Carnegie Mellon University
Research Showcase @ CMU

Robotics Institute | School of Computer Science

2008

Sliding Autonomy for Peer-To-Peer Human-Robot Teams

M. Bernardine Dias
Carnegie Mellon University

Balajee Kannan
Carnegie Mellon University

Brett Browning
Carnegie Mellon University

Gil Jones
Carnegie Mellon University

Brenna Argall
Carnegie Mellon University

See next page for additional authors

Follow this and additional works at: http://repository.cmu.edu/robotics

Part of the Robotics Commons

This Conference Proceeding is brought to you for free and open access by the School of Computer Science at Research Showcase @ CMU. It has been accepted for inclusion in Robotics Institute by an authorized administrator of Research Showcase @ CMU. For more information, please contact research-showcase@andrew.cmu.edu.

Authors
M. Bernardine Dias, Balajee Kannan, Brett Browning, Gil Jones, Brenna Argall, M. Freddie Dias, Marc Zinck, Manuela M. Veloso, and Anthony Stentz

This conference proceeding is available at Research Showcase @ CMU: http://repository.cmu.edu/robotics/104

Sliding Autonomy for Peer-To-Peer Human-Robot Teams[1]

M. Bernardine DIAS, Balajee KANNAN, Brett BROWNING, E. Gil JONES,
Brenna ARGALL, M. Freddie DIAS, Marc ZINCK,
Manuela M. VELOSO, and Anthony J. STENTZ
School of Computer Science, Carnegie Mellon University

Abstract. The vision of humans and robots working together as peers to accomplish complex tasks has motivated many recent research endeavors with a variety of applications ranging from lunar construction to soccer. However, much of this research is still at an early stage, and many challenges still remain in realizing this vision. A key requirement for enabling robustness and efficiency in human-robot teams is the ability to dynamically adjust the level of autonomy to optimize the use of resources and capabilities as conditions evolve. While sliding autonomy is well defined and understood in applications where a single human is working with a single robot, it is largely unexplored when applied to teams of humans working with multiple robots. This paper highlights the challenges of enabling sliding autonomy in peer-to-peer human-robot teams and extends the current literature to identify and extend six key capabilities that are essential for overcoming these challenges. These capabilities are *requesting help*, *maintaining coordination*, establishing *situational awareness*, enabling interactions at different levels of *granularity*, *prioritizing* team members, and *learning* from interactions. We demonstrate the importance of several of these characteristics with results from a peer-to-peer human-robot team engaged in a treasure hunt task.

Keywords. Sliding autonomy, human-robot teams, peer-to-peer teams, pickup teams, autonomous teamwork, multi-agent coordination, adjustable autonomy.

Introduction

The vision of humans and robots working together to accomplish complex team tasks is driving much of the current research in the area of autonomous teamwork. As robots become more capable they can handle increasingly complex tasks and highly uncertain environments, but the robotic capabilities in many domains are still insufficient to execute these tasks robustly and efficiently. In these scenarios, robots can still accomplish the tasks with human assistance as human capabilities are often better-suited for some tasks and complement robot capabilities in many situations. Thus, for robots to become an integral part of society, human-robot teams must be effective in a variety of settings.

[1] This work is primarily funded by the Boeing Company Grant CMU-BA-GTA-1. The content of this publication does not necessarily reflect the position or policy of the Boeing Company and no official endorsement should be inferred. This work was also enabled in part by funding from the Qatar Foundation for Education, Science and Community Development and from the U.S. Army Research Laboratory, under contract Robotics Collaborative Technology Alliance (contract number DAAD19-01-2-0012).

In most of the published work on human-robot teams, the human's role is limited to either a supervisor [6] or end-user [9] when interacting with a single or multiple robots or agents [10]. While this hierarchical relationship between human and robot is appropriate in some domains, there are many applications where a peer-to-peer relationship enables more effective use of the complimentary capabilities of humans and robots. In this work, we focus on *peer-to-peer human-robot teams* where humans and robots can assign tasks to each other through direct requests/commands, or through automated task allocation systems. Several research efforts are emerging in this area [7] with applications ranging from lunar construction [4] to soccer [2]. Within the topic of human-robot teams, we are especially interested in *pickup teams* [5] where the composition of the team is not previously known and where members joining the team can vary in their capabilities, expertise, and knowledge of the task. Pickup teams should absorb this wide variety of members to quickly form effective teams, and improve over time as the strengths and weaknesses of different members are discovered and accounted for in the team strategy.

An important aspect of enabling effective human-robot pickup teams is allowing the team to adjust its level of autonomy as necessary. *Sliding autonomy*[2] was introduced to optimize performance by allowing a system to adapt its level of autonomy during execution to accommodate dynamic conditions. The agent and robotics literature are populated with many studies on sliding autonomy applied in different scenarios. However, this work has not been extended to peer-to-peer human-robot teams to date. This paper explores the challenges in applying sliding autonomy in peer-to-peer human-robot team settings and proposes a set of guidelines for accomplishing this task. The proposed guidelines are used to implement a system of humans and robots engaged in a treasure hunt task.

1. Sliding Autonomy in Peer-To-Peer Human-Robot Teams

"Autonomy" is defined in terms of a system's ability to function effectively without human intervention. For example, a fully autonomous system (or "pure autonomy") is said to require no human intervention to complete a task [6]. "Sliding autonomy" is similarly defined in terms of the system's ability to incorporate human intervention when needed (and to otherwise operate independently) [10]. Both of these definitions must change when humans are a part of the "system" or team and where the humans and robots interact as peers. We extend the definition of autonomy presented by Maheswaran et al. [9] where the ability to decide transfer (or sharing) of control governs the level of autonomy. Thus, s*liding autonomy in peer-to-peer teams means that members of the team (humans, robots, and software agents) can actively decide if and when to transfer control to (or share control with) another member of the team or, in some cases, to some entity outside the team*. Because the team members are heterogeneous, some team members may not be capable of making their own decisions. Hence, the decision-making control can shift between different members of the team as

[2] Sliding autonomy and adjustable autonomy are used interchangeably in parts of the literature. We prefer the term "sliding" autonomy because it implies that the level of autonomy can be dynamically adjusted during execution.

needed, but all team members may not possess this capability equally. We also allow prioritization of different team members such that higher priority members can seize control from lower priority members if deemed necessary.

Our work builds on the methodology for sliding autonomy in multi-agent teams proposed by Sellner et al. [6] and work on mixed-initiative teams reported by Bruemmer and Walton [11]. Specifically, we identify six necessary capabilities for enabling sliding autonomy in peer-to-peer human robot teams[3]. The first three of these capabilities are extended forms of characteristics presented by Sellner et al. [6] as major issues that affect human awareness in multi-agent teams. They are the ability to request help, the ability to maintain team coordination during interventions, and the ability to provide situational awareness. The second set of characteristics are augmented versions of capabilities identified by Bruemmer and Walton [11] in the context of robots in mixed-initiative teams. They are the ability to interact at different granularities, the ability to prioritize team members, and the ability to learn from interactions. We next discuss how these six characteristics enable sliding autonomy in a peer-to-peer team setting of humans and robots.

In peer-to-peer teams no single member is necessarily aware of the entire team state. Hence, these teams are more effective when individual agents and sub-teams can identify situations where they need to ***request help*** from other members of the team. These requests for help primarily occur in situations where an agent discovers a failure it cannot rectify without the help of another agent and hence must adjust its level of autonomy [8]. In some situations a team member may not be capable of asking for help or assisting in a recovery process from a failure [8]. In this event, other team members will need to recognize this failure and adapt the team strategy as needed. Monitoring teammates becomes more difficult in pickup teams since team composition can change over time and unfamiliarity with identifiable characteristics that indicate faults in new team members can impede the process of fault recognition and identification. Another type of help request can occur when an agent is assigned a task that requires resources or capabilities that the agent doesn't possess. In this case, the agent needs to recruit others to assist with the task and, in some cases, may need to adjust its level of autonomy to accomplish the task.

Maintaining coordination during interventions is important for effective team performance. For example, if one robot in the team suffers a failure during the execution of a team task, the rest of the team should maintain their coordination while re-strategizing to assist in the recovery from the failure. The team may continue to execute the task despite the failure, or discover that they are unable to complete the task due to the failure and hence request further assistance from an entity outside the team. Therefore, autonomy may be adjusted for some tasks that require external intervention while other tasks are carried out autonomously. Furthermore, if the team is able to overcome the failure without external intervention, it is important that coordination is maintained while the failure is addressed by the relevant team members and that other members of the team continue with their tasks.

Gaining and maintaining situational awareness is perhaps the biggest relevant challenge in a team setting. Situational awareness is a key factor in executing early and successful interventions, and in decisions for adjusting autonomy. In teams with

[3] We assume the teams can also be pickup teams [5] and that the team composition and capabilities can change during execution.

multiple mobile humans, it is not sufficient to capture information in a single graphical interface (GUI), and customization of the state information for the different members of the team may be required. Furthermore, the state of the humans and the dialog and gestures that are a natural part of human-to-human communication must be captured and made transparent to the robots on the team since interventions might be carried out by robots. Situation awareness in pickup teams is also difficult because we must be able to accommodate new capabilities and resources as members join the team, and we must be able to expose the state of the current team to the new member quickly and effectively. Several research efforts are focused on a variety of communication strategies for human-robot teams that include tools such as GUIs, 3-D interactive environments, dialog, and gestures ([1], [3], [4], [5], [6], [11]). However, there is still much to be accomplished in this area of research.

The ***granularity of interaction*** must often be flexible in peer-to-peer teams and this also translates to sliding autonomy. This primarily impacts exposure to information and interactions between team members. The level of granularity of the information presented to any team member will often need to be adaptable for individual components of the system for effective comprehension by different team members. This directly correlates to the previous requirement for effective situational awareness. In terms of interactions among team members, the most effective teams allow for some members to interact in a tightly-coordinated manner to accomplish some tasks, while others act independently. In the case of sliding autonomy, this is also important because autonomy might be adjusted by a single agent's intervention or by a sub-team intervention. For example, two or more agents may coordinate to assist a robot that is stuck in the mud by tightly coordinating in a sub-team, but once the robot is out of the mud, they may coordinate more loosely to accomplish other tasks.

Explicit ***prioritization of team members*** is important in peer-to-peer teams because we cannot assume an inherent hierarchy. The prioritization of the team members is ideally adjustable and specifiable along different dimensions. For example, human members might be prioritized in safety considerations, but a robot with powerful computing capability might be prioritized for planning tasks. These priorities can change when the team composition changes, and also due to other dynamic conditions and should also translate to sliding autonomy. Higher priority members of the team might be eligible for earlier interventions when there is contention for limited resources. Another possibility is that higher priority members can override the autonomy of lower priority members and temporarily seize control for certain tasks.

Finally, ***learning from interactions*** is important for effective team performance. For example, in pickup teams, the prioritization of new members for different tasks and concerns may not be initially known and instead have to be learned based on interactions over time. Also, in terms of sliding autonomy, team members may need to learn to detect indicators of failure so that autonomy can be adjusted when necessary.

In summary, we have discussed six necessary capabilities for sliding autonomy in human-robot peer-to-peer teams: R*equesting help*, *maintaining coordination*, establishing *situational awareness*, enabling interactions at different levels of *granularity*, *prioritizing* team members, and *learning* from interactions. Next, we describe an initial implementation of some of these capabilities for a peer-to-peer human-robot team engaged in a treasure hunt task.

2. Approach and Implementation Details

Our implementation (described in further detail in previous publications [3] and [5]) supports two different *granularities* for tasking human-robot teams. First, a high-level task objective is issued to the system. The system responds by autonomously selecting a plan that is potentially multi-agent and tightly-coordinated to accomplish the objective. A pickup sub-team is created from available team members with the necessary capabilities to efficiently execute this plan. Once the pickup team has been selected, the agents *coordinate* via simple communication protocols during execution, handle errors, and report status to each other. Occasionally errors occur or new information is discovered that cannot be addressed by the robotic agents alone. In these cases, the robots *request help* from a human peer who may or may not have been part of the original sub-team team. The human can join the team and seize control of the system, physically intervene, or issue low-granularity commands to the robot participants. Currently, we use a fixed *prioritization* technique on the robots where low-level commands from humans override robot commands. We do not yet address *learning* (an area of future work). Finally, the flow of information between robots and humans is directed through a GUI application. *Each human* can use a tablet PC that runs the somewhat customizable GUI so that they are *aware* of the current system state. Human actions are explicitly communicated to robots in the current implementation.

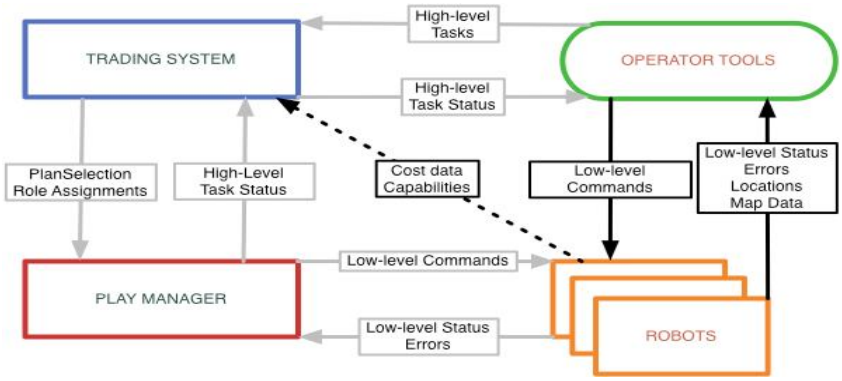

Figure 1: The four components of our system, with arrows indicating the flow of information. The pathway for high-level tasks and resulting high-level status is shown in grey arrows. The pathway for low-level commands directly to robots for error recovery is shown by black arrows. The dotted arrow represents information supplied by the robots to be used in high-level planning and allocation.

Our system consists of four main components, as shown in Figure 1: a human interface tool (OPERATOR TOOLS), a distributed market-based planning and allocation system (TRADING SYSTEM), a component for synchronized tightly-coordinated multi-agent plan execution (PLAY MANAGER), and robot software that supports sensing and acting in the environment (ROBOTS). The operator tools, allow an operator or human peer to issue both high- and low-level tasks and to process status messages in addition to displaying state information of the robots. These are our primary method for supporting visual situational awareness for the humans. Tasks can be autonomously allocated via the Trading System, or can be issued directly to specific agents. The Trading System uses an instantaneous allocation approach, where agents

will only participate in the formation of a new sub-team if they are not actively involved in another high-level task. Multiple sub-teams can simultaneously address different tasks and errors are reported to both humans and robots. We use a tiered auction approach, where individual agents attempt to generate plans and recruit other agents' participation in those plans; the trading system selects the most efficient plan and allocation from the submitted bids. Agents provide information about their capabilities and other cost data that helps the trading system determine plan efficiency. Capability information is used to determine which agents can best fill particular roles in a possible plan. Cost data is used to differentiate between different capable agents when determining which agents can most efficiently fill a particular role. The trading system provides the ability to dynamically form sub-teams that will be maximally efficient in addressing high-level objectives and maintaining coordination even if team composition changes during operation. Currently, humans do not participate in the auctions for tasks but instead are explicitly enrolled in relevant tasks.

Once a plan is selected and roles assigned the information is passed to the Play Manager to coordinate the execution of actions by the sub-team of multiple agents. The Play Manager sends a series of low-level commands to the agents assigned to participate in the coordinated plan. If execution concludes successfully, status is reported to all participating agents. In some cases, due to the highly unstructured and dynamic nature of the environment and the realities of robot hardware, agents may fail and may have no contingency plan. In this case they report errors and directly request help. Help requests primarily take the form of visual cues on the operator GUI (as shown in Figure 2), but we have also experimented with dialog-based error notification [3]. Intervention to recover from failures can take two primary forms – physical interaction and direct low-granularity commands. When resolving failures physically, a human directly interacts with robot hardware. For instance, a robot may experience a problem with its laser range finder that can only be resolved by power cycling the laser unit; if a human performs this action in a timely manner, execution of the original plan resumes. In direct robot command, the human can use the operator tools to issue low-granularity commands to a particular robot. For instance, a robot may become trapped or lost. In these situations a human can issue a series of relative waypoints to free the robot or to move it back to a known area, after which plan execution can continue. Thus, sliding autonomy increases robustness and adaptability.

The final component of our system is the robot/agent software. The pickup team formulation depends on abstracting away many elements of robot software implementation in order to support the seamless integration of new team members. We represent robots in terms of their capabilities, the actions they can perform, and the sensor data and errors they can produce. If agents can represent themselves in this manner our system can easily accommodate pickup teams, with members fluidly participating in sub-team formation and execution of tasks.

3. Experiments, Results, and Discussion in the Treasure Hunt Domain

We demonstrate the effectiveness of our approach in the "Treasure Hunt" domain [5], which is motivated by applications such as de-mining where human exposure to danger must be minimized but humans are needed to deal with safe maneuvering of the discovered items. The task requires a human-robot team to locate and retrieve items of

interest or "treasure" (visual fiducials) in an unknown environment. The key tasks include exploration, mapping, search and localization of treasure, and retrieval of treasure to a "home" location. We use heterogeneous platforms with complimentary capabilities: Pioneer IIDX robots equipped with SICK LiDar and fiber optic gyros, and Segway RMPs and ER1s equipped with cameras. Pioneers build maps and maintain an accurate pose while Segways can locate Pioneers and treasure, and localize based on the observed position and report location of the Pioneers. ER1s are similar in capabilities to the Segways but are teleoperated. Humans cannot directly observe the operational area from the home location, interact with robots via GUIs, and perform retrievals by following Pioneers to the treasure location and back home. The team *requires* coordination to achieve the task since no team member can perform all operations. Figure 2 shows a screen shot from the GUI, which provides *situational awareness* to the humans. Three types of errors are identified and reported. Laser errors relate to a problem with the Pioneer laser, pose errors occur when a robot's localization becomes corrupted, and arc errors occur when a robot cannot independently discover a safe path. The experiments were performed in a large, complex, cluttered, and dynamic indoor environment (see Figure 3).

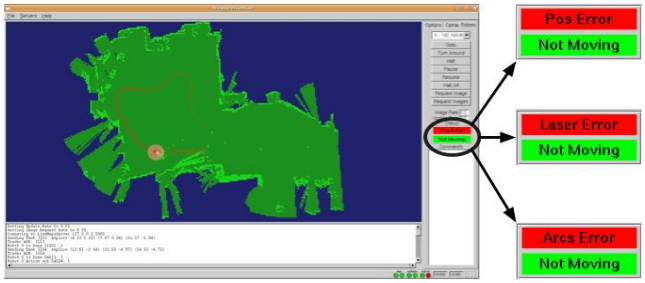

Figure 2: A screenshot from a GUI showing the fused map built from the Pioneer robots. Shown is the recent trajectory of a pioneer (red trail), with a pose/laser error. Other errors show up in a similar manner. A human can issue commands and monitor the state of the team.

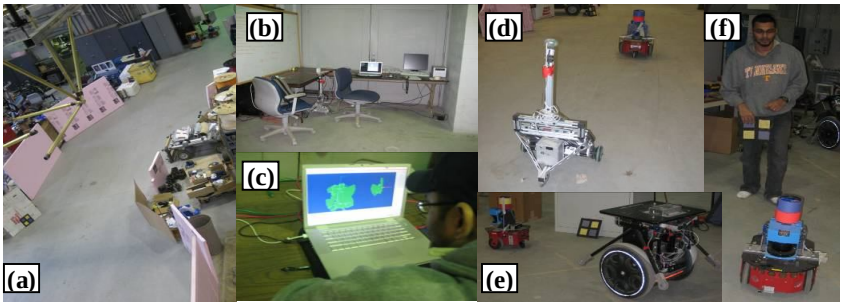

Figure 3: (a) Overhead view of the operating environment where 7 "treasures" are randomly placed, (b) "Home" location, (c) A human team member observing the map being built by the Pioneer robots, (d) An ER1 robot teleoperated to follow a Pioneer robot to search for treasure, (e) A Segway robot autonomously following a Pioneer robot to search for treasure (an item of "treasure" is seen between the two robots), (f) A human being lead back to the "home" location after successfully retrieving an item of treasure.

We measure performance based on the number of successfully identified and retrieved treasure items in a limited time-frame. The first two experiments compare team performance with sliding autonomy enabled versus disabled, while keeping the task parameters constant. In the third and fourth configuration of the system, we replace the autonomous Segway with a teleoperated ER1 robot, and increase the number of humans to evaluate the adaptability of the approach.

Figure 4: Laser map of Highbay area with different treasure configuarations

We perform initial experiments with 3 different treasure configurations (see Figure 4). Each run is conducted over a fixed time period of 15 minutes with a total of 7 "treasure" items scattered throughout the environment. The first set of experiments was performed for a team consisting of 2 humans, 1 pioneer and 1 Segway robot. Table 1 shows the experimental results with sliding autonomy enabled. During these runs, *requests for help* were generated and handled by the system, while maintaining *coordination during intervention*. In contrast, Table 2 shows the experimental results with sliding autonomy disabled. For this experiment, no *requests for help* were generated by the system. The time at which each error occurred is also shown. For all experiments, a combination of errors was either randomly artificially generated (G) or occurred naturally over the course of operation (N). A comparison of the results in Table 1 and Table 2 show that the productivity of the team, measured by the number of treasure items identified and retrieved, decreases in the absence of sliding autonomy.

The third set of experiments demonstrated an alternative human-robot sub-team capable of performing the treasure hunt task. The Pioneer robot in this experiment was autonomous while the ER1 robot was teleoperated by a human. Table 3 shows the experimental results with a sub-team consisting of 3 humans, one Pioneer and one ER1. Table 4 shows the experimental results with error handling turned off for the ER1-Pioneer team. In order to avoid human-biasing as a result of familiarity with the environment and the system, the experiments were performed by two-different humans with no prior experience and one human with prior experience dealing with the robots. During these runs, *requests for help* were generated and handled by the system, while maintaining *coordination during intervention*. In the case of experiments where the sliding autonomy was turned off, no human intervention was provided when the robots fail to autonomously handle errors.

Table 1: Results of 3 runs with sliding autonomy enabled. Type of Errors – Arc (A), Laser (L), and Pose (P). # Error Generated Type – Artificial/manually induced (G) or naturally occurring as part of the system/environment (N) # Robots – R1: leader/explorer Pioneer, R2: retriever Pioneer.

Run	Treasure seen (recovered)	Error Types	Error Source	Error per Robot
T_1	4 (2)	**Total: 5** [L(1), A(2), P(2)]	N(5)	R1(2), R2(3)
T_2	3 (2)	**Total: 6** [L(4), A(1), P(1)]	G(2), N(4)	R1(2), R2(4)
T_3	2 (0)	**Total: 2** [P(1), L(1)]	N(2)	R1(1), R2 (1)

Table 2: Results of 3 runs with sliding autonomy disabled

Run	Treasure seen (recovered)	Error Types	Error Source	Error per Robot
T_1	2 (2)	**Total: 1** [L (6.5 min)]	G(1)	R1(1)
T_2	1 (1)	**Total: 2** [P (2 min), L (5 min)]	G(2)	R1(1), R2(1)
T_3	0 (0)	**Total: 1** [P (7.5 min)]	G(1)	R1(1)

Table 3: Results of 3 runs with a sub-team of humans, a Pioneer and an ER1. Labeling as in Table 1.

Run	Treasure seen (recovered)	Error Types	Error Source	Error per Robot
T_1	4 (4) (Skill level - Novice)	**Total: 2** [L(1), P(1)]	G(2)	R1(1), R2(1)
T_2	6 (3) (Skill level - Expert	**Total: 5** [L(1), A(3), P(1)]	G(2), N(3)	R1(3), R2(2)
T_3	4 (2) (Skill level - Novice)	**Total: 3** [L(1), A(1), P(1)]	G(2), N(1)	R1(2), R2(1)

Table 4: Results of 3 runs with for Pioneer, ER1, human team with sliding autonomy disabled.

Run	Treasure seen (recovered)	Error Types	Error Source	Error per Robot
T_1	3 (2) (Skill level - Novice)	**Total: 1**[L (1)]	G(1)	R1(1)
T_2	0 (0) (Skill level – Novice)	**Total: 1** [A (1)]	N(1)	R1(1)
T_3	4 (2) (Skill level - Novice)	**Total: 1** [L(1)]	G(1)	R1(1)

Overall, these experiments primarily demonstrate that sliding autonomy can improve team performance and that the implemented system can be flexible in accommodating different team configurations for accomplishing the same task.

4. Conclusions and Future Work

The ability to dynamically adjust the level of autonomy during execution can enhance the performance of human-robot teams. This paper extends the framework for sliding autonomy to address peer-to-peer human-robot teams. We highlight six important

characteristics in this context: *requesting help*, *maintaining coordination*, establishing *situational awareness*, enabling interactions at different levels of *granularity*, *prioritizing* team members, and *learning* from interactions. We implement several of these characteristics and demonstrate them in a peer-to-peer human-robot team engaged in a treasure hunt task. While initial experiments show promising results, the current implementation can be improved in several ways. Situational awareness should be enhanced for the robots and humans by capturing human state and communication among human team members. The prioritization of team members should be dynamically adapted to allow for changes in team composition and task priorities. Finally, the system does not currently incorporate any learning. If team members can learn to perform better based on their interactions and other performance metrics, the overall team performance should improve. Ongoing work addresses these extensions.

Acknowledgements

We acknowledge the assistance of Victor Marmol, Sarah Belousov, Ferdinand Justus, and Murali Krishnamoorthy in carrying out the reported experiments. We thank Thomas Harris, Alexander Rudnicky, and Matthew Marge of Carnegie Mellon University, Scott Smith, Ronald Provine, Richard Wojcik, and Michael Kerstetter from the Boeing Company, and the members of the rCommerce and the CORAL groups.

References

[1] F. Driewer, M. Sauer, and K. Schilling, Discussion of Challenges for User Interfaces in Human-Robot Teams, Proceedings of the Third European Conference on Mobile Robots, 2007.

[2] B. Argall, Y. Gu, B. Browning, and M. Veloso, "The First Segway Soccer Experience: Towards Peer-to-Peer Human-Robot Teams," Proceedings of the First Annual Conference on Human-Robot Interactions, 2006.

[3] M. B. Dias, T. K. Harris, B. Browning, E. G. Jones, B. Argall, M. Veloso, A. Stentz, and A. A. Rudnicky, "Dynamically Formed Human-Robot Teams Performing Coordinated Tasks," AAAI Spring Symposium "To Boldly Go Where No Human-Robot Team Has Gone Before," 2006.

[4] T.W. Fong, J. Scholtz, J. Shah, L. Flueckiger, C. Kunz, D. Lees, J. Schreiner, M. Siegel, L. Hiatt, I. Nourbakhsh, R. Simmons, R. Ambrose, R. Burridge, B. Antonishek, M. Bugajska, A. Schultz, and J.G. Trafton, "A Preliminary Study of Peer-to-Peer Human-Robot Interaction," International Conference on Systems, Man, and Cybernetics, IEEE, 2006.

[5] E. Jones, B. Browning, M.B. Dias, B. Argall, M. Veloso, and A. Stentz, "Dynamically Formed Heterogeneous Robot Teams Performing Tightly-Coordinated Tasks," Proceedings of the International Conference on Robotics and Automation, 2006.

[6] B. Sellner, F. W. Heger, L. M. Hiatt, R. Simmons, and S. Singh, "Coordinated Multiagent Teams and Sliding Autonomy for Large-Scale Assembly," Proceedings of the IEEE, Vol. 94, No. 7, 2006.

[7] F. Tang and L. E. Parker, "Peer-to-Peer Human-Robot Teaming through Reconfigurable Schemas," AAAI Spring Symposium on ``To Boldly Go Where No Human-Robot Team Has Gone Before", 2006.

[8] M. B. Dias, R. Zlot, M. Zinck, and A. Stentz, "Robust Multirobot Coordination in Dynamic Environments", Proceedings of the IEEE International Conference on Robotics and Automation, 2004.

[9] R. T. Maheswaran, M. Tambe, P. Varakantham and K. Myers, "Adjustable Autonomy Challenges in Personal Assistant Agents: A Position Paper," in *Agents and Computational Autonomy: Potential, Risks, Solutions*, M. Nickles, M. Rovatsos and G.Weiss (Eds.), Lecture Notes in Artificial Intelligence, vol. 2969, pp. 187-194, Springer-Verlag, 2004.

[10] P. Scerri, K. Sycara, and M. Tambe, "Adjustable Autonomy in the Context of Coordination," AIAA 3rd "Unmanned Unlimited" Technical Conference, Workshop and Exhibit, 2004.

[11] D. J. Bruemmer and M. Walton, "Collaborative tools for mixed teams of humans and robots," Proceedings of the Workshop on Multi-Robot Systems, 2003.