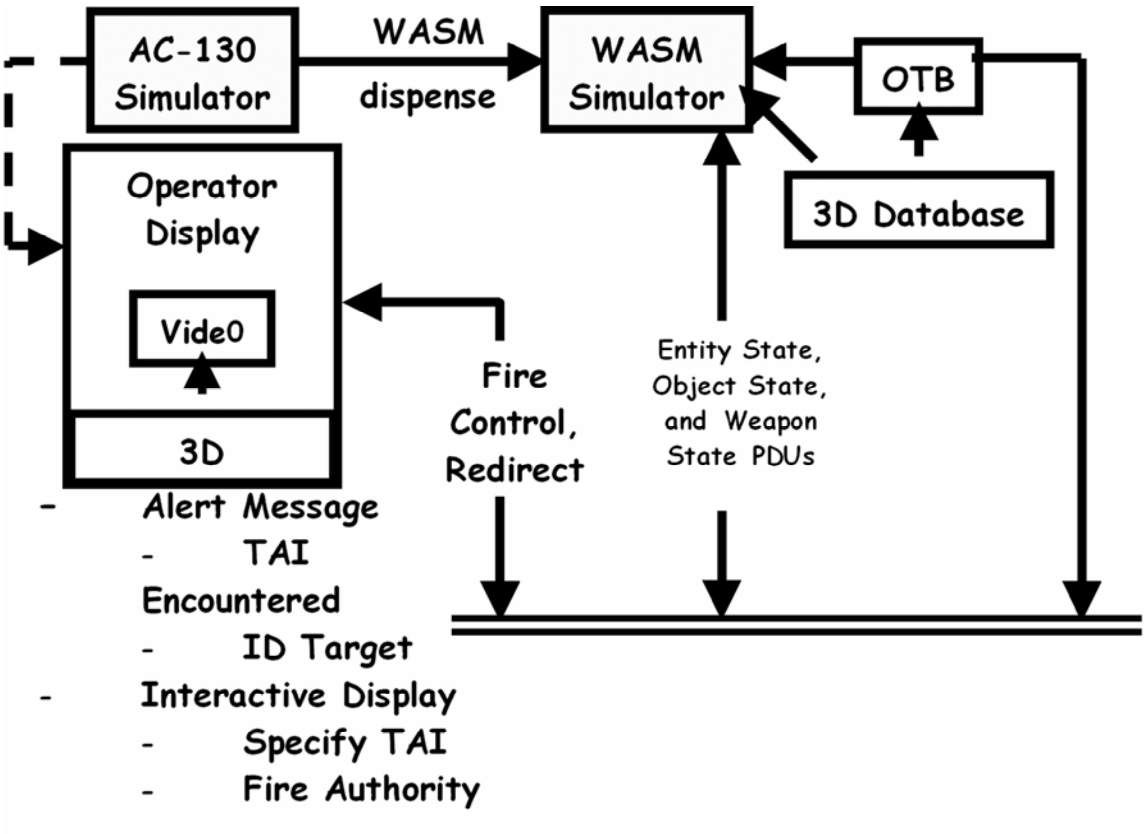

Figure 4. Information flow for AC-130 flank patrol tests

The tasks of monitoring team performance and intervening when trouble is detected are expected to increase rapidly in difficulty with the size of the team. To stay within limits of attention the team must identify situations where human input might be needed and explicitly transfer responsibility for making that decision to a human. We have identified three types of potential coordination problems likely to be susceptible to detection and resolution: unfilled task allocations, untasked team members, and unusual plan performance.
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. For this experiment decisions were made at various lags to simulate human performance.
These "human" decisions were made between five seconds and two minutes after control was transferred providing the "human" was not occupied with another task. The experiments involved a team of 80 WASMs operating in a large environment. Plans were simple, requiring a single WASM to hit each found target. If a target was not hit within twelve minutes 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 twenty minutes. Finally, meta-reasoning was invoked when no WASM was available to hit a found

target. Six different scenarios were used, each varying 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. Figure 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.

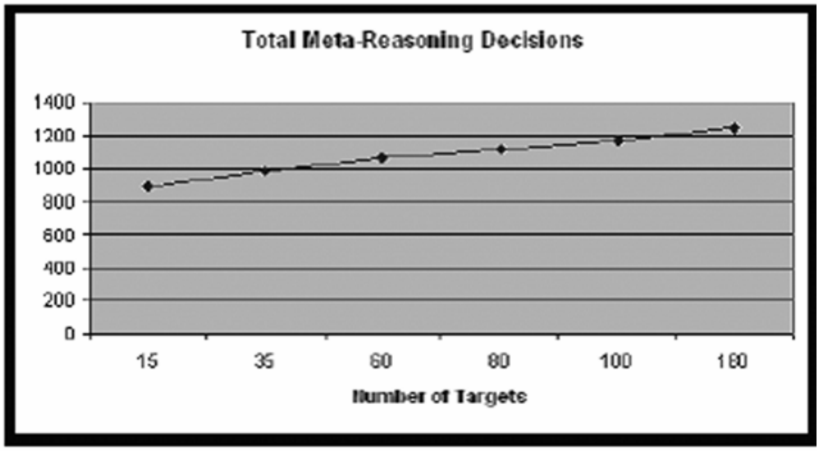

Figure 5. Meta-reasoning decisions and number of targets (total over 10 trials)

The second task involved coordination between a UGV, 4 UAVs, and 10 vehicle detecting sensors to assist the UGV in navigating through hostile territory. The simulated operator used heuristics to control the team varying his response in proactivity (act at every opportunity-never act) in one experiment and reaction time in the other. The simulation was run 50 times in each configuration.

As the results show success in traversal benefited from proactivity as did vehicle finding at the highest level of intervention. Lengthening reaction time, by contrast had little effect except at the 300 msec lag where slightly fewer vehicles were found. These results suggest that control exhibiting these sorts of human limitations should have relatively minor effects on team performance providing proper heuristics were followed. Degradation of performance to the degree reported by [22] for human operators therefore seems likely due to the actions they took rather than any hesitation in taking them.

While simulations of operators can give some indication of the expected impact of a human on team performance they tell us very little about how well such a system can be bent to achieving the operator’s intent. It is precisely this question of controllability that is crucial for developing effective UAV teams. Don Norman proposed a “seven stages of action” model picturing a closed loop in which interactions could fail if anything went wrong in perceiving, interpreting, or evaluating the state of the world on the input side or in forming intentions, planning, or executing actions on the output.

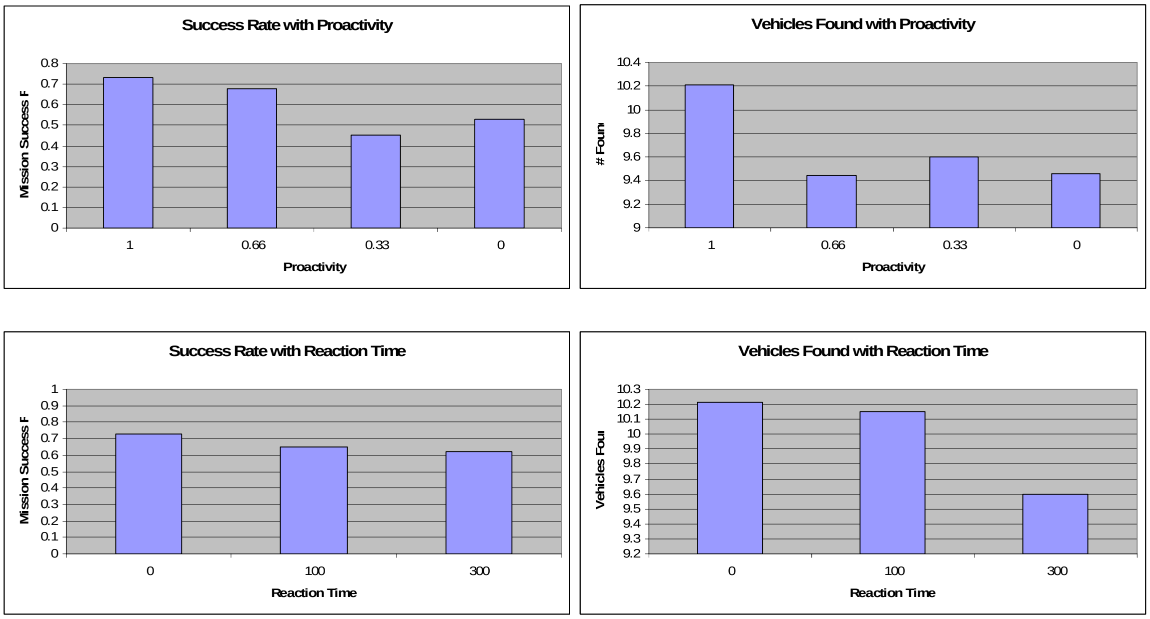

Figure 6. Team performance varying characteristics of simulated operator

Control over subteams of cooperating UAVs pose difficulties on both sides of the cycle. In the transfer of control experiment the system identified abnormal conditions in order to pass the problem on to the faux-operator. This begs the question of what the operator might do the 20% of the time he was not already busy with something else. Trying to make sense out of what 20+ perhaps overlapping cooperating subteams are doing at any one time is a cognitively challenging task. Even in far simpler environments with fewer robots researchers [23] have found it difficult to convey cooperative behavior unambiguously to an observer. As the number of vehicles increases use of color, highlighted targets, indicated paths, and other common techniques are unlikely to continue to allow an observer to separate out cooperating cliques and infer their intentions. Because monitoring and intervening to resolve unexpected events is a primary task reserved for humans in highly automated systems, an inability to comprehend the situation without relying on system issued alerts such as those in the meta-reasoning experiment would make this role particularly difficult. These problems are mirrored on the action side. Of the available levers for controlling UAV teams (altering goals, setting waypoints, response to alerts, plan instantiation, and parameter tuning) most make it difficult to connect intended behaviors with the necessary control actions. Answering questions about proper techniques for

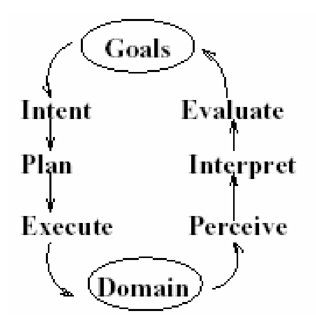

Figure 7. Norman's 7 Stages of Action

displaying large cooperating systems and guiding response where control may be indirect is going to require proper experimentation with human participants. In this process, determining those tasks humans cannot perform will be as helpful for designers as finding those they can.

Team control at other levels of aggregation should prove easier. O(1) control through team goals such as designating search areas or targets through our FalconView interface are simple and direct by avoiding the complexities of subteams and their plans. Under circumstances in which O(n) control tasks could be made independent of other tasks/UAVs, assigning additional operators could satisfy demands for human control without undermining team coordination. In a mission that required human verification of targeting prior to weapons release, for example, a team of operators could be kept standing by in the command center. As authorization requests came in they could be routed to an available operator who would examine targeting data and clear or abort the attack. Because operators would not be tied to any particular UAV or sub team of UAVs they would be available to respond to fluctuations in demand as might occur, for example, when a subgroup of UAVs overflew an enemy emplacement. In this *Call Center* approach operators respond to independent requests from a general population rather than controlling preassigned subteams. Other common O(n) tasks such as responses to requests for help would seem to follow a similar pattern. Supervisory control of the overall team would be left with the commander.

4.0 CONCLUSIONS

In this paper we have examined human tasks in controlling UAV teams by considering their computational complexity in the number of UAVs. This analysis suggests that controlling or following fine grained cooperation among many subteams of UAVs is the most difficult task and probably beyond human capabilities for teams much larger than 10. We introduced Machinetta, a *teamwork* system that automates this fine grained level of cooperation through proxies and uses a heuristic approach based on small world networks to allow coordination of large teams. We described an interface based on the FalconView pfps system that has so far been used to control small teams of UAVs but has been integrated with Machinetta to allow evaluation with larger teams in the future. In the final section we describe likely obstacles to human control of larger teams and introduce a 3 tiered architecture in which a commander performs O(1) control tasks, a group of secondary operators perform O(n) control tasks on a *call room* basis, and a *teamwork* system such as Machinetta automates the O(mn) coordination tasks. We hope that considering human multi-vehicle control tasks can lead to other novel architectures, help discriminate feasible from infeasible solutions, and lead to improved designs.

5.0 REFERENCES

[1] Vick, A., Moore, R., Pirnie, B., Stillion, J. Aerospace Operations Against Elusive Ground Targets, RAND Documents, 2001.

[2] Miller, C. Modeling human workload limitations on multiple UAV control, Proceedings of the Human Factors and Ergonomics Society 47th Annual Meeting, 526-527, 2004.

[3] Cummings, M. and Guerlain, S. An interactive decision support tool for real-time in-flight replanning of autonomous vehicles, AIAA Unmanned Unlimited Systems, Technologies, and Operations, 2004.

[4] Ruff, H., Narayanan, S. and Draper, M. Human interaction with levels of automation and decision-aid fidelity in the supervisory control of multiple simulated unmanned air vehicles, Presence, 11(4), 335-351, 2002.

[5] Olsen, D. and Wood, S. Fan-out: Measuring human control of multiple robots, Proceedings of the 2004 Conference on Human Factors in Computing Systems (CHI 2004), April 24-29, Vienna, Austria, ACM, 231-238, 2004.

[6] Lewis, M., Polvichai, J., Sycara, K., and Scerri, P. Scaling-up human control for large scale systems, In Cooke, N., Pringle, H., Pedersen, H., and Connor, O. (Eds.) Human Factors of Remotely Operated Vehicles, New York: Elsevier, 237-250, 2006.

[7] Gerkey, B. and Mataric, M. A formal framework for the study of task allocation in multi-robot systems, International Journal of Robotics Research, 23(9), 939-954, 2004.

[8] Albert, D., Garstka, J., and Stein, F., Network Centric Warfare : Developing and Leveraging Information Superiority, CCRP Publication, http://www.dod.mil/nii/NCW/, 1999.

[9] Tambe, M. "Towards Flexible Teamwork" Journal of Artificial Intelligence Research, Volume 7, Pages 83-124, 1997

[10] Scerri, P. Pynadath, D. and Tambe M. "Towards adjustable autonomy for the real-world", Journal of AI Research, Volume 17, Pages 171-228, 2002.

[11] Ortiz, C. and Vincent, R. and Morisset, B. Task Inference and Distributed Task Management in the Centibots Robotic System, in The Fourth International Joint Conference on Autonomous Agents and Multi Agent Systems, ACM, Jul 2005

[12] Goldberg, D. and Cicirello, V. and Dias M. and Simmons R. and Smith S. and Stentz A., Market-Based Multi-Robot Planning in a Distributed Layered Architecture, Proceedings of the International Workshop on Multi-Robot Systems, 2003

[13] Belta, C., and Kumar, V. Abstractions and Control Policies for a Swarm of Robots, *IEEE Transactions on Robotics*, vol.20, no.5, 865-875, 2004.

[14] Mondada F., Gambardella L.M., Floreano D., Nolfi S., Deneubourg J.-L., Dorigo M. SWARM-BOTS: Physical Interactions in Collective Robotics, *IEEE Robotics & Automation Magazine*, vol. 12, 21-28, 2005.

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

[16] Parker, L., ALLIANCE: An architecture for fault tolerant multi-robot cooperation, *IEEE Transactions on Robotics and Autonomy*, 14(2), 1998.

[17] Rasmussen, S and Chandler, P. MultiUAV: A multiple uav simulation for investigation of cooperative control, Proceedings of the 2002 Winter Simulation Conference, 869-877, 2002.

[18] Bererton, C. 2004. Multi-Robot Coordination and Competition Using Mixed Integer and Linear Programs. Ph.D. Dissertation, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.

[19] Roth, E., Hanson, M., Hopkins, C., Mancuso, V., and Zacharias, G. 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, 2004.

[20] Xu, Y., Scerri, P., Yu, B., Okamoto, S., Lewis, M. and Sycara, K. An integrated token-based algorithm for scalable coordination, Fourth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'05), 407-414, 2005.

[21] FalconView Website. Available via http://www.falconview.org [accessed August 2, 2006].

[22] OneSAF Website. Available via http://www.onesaf.org [accessed August 2, 2006].

[23] Schurr, N., Marecki, J., Tambe, M., Scerri, P., Kasinadhuni, N., and Lewis, J. The future of disaster response: Humans working with multiagent teams using DEFACTO, In *AAAI Spring Symposium on AI Technologies for Homeland Security*, 2005.

[24] Shankar, S., Jin, Y., Adams, J., and Bodenheimer, B. “Enhancing RoboFlag Users’ Situational Awareness,” Proceedings of the 2004 Human Factors and Ergonomics Society 48th Annual Meeting, 2004.