Single Operator, Multiple Robots: Call-Request Handling in Tight-Coordination Tasks*

Gal A. Kaminka and Yehuda Elmaliach

The MAVERICK Group, Computer Science Department
Bar Ilan University, Israel
`{galk,elmaley}@cs.biu.ac.il`

Summary. Many applications of robots require a human operator to supervise and operate multiple robots. In particular, the operator may be required to resolve *call requests* when robots require assistance. Previous investigations assume that robots are independent of each other, and allow the operator to resolve one request at a time. However, key challenges and opportunities arise when robots work in tightly-coordinating teams. Robots depend on each other, and thus a single failing robot may cause multiple call requests to be issued (by different robots). Moreover, when the operator switches control to a robot, its teammates must often wait idly until the call request is resolved. We contrast previous approaches with two novel *distributed* methods, where the call-request resolution is itself considered a collaborative problem-solving activity, and non-failing robots use their knowledge of the coordination to assist the operator. We empirically compare the different approaches in several scenarios involving tight coordination, where an operator seeks a dead robot in order to assist it. Extensive experiments with 25 human operators show that this new technique is superior to existing methods, in terms of reducing the time to locate the dead robot. We also show that the new method has much more consistent performance across different operators.

1 Introduction

There is need for human intervention in applications of robot teams. Teams of multiple robots can carry out mundane or dangerous tasks. However, many applications require occasional human intervention, either for safety reasons, or because the robots suffer from some failure that requires resolution by the operator. Examples of such applications include search and rescue operations [7], multi-rover planetary exploration, and multi-vehicle operations [4].

Previous work has examined centralized methods in which a single operator interacts with multiple robots, both for monitoring their activity, as well as for resolving contingencies [1, 3, 4, 8, 11]. Here, robots are assumed to operate autonomously, as direct teleoperation of all robots in parallel is impractical. Robots that require the operator's assistance initiate or are issued *call-requests*, which are queued for the operator. The operator switches control between robots, and uses single-robot teleoperation with individual robots to resolve the call requests in some (prioritized)

---

* This research was supported in part by BSF Grant #2002401 and by the Israel Ministry of Science and Technology (MOST).

sequence. This method works well in settings where the task of each robot is independent of its peers, and thus the resolution of call requests can be done in sequence, independently of other call-requests.

Unfortunately, these centralized methods face difficulties in *coordinated tasks*—tasks that require tight, continuous, coordination between the robots, i.e., robot teams where robots are highly inter-dependent. First, due to the coordinated nature of the task, robots depend on each other's execution of subtasks; thus a single point of failure (e.g., a stuck robot) will quickly lead to multiple call requests. Second, when the operator switches control to a robot, the other robots must wait for the resolution of the call-request, because their own decision-making depends on the results of the operator's intervention. As a result, robots wait idly while the call request is resolved. While monitoring and diagnosis techniques can help localize call-requests to the relevant robot [5], minimizing the duration of call-request resolution remains a key challenge.

Operating a team of coordinated robots raises the opportunity for novel resolution methods, in which the responsibility for the resolution of the call request is *distributed*. Rather than having the operator centrally take all actions to resolve a failure, the otherwise-idle robot teammates can offer assistance, e.g., in providing useful information or in carrying out sub-tasks associated with the resolution process.

For example, consider the task of controlling three robots moving in formation (a task requiring tight continuous coordination between robots). Suppose one of the robots gets stuck, and is unable to move. A call request is issued to the operator, which must identify the failure and attempt to resolve it in some fashion. Previous approaches would have the operator attempt to teleoperate the robot in an attempt to dislodge it, while the other robots are idle [1,4].

However, the operator could take advantage of the other robots to resolve the failure. First, the other robots could be used to provide video imagery of the stuck robot from various angles. Second, the robots may assist the operator to determine the location of the robots—since they can calculate its expected position with respect to their own position—based on its position within the formation.

This paper takes first steps towards allowing robots to use their knowledge of the coordination to autonomously assist the operator. We examine several variations of a distributed control methodology in which functioning members of the team, rather than switching to an idle mode of operation, actively seek to assist the operator in determining the failure. The key idea is that the responsibility for resolving the call-request is distributed among the team-members *in addition to* the operator.

We empirically evaluate these variations (and contrast them with previous approaches) in extensive experiments with 25 human operators, operating a team of 3 Sony AIBO robots. The experiments evaluate several concrete call-request scenarios, in which a stuck robot must be located by the operator. The results show that distributed call-request resolution leads to shorter failure-recovery times. Moreover, the results show that a key factor in the success of the distributed method lies in the robots' use of organizational knowledge (i.e., their knowledge of the coordination). However, even in cases where this organizational knowledge fails, the operator is able to compensate. Thus the use of our distributed approach is always better than either the operator or the robots resolving the call request by themselves. A final promising result is that the distributed methods lead to improved operator consistency, reducing the variance in performance between operators.

2 Background and Motivation

The bulk of existing work on controlling multiple robots put the operator in a centralized role in attending to robots, and do not often distinguish between different task types on the basis of the coordination involved. Indeed, many existing approaches implicitly assume that robots are relatively independent in their execution of subtasks. As a result, a centralized control scheme does not interfere with task execution. Fields [3] discusses unplanned interactions between a human and multiple robots in battlefield settings, where otherwise-autonomous robots send call requests to the human operator to ask for assistance. These call requests are queued, and the operator resolves the problems one by one. Fong et al. [4] propose a *collaborative control* system that allows robots to individually initiate and engage in dialog with the human operators. The call requests are queued based on priority, and resolved serially.

Myers and Morely [8] discusses an architecture called TIGER that uses a coordinating agent that mediates between the operator and autonomous software agents. This agent centralizes the information from all agents, and can present it to the operator (or provide it to other agents). The agent is also responsible for translating operators instructions to the team. This approach thus assumes that call requests may be resolved autonomously by the robots, given appropriate high-level commands to the team. In contrast to this approach, we believe that often, the operator must directly interact with a failing robot or its teammates to resolve a call request. We thus allow the operator to directly interact with any single robot, while others assist.

ACTRESS (Actor-based Robots and Equipment Synthesis System) [10] is a multi-agent robot architecture which incorporates an interface for monitoring and controlling robots. The operator may issue commands that affect groups or individual robots. However, ACTRESS does not utilize collaboration between the operator and robots in resolving call requests. The operator may issue commands to robots that assist in such resolution, but the robots are otherwise idle.

In contrast to the above centralized approaches, we believe that resolving call-requests is in the interests of all robots currently coordinating with the robot requiring assistance—and thus they should actively collaborate with the operator to resolve the call request. Other work has also examined distributed paradigms for human/robot interaction. Tews et al. [11] describe a scalable client/server architecture that allows multiple robots and humans to queue service requests for one another. Scerri et al. [9] describe an architecture facilitating teamwork of humans, agents and robots, by providing each member of the team with a proxy and have the proxies act together as a team. Our work differs from both of these investigations in that we do not attempt to put humans and robots on equal ground. In our current work, only the human can initiate the distribution of a task to resolve a call-request. However, once initiated, the task is carried out by all members of the robotic team and the operator.

Ali [2] compares different classes of human-robot team interaction (*Direct manual control, supervisor control, individual* and *group control*). The parameters measured are effectiveness (in term of task completion and speed of completion), safety (both for the robots and their environment), and ease of use. While we similarly evaluate different interaction methods, we focus only on the case of one operator and multiple robots. However, within those, we distinguish several different types. Moreover, we provide new distributed resolution types.

3 Distributed Call-Request Resolution

As previously discussed, centralized resolution of call requests, by the operator, may work well when robots' tasks are independent of each other. However, in coordinated tasks, many robots may have to stop their task execution until a call request is resolved, because their own task execution depends on that of the robot that requires the resolution. In such cases, it is critical to minimize the time it takes to resolve a call request.

We thus focus on a distributed control approach, whereby the robots who depend on the resolution of the call-request take active steps to resolve it, in collaboration with the operator. This approach takes advantage of the robot teamwork, by turning the resolution of the call-request into a distributed collaborative task for all involved. Moreover, the active robots (that do not require assistance) are involved in a coordinated effort with the robot requiring assistance, and thus may be in a better position to assist it.

The key idea behind this approach is that call-request resolution is best viewed as an instance of cooperative problem-solving. During task execution, robots collaborate to achieve the operator goal. If task execution is halted due to a failure, a new collaboration problem instance is generated (resolving the call-request), which should then be addressed by the team-members that are affected by the failure, since they have knowledge which they can bring to bear on the problem.

Concretely, we investigate distributed resolution in repairing broken formations of Sony AIBO 4-legged robots. Formation-maintenance tasks require tight, continuous coordination between robots [6]. When a robot fails and is unable to move, the formation cannot proceed until the failure is resolved in some fashion: Either the robot becomes unstuck, or it is declared dead and the formation proceeds without it. A stuck robot often cannot report on why it is stuck, due to sensory range limitations. For instance, in the AIBO robots, the camera (mounted in the head) cannot pan and tilt to cover the rear legs. Thus if one of them is caught by something, the robots own sensors cannot identify it. The robot must then issue a call-request for assistance. The operator, in turn, must use one of the other robots to locate the stuck robot and get video imagery of its state. This act of locating the other robot and getting sufficiently close to it is a key factor in the resolution of the call request in this case.

We contrast different resolution schemes. The first, *teleoperated* scheme corresponds to the centralized control used in previous approaches (e.g., [1, 4]). In this scheme, the operator would switch control from one active robot to the next, as deemed necessary, and manually teleoperated controlled robots (one at a time) until the disabled robot was found. When one robot is controlled, the others remain idle. Another previously-investigated approach is the fully *autonomous* scheme, that lets the active robots (but not the operator) search for the failing robot. This scheme corresponds roughly to the method described in [8], where the robots receive general instructions (here, "search!") by the operator, but are left to translate and follow these commands autonomously, without direct manipulation.

We compare these previous approaches to two variations of distributed call-request resolution. In the first (*semi-distributed*), the robots assist the operator by autonomously beginning to search for the failing robot as soon as the call request is received. The operator views a split-screen view of their video imagery, and as soon as it identifies the stuck robot in one of the displays, can switch control to the robot associated with the display. Once a robot is taken over by the operator, the others become idle. The operator may still switch control to these other robots, but they no longer work in an autonomous fashion.

The fully-distributed scheme mixes teleoperation and autonomous search all through the call resolution process. The operator may teleoperate any robot at any time, and may switch between controlled robots as needed. When not operator-controlled, the robots first head towards the expected position of their stuck peer. This position is estimated based on their knowledge of the formation (organizational knowledge), under the assumption that the robot became stuck in its previous location within the formation. If they fail to find it there, they begin a spiral search pattern that is likely to find the robot, but may take relatively long time. Thus the operator and the other robots work in parallel: The search ends when either an autonomous robot or a teleoperated robot are sufficiently near the stuck robot.

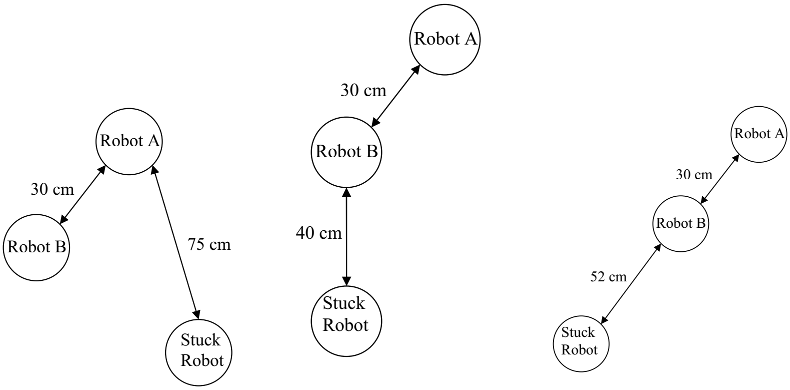
The motivation for the distributed scheme is that the robots may be able to use their knowledge of the robot's role in the formation to attempt to locate it. The robots that maintain the formation have improved chances to localize themselves with respect to the formation, then an operator which takes control of a robot in the formation, without the situational awareness of the robots. On the other hand, the operator has superior inference and vision, and may be able to locate the stuck robot in the video imagery, even in cases where the robots would be unable to do it.

4 Experimental Evaluation of Call-Request Resolution Methods

We now turn to empirically evaluate these call-request resolution schemes with human operators. We use all schemes in failure scenarios in the context of a triangular formation of three robots. In each of the failure cases, we disable one of the robots to simulate a catastrophic failure, not letting it move or communicate. In accordance with previous approaches, a call-request is then issued to determine the whereabouts of the failing robot. The robots and operator then begin the search process as described above. The search stops when any robot is within a predetermined distance of its failing teammate.

We examine three scenarios. In all, the right follower robot was disabled, and color marked to allow its detection by the other robots (called *active robots*) and the operator. A potential advantage of the distributed and autonomous schemes is that they can utilize the robots' own knowledge of the coordination to locate the stuck robot. In particular, because the robots have moved in formation prior to the call-request, they may have an easier time guessing their peer's location than the operator (who needs to orient herself in space via the teleoperated camera).

To evaluate the importance of this advantage, we varied the position of the disabled robot (Figures 1, 2): The *easy* setup placed the disabled robot at approximately where it would be had it just stopped in its tracks prior to the team getting notification of the call request, i.e., a bit farther behind its location within the formation (Figures 1-a, 2-a). The *medium* setup placed the robot behind the left follower robot (1-b, 2-b). The *difficult* setup placed the robot to the left of the left follower robot, and behind it, i.e., completely out of place compared to the formation (1-c, 2-c). Thus the locations progress from a location easily predictable by the robots, to a location unpredictable to them.



(a) Easy (b) Medium (c) Difficult

Fig. 1. The three experimental setups, distinguishing predictable, semi-predictable, and unpredictable locations of the stuck robot.

(a) Easy (b) Medium (c) Difficult

Fig. 2. AIBO robots in initial places for the three experimental setups.

We tested 25 human operators with each of the failure scenarios (22 male, 2 female; 22 of these—including the two females—were graduate or undergraduate students). All operators were novices; none had previous experience controlling multiple robots. Each operator tried all the the resolution schemes previously described, in each of the three scenarios. The ordering of the scenarios was randomized between operators to prevent biasing the results.

We distinguished two phases: The first phase of the resolution involved recognition of the disabled failure from any distance. The second phase involved its localization by another robot reaching within 35 centimeters of it. Each scenario began with the simulated disabling of the robot (and issuing of the call request), and ended with its localization by at least one robot—teleoperated or autonomous.

For each of the failure scenarios and for each method, we measure the duration of the two phases. This is an objective performance measure because the initial locations of the robots are fixed, the searching speed is constant for non-teleoperated robots, and the termination condition for the search are fixed (robots within specific distance of the failing robot). Thus other than the typical robot sensor uncertainty, performance variance is introduced solely by operator intervention. The first measured duration is that of the time that it took the operator to recognize the disabled robot in any one of the cameras (the operator uses the split-view interface in this task), i.e., the duration of the first phase. In all but the teleoperated scheme, the operator is completely passive during this interval. We then measure the time that it takes for an active robot—autonomous or teleoperated—to reach the disabled robot, i.e., the duration of the second phase. Since the motivation behind the distributed control scheme is to reduce the time spent awaiting resolution, we prefer shorter overall durations.

We begin by examining the bottom line—the total time it takes to identify the location of the disabled robot. Figure 3 shows the average total duration for the 25 operators. The vertical axis measures the time in seconds, while the horizontal axis shows the three experiment setups. In each, four bars are shown corresponding to the different resolution schemes (left-to-right: Autonomous, semi-distributed, distributed, and teleoperated).

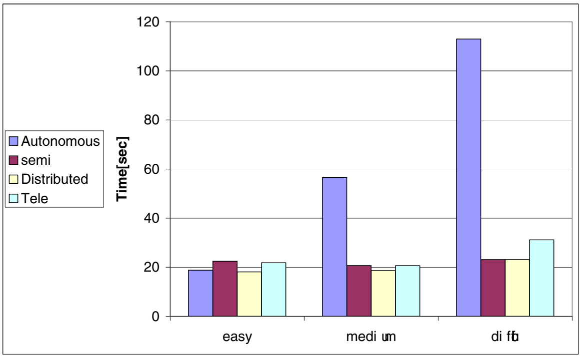


Fig. 3. Total Time to Resolution (in seconds).

The results show that in all *easy*, *medium* and *difficult* locations, the distributed approach is preferable to the both centralized teleoperation approaches, and the fully autonomous approach. Full distributed search does better than the semi-distributed approach in all locations, and better or same than the autonomous approach or same.

Overall, the distributed collaboration between the operator and active robots in the distributed approach proves to be a powerful technique for significantly reducing the time to complete the task of locating the disabled robot.

The results have been tested using a one-tailed t-test assuming unequal variances. In the easy setup, the distributed scheme is not significantly different than the autonomous scheme, and only moderately different ($p < 0.12$) than the semi-distributed and teleoperated schemes. However, as we move to the medium and difficult setups, the situation changes. The total time for the distributed scheme is significantly lower than the total time for the autonomous scheme in the latter setups ($p < 0.00004$ and $p < 10^{-12}$, resp.). The distributed scheme does better than the teleoperated scheme in the difficult setup ($p < 0.02$), and is moderately better in the medium setup ($p < 0.13$).

The figure also carries other lessons. First, the ability of the robots to use organizational knowledge of the formation can be very useful in reducing the resolution time, and thus in assisting the operator. When the stuck robot was located approximately where it was predicted to be in terms of its position in the formations, the robots were able to quickly locate it, in fact beating the operator in terms of total time (see more on this below). However, the distributed scheme was superior even in these cases, because even in where the robots were not as successful, the operator (working in collaboration with the robots) was able to compensate. This is particularly evident as the difficulty of the different setups increased, and the location of the stuck robot was unpredictable to the robots.

To better understand these results, we should consider separately the results for the first phase of the search (when an remote identification of the stuck robot was made by the operator), from the second phase, in which an active robot was to approach the stuck robot to localize it. Figure 4 shows the results of the different control schemes for the first phase, averaged across operators. The figure measures the average time (in seconds) it took the operator to recognize the disabled robot from afar, in the split-view camera display. In the autonomous approach, the operator did not intervene in the operation of the robots, only indicated that the stuck robot was recognized. In the teleoperated scheme, the operator manually turned a robot around until a heading to the remote robot was recognized.

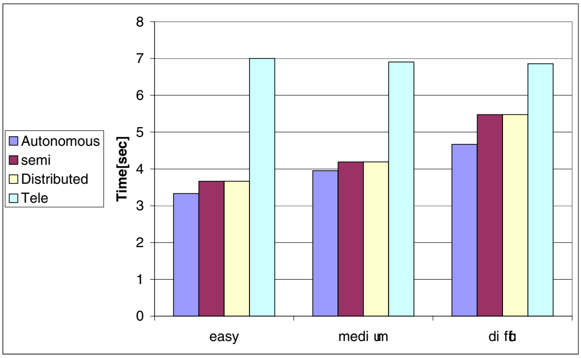


Fig. 4. Phase 1 *Time until initial (remote) identification (in seconds).*

Clearly, all approaches in which robots attempt to orient themselves towards the predicted location of the disabled are superior to a teleoperated (centralized) approach. Note that in all approaches, the operator recognizes the robot from afar. The active robots do not necessarily recognize the other robot from afar, and as we will see below, may end up searching for it in the wrong location. This significantly shorter initial recognition is a beneficial side-effect of the distributed approaches. However, the initial benefits of the robots to orient themselves towards the stuck robot is lost in more difficult settings.

Figure 4 also shows an important property of the usefulness of human operators: Human ability to recognize the robot from afar is virtually identical in all three difficulty settings. Thus humans bring to bear consistent robust (if slow) capabilities. These can be useful in real applications, where the stuck robot may be partially hidden behind obstacles or otherwise not visible at all to the robots.

An examination of the second phase of the search (once an approximate heading towards the stuck robot is determined) is also telling with respect to this issue. Figure 5 shows the results for this phase, where the task is to arrive within the proximity of the disabled robot. Despite its poor performance in phase 1, the teleoperated approach does quite well in phase 2. This is easily explained—here the disabled robot is already recognized, and the teleoperated approach simply allows the operator to now drive the teleoperated robot as quickly as possible, outrunning automatic approaches that move in constant (and typically conservative) speed. Thus again, the operator brings to bear capabilities that cannot be duplicated by the robots.

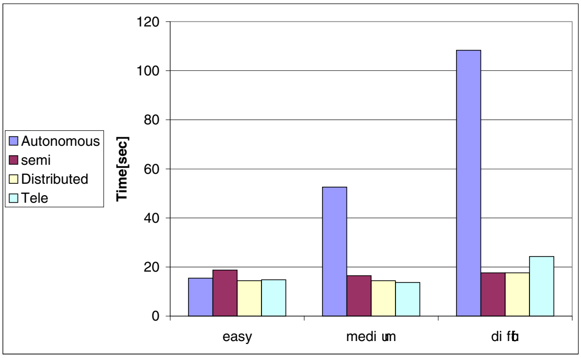


Fig. 5. Phase 2 *From initial identification to localization of the stuck robot (in seconds).*

However, the best performances was by the distributed approach, because it essentially turns this phase into a race between a teleoperated robot and an autonomous robot, as to who gets to the disabled robot first. Moreover, unlike the semi-distributed approach, where there's an overhead of a few seconds while the operator takes over control (see the results for the easy/medium location), here the transition from phase 1 to phase 2 is fairly smooth, because one active robot continues to search even while the operator is taking over control of the other. Thus there is here a composition between the *Autonomous* approach and the *Teleoperated* approach.

Indeed, contrasting the results of the *Autonomous* and *Distributed* approaches is telling. As we move from the easy location to medium to difficult, the gap between the methods is grows in favor to the *Distributed* approach. That happens as a result of the inability of the *Autonomous* approach, to locate the stuck robot in unpredictable places. The collaboration between the human operator and the robot team is superior to either, alone.

An final lesson is revealed by examination of the standard deviation of the results for total task-completion time. Table 1 shows the standard deviation for the different approaches, in the three experiment setups. Each row corresponds to a different method, and each column to different setup. We can see that in the easy setup, the autonomous, semi-distributed, and distributed schemes all have essentially the same standard deviation, indicating similar performance. However, the standard deviation for the autonomous scheme in the medium setup is much higher than for the other approaches. In the hard setup, both the autonomous and teleoperated approaches suffer from greater standard deviation in performance than the two distributed schemes. This shows an additional benefit of the distributed methods: A more consistent performance of operators in the distributed and semi-distributed cases.

	Easy	Medium	Difficult
Autonomous	11.21	34.64	23.82
Semi-Dist.	11.30	5.07	7.78
Distributed	11.29	5.16	7.90
Teleoperated	7.68	5.96	15.87

Table 1. Standard deviation of call-resolution times (in seconds).

5 Summary and Future Work

This paper explores novel first steps towards distributed call-request resolution schemes, in which the operator and robots collaborate to resolve failures. This scheme is particularly suited to situations where robots are tightly coordinated, and thus are able to use their knowledge of the coordination to effectively assist the operator. The technique builds on a key idea, that the resolution of failures in cooperative tasks should be viewed as a cooperative task in itself. Previous techniques (teleoperation of one robot at a time, autonomous operation of the robots) were meant for tasks that do not require tight coordination between the robots.

We empirically evaluate this new technique and compare it to previous work, in extensive experiments with 25 human operators. The results show that the distributed control scheme, exploiting teamwork between the operator and all robots, reduces the time of resolving failures (compared to the centralized and autonomous approaches), and was superior in all cases. Moreover, the technique leads to reduced variance between operators. However, its overall improvements with respect to the centralized teleoperated approach was only significant in a subset of the experimental conditions.

The promising results presented in the paper also raise important questions for future work. We are particularly interested in integrating the new technique with complete human-robot interaction systems, in order to evaluate its effectiveness not only within call-request resolution, but in more general settings of operating the robots even in non-failures cases.

Acknowledgments

We thank the anonymous reviewers and Avi Rosenfeld for their helpful comments. Thanks to K. Ushi.

References

1. J. A. Adams. *Human Management of a Hierarchical System for the Control of Multiple Mobile Robots*. PhD thesis, University of Pennsylvania, 1995.
2. K. S. Ali. *Multiagent telerobotics: Matching systems to tasks*. PhD thesis, Georgia Institute of Technology, 1999.
3. M. Fields. Modeling the human/robot interaction in onesaf. In *Proceedings of the 23rd Army Science Conference*, 2002. (Poster).
4. T. Fong, C. Thorpe, and C. Baur. Multi-robot remote driving with collaborative control. *IEEE Transactions on Industrial Electronics*, 50(4):699–704, August 2003.
5. G. A. Kaminka and Y. Elmaliach. Experiments with an ecological interface for monitoring tightly-coordinated robot teams. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA-06)*, 2006.
6. G. A. Kaminka and R. Glick. Towards robust multi-robot formations. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA-06)*, 2006.
7. R. Murphy, J. Casper, M. J. Micire, and J. Hyams. Mixed-initiative control of multiple hetrogeneous robots for urban search and rescue. Technical Report CRASAR-TR2000-11, Center for Robot-Assisted Search & Rescue, University of Southern Florida, 2000.
8. K. L. Myers and D. N. Morely. Human directability of agents. In *Proceedings of the First International Conference on Knowledge Capture, K-CAP 2001*, Canada, 2001.
9. P. Scerri, L. Johnson, D. Pynadath, P. Rosenbloom, M. Si, N. Schurr, and M. Tambe. A prototype infrastructure for distributed robot, agent, person teams. In *AAMAS-03*, 2003.
10. T. Suzuki, K. Yokota, H. Asama, H. Kaetsu, and I. Endo. Cooperation between the human operator and the multi-agent robotic system: Evaluation of agent monitoring methods for the human interface system. In *Proc. of the 1995 IEEE/RSJ International conference on intelligent robots and systems*, pages 206–211, 1995.
11. A. D. Tews, M. J. Mataric, and G. S. Sukhatme. A scalable approach to human-robot interaction. In *ICRA-03*, 2003.