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

M. Bernardine Dias   Balajee Kannan   Brett Browning   E. Gil Jones
Brenna Argall   M. Freddie Dias   Marc Zinck   Manuela M. Veloso
Anthony J. Stentz

CMU-RI-TR-08-16

March 2008

School of Computer Science
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

*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).

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 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 these characteristics with results from a peer-to-peer human-robot team engaged in a treasure hunt task.

Contents

1 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 are able to handle increasingly complex tasks and highly uncertain environments, but the robotic capabilities in many domains are still insufficient to execute tasks robustly and efficiently in a variety of difficult situations. In these scenarios, robots can still accomplish the tasks with human assistance because human capabilities are often better-suited for some tasks and complement robot capabilities in many situations. Thus, if robots are to become an integral part of society, human-robot teams must effectively work together in a variety of settings.

In most of the published work on human-robot teams, the human has a role as either a supervisor [7] or user[8] when interacting with a single robot, or in some cases with multiple robots or agents [9]. While this hierarchical relationship between human and robot is relevant in some domains, there are many applications such as hazardous exploration, surveying, and maintenance where a peer-to-peer relationship enables more effective use of the complimentary capabilities of humans and robots. In this work, we are interested in 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 [11] with applications ranging from lunar construction [6] to soccer [1]. We are further interested in pickup teams [7] where the composition of the team is not previously known and where members joining the team can have a variety of 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. With this understanding of human-robot teams, we can now explore the challenges of sliding autonomy in these peer-to-peer teams.

Sliding autonomy [1] was introduced to optimize performance by allowing a system to adapt its level of autonomy during execution to accommodate dynamic conditions. The agents literature and also the 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.

2 Sliding Autonomy in Peer-To-Peer Human-Robot Teams

Many publications in the agents and robotics literature define the term “autonomy” in relation to the system dependence on human intervention. For example, a fully autonomous system (or “pure autonomy”) is said to require no human intervention to complete a task [7]. Sliding autonomy is similarly defined in terms of the systems ability to incorporate human intervention when needed (and to otherwise operate independently) [9]. 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. [8] where the ability to decide transfer (or sharing) of control governs the level of autonomy. Thus, sliding autonomy in peer-to-peer teams means that members of the team (humans, robots, and software agents) can decide if and when to transfer control to (or share control with) another member of 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 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. Implementing this definition of sliding autonomy in peer-to-peer human-robot teams manifests several challenges.

Our work primarily builds upon the methodology for sliding autonomy in multi-agent teams proposed by Sellner et al. [7] and work on mixed-initiative teams reported by Bruemmer and Walton [2]. Specifically, we identify six necessary

[1] 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.

capabilities for enabling sliding autonomy in peer-to-peer human robot teams [2]. The first three of these capabilities are discussed by Sellner et al. [7] 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 human-interventions, and the ability to provide situational awareness to the human. The second set of characteristics are augmented versions of capabilities identified by Bruemmer and Walton [2] 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 team 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 can be of two distinct types. The more popular type of help request occurs in situations where an error or failure is discovered and the agent who identifies that error cannot rectify the problem without the help of another agent. There may, however, be situations where a team member is not capable of asking for help or assisting in a recovery process from a failure [4]. In these situations, other team members will need to collectively recognize such failures and adapt the team strategy as needed. Monitoring team members becomes even 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. A second type of help request can occur when an agent is assigned a task that requires resources or capabilities that the agent doesnt possess. In this case, the agent needs to recruit others to assist with the task. The primary difference between the two types of help requests is the consequence of the request going unanswered. In the first case, the team will function at reduced capacity if the error or failure is not addressed. In the second case, the task may not get completed.

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 re-strategize to assist in the recovery from the failure and then continue with the task execution, to execute the task despite the failure, or to report the inability to complete the task. Pickup teams further require that coordination strategies adapt dynamically to accommodate new members whose capabilities and resources are not known with certainty.

Gaining and maintaining situational awareness is perhaps the biggest challenge in a team setting. In teams with multiple mobile humans, it may not be sufficient to capture information in a single GUI, and we may also need to allow for customization of the state information for the different members of the team. 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. Finally, in pickup teams 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 ([5], [3], [6], [7], [2]). However, there is still much to be done in this area of research.

The **granularity of interaction** must often be flexible in peer-to-peer teams. This primarily impacts interaction with 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. 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.

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.

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.

[2] We assume the teams can also be pickup teams [7] and that the team composition and capabilities can change during execution.

In summary, we have discussed six necessary capabilities to enable effective sliding autonomy in peer-to-peer human-robot teams. These capabilities are *requesting help, maintaining coordination*, establishing *situational awareness*, enabling interactions at different levels of *granularity*, *prioritizing* team members, and *learning* from interactions. Next, we will describe how we have begun to implement these capabilities for a peer-to-peer human-robot team engaged in a treasure hunt task.

3 Approach and Implementation Details

We have incorporated most of the characteristics of sliding autonomy into our framework for peer-to-peer human robot teams. Our approach supports two different granularities for tasking human-robot teams. In the first method, a high-level task objective is issued to the system. The system responds by determining a multi-agent, potentially tightly-coordinated plan that, if allocated and executed, will address the objective. The system creates a pickup sub-team from all available team members with the necessary capabilities to efficiently execute the plan. Once the pickup team has been created and tasked, the agents coordinate during execution, handle errors, and report back status. Occasionally, errors occur or new information will be discovered that cannot be addressed without human intervention. In these cases, a human peer that may or may not have been part of the original sub-team team can take more control of the system and physically intervene or directly issue low-granularity commands to the robot participants; the low-level command pathway is the second method of tasking human-robot teams. Note, we use a fixed prioritization where low-level commands from humans override system generated commands, and do not yet address learning (an area of future work). We now describe our approach in detail.

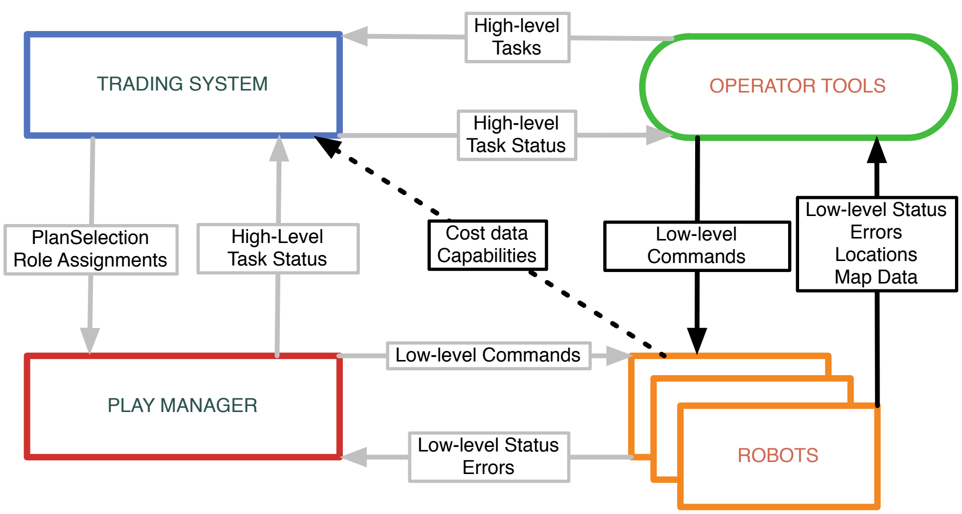

Figure 1: The four components of our system, with arrows indicating the flow of information. The pathway for high-level tasks and resulting high-level status is shown in grey arrows. The pathway for low-level commands directly to robots for error recovery is shown by black arrows. The dotted arrow represents 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: an operator interface tool (OPERATOR TOOLS), a distributed market-based planning and allocation system (TRADING SYSTEM), a component designed for synchronized tightly-coordinated multi-agent plan execution (PLAY MANAGER), and robot software that supports sensing and acting in the environment (ROBOTS).

The first component, the operator tools, allow an operator or human peer to issue both high- and low-level tasks and to process high- and low-level status messages in addition to displaying agent locations and maps. The operator tools are our primary method for supporting visual human situational awareness. High-granularity tasks are passed to the Trading System component, and low-level tasks can be issued to particular robots. Additionally, errors are reported to the operator tools.

The Trading System receives high-level tasks from the operator tools and tries to determine a plan and to form a pickup sub-team of available agents. We use 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 be involved in simultaneously addressing different high-level tasks. The allocation method we used is a tiered auction based approach, where individual robot traders attempt to generate plans and recruit other agents participation in those plans; the trading system selects the most efficient plan and allocation from the individual trader plans. In this system, robots are called on to provide information about their capabilities and also data that helps the trading system determine plan efficiency (Cost data). Capability information is used to determine which agents can best fill particular roles in a possible plan a role may require a certain sensing modality or action type, and if an agent cannot sense in the required way or perform the required actions it should not be assigned to the role. Cost data is used to differentiate between agents with a particular capability, aiding in determination of who among possible agents can most efficiently fill a particular role. (See Jones et. al. [10] for more details). The trading system provides the ability to dynamically form sub-teams that will be maximally efficient in addressing high-level objectives, maintaining coordination even if team composition changes during operation.

Once a plan is selected and roles assigned the information is passed to the Play Manager, which is designed to coordinate the execution of actions 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 back to the operator tools through the trading system pathway. In some cases, however, 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 to the operator, directly requesting help. Help requests primarily take the form of visual cues on the operator GUI (as shown in Figure 3), but we have also experimented with dialog-based error notification [3]. Human intervention for unhandled errors can take two primary forms physical interaction and direct low-granularity robot command. When resolving errors physically, the 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 once a human has performed this action then plan execution can continue. 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 the environment. In these situations the human can potentially 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. With sliding autonomy for error recovery within our system we hope to increase its 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.

4 Experiments in the Treasure Hunt Domain

We demonstrate the effectiveness of our approach in the "Treasure Hunt" domain [7], which is motivated by applications such as some de-mining scenarios 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 specified treasure items (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 orthogonal 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 teleoperated.

Humans cannot directly observe the operational area from the home location, interact with robots via a GUI, 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 operators. Three types of errors are supported. Laser errors relate to a problem with the Pioneer laser, pose errors occur when a robots 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).

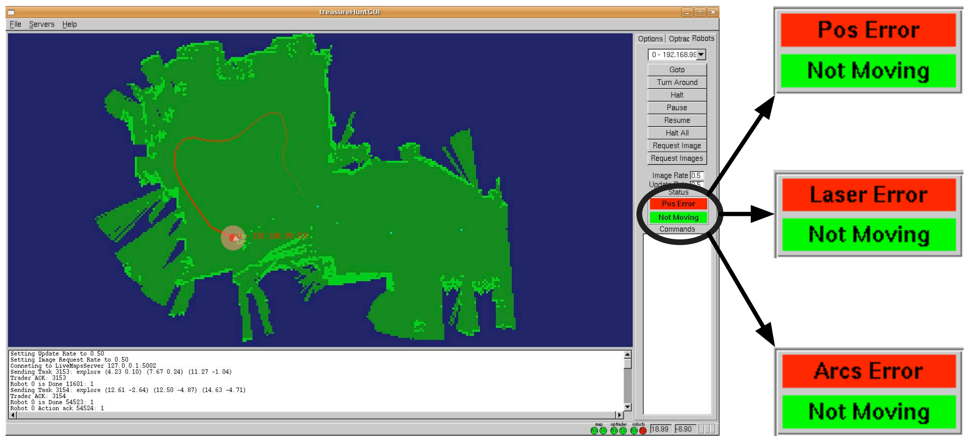

Figure 2: A screenshot from the robot GUI showing the fused map built from the pioneers. Shown is the recent trajectory of a pioneer (red trail), with a pose/laser error (artificially induced). Other errors show up in a similar manner. The operator can give task and low-level commands, and see the state of the teams.

We measure performance of the team based on the successful identification and retrieval of treasure in a limited time frame. In all cases, the environment is unknown a priori. The first two experiments compare team performance with sliding autonomy enabled versus disabled, while keeping the task parameters constant. As a third 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 architecture. In summary, we report three sets of experiments: the full system, the system without error handling, and an alternate human-robot team configuration.

5 Results and Discussion

We perform 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. In order to avoid human-biasing as a result of familiarity with the environment

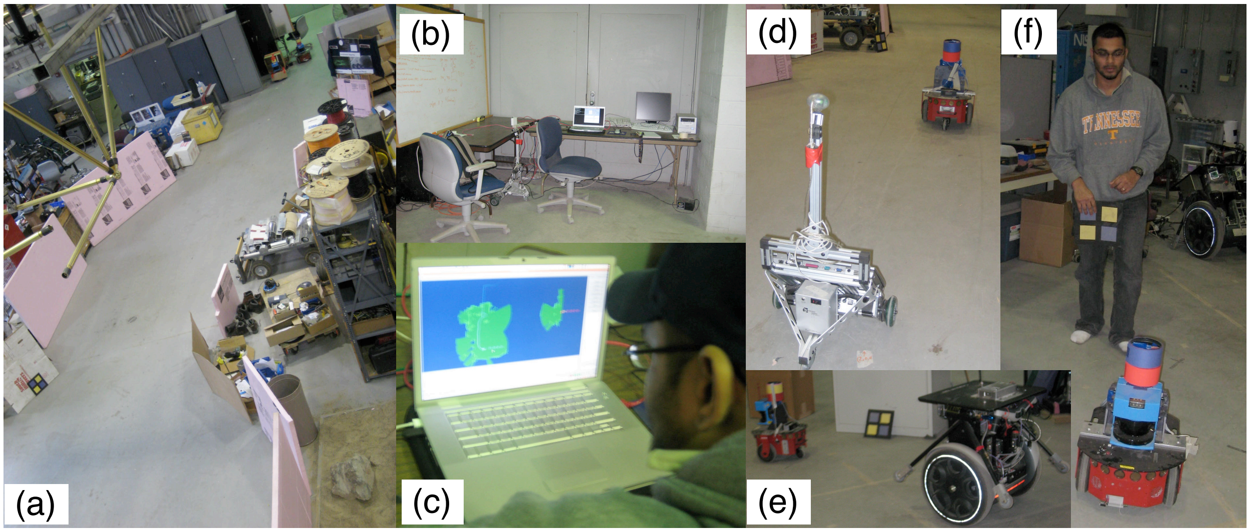

Figure 3: Execution of the Treasure Hunt task. From left to right and top to bottom, the images show (a) overhead view of the operating environment where 7 "treasure items" are randomly placed, (b) the secluded "home" location, (c) a human team member observing the map being built by the Pioneer robots, (d) the ER1 robot being teleoperated to follow the Pioneer robot and search for treasure, (e) the Segway robot autonomously following the Pioneer robot and search for treasure (an item of "treasure" is seen between the two robots), and (f) a human being lead back to the "home" location after successfully retrieving treasure.

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

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*.

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. Team performance can be further enhanced in several ways. For example, several other requests for help can be generated by the robots and the humans and are likely to improve performance. Situational awareness can also be enhanced by capturing human state and communication among human team members. The prioritization of team members can 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.

Run	Treasure seen (recovered)	Error Types	Error Source	Errors 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 1: Results of 3 runs with sliding autonomy was enabled. Type of Errors – Arc (A), Laser (L), and Pose (P). #Error Generated Type – Artificial/manually induced (H) or occurring as part of the system/environment (S) #Robots – R1: leader/explorer pioneer, R2: retriever pioneer.

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 2: Results of 3 runs with sliding autonomy disabled.

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

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

6 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 concept of sliding autonomy to peer-to-peer teams and identifies six important capabilities to optimize performance. 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 implement several of these capabilities and demonstrate their importance in a peer-to-peer human-robot team engaged in a treasure hunt task. Future work includes conducting more extensive experiments, and extending the implementation to incorporate learning, enhanced situational awareness, and dynamic prioritization of team members.

7 Acknowledgments

The authors wish the acknowledge the assistance of Victor Marmol, Sarah Belousov, Ferdinand Justus, and Murali Krishnamoorthy with carrying out the experiments reported in this paper. The authors also wish to 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 Laboratory and the CORAL group.

References

[1] B. Argall, Y. Gu, B. Browning, and M. Veloso. The first segway soccer experience: Towards peer-to-peer human-robot teams. In *Proceedings of the First Annual Conference on Human-Robot Interactions*, 2006.

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

[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. In *AAAI Spring Symposium To Boldly Go Where No Human-Robot Team Has Gone Before*, 2006.

[4] M. Bernardine Dias, Marc Zinck, Robert Zlot, and Anthony Stentz. Robust multirobot coordination in dynamic environments. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2004.

[5] F. Driewer, M. Sauer, and K. Schilling. Discussion of challenges for user interfaces in human-robot teams. In *Proceedings of the Third European Conference on Mobile Robots*, 2007.

[6] 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. In *International Conference on Systems, Man, and Cybernetics, IEEE*, 2006.

[7] E. Gil Jones, Brett Browning, M. Bernardine Dias, Brenna Argall, Manuela Veloso, and Anthony Stentz. Dynamically formed heterogeneous robot teams performing tightly-coupled tasks. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2006.

[8] 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. In M. Nickles, M. Rovatsos, and G.Weiss, editors, *Lecture Notes in Artificial Intelligence*. Springer-Verlag, 2004.

[9] P. Scerri, K. Sycara, and M. Tambe. Adjustable autonomy in the context of coordination. In *AIAA 3rd Unmanned Unlimited Technical Conference, Workshop and Exhibit*, 2004.

[10] B. Sellner, F. W. Heger, L. M. Hiatt, R. Simmons, , and S. Singh. Coordinated multiagent teams and sliding autonomy for large-scale assembly. In *Proceedings of the IEEE - Special Issue on Multi-Robot Systems*, 2006.

[11] F. Tang and L. E. Parker. Peer-to-peer human-robot teaming through reconfigurable schemas. In *AAAI Spring Symposium on To Boldly Go Where No Human-Robot Team Has Gone Before*, 2006.