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 that drives this work is that humans and robots working together as peers in a team setting can accomplish complex tasks that are not possible with current technology. However, maximizing robustness and efficiency in these teams is non-trivial. A key requirement for effective coordination is the ability to dynamically adjust the level of autonomy of the team to accommodate different situations and failures, and also to optimize the use of the team capabilities in different settings. While the notion of sliding autonomy is well studied in cases where a single human is working with a single robot and in the case where humans are working with assistive software agents, the methodology for sliding autonomy in the case of humans working with multiple robots is largely unexplored. This paper builds upon related work in sliding autonomy for teams to explore the challenges and provide a methodology for sliding autonomy in the setting of peer-to-peer human-robot teams. First results are reported for a heterogeneous 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 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. Furthermore, robots and humans must be able to work naturally together if robots are to become an integral part of society.

The notion of sliding autonomy (also referred to as adjustable autonomy in some parts of the literature) was introduced to allow a system to adapt its level of autonomy as needed during execution to produce the most effective performance. Sliding

[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).

autonomy has been well studied in the agents literature and also in the robotics literature for different scenarios. However, sliding autonomy has not been extended to the peer-to-peer human-robot team applications to date. The reported work explores the challenges and proposes a set of guidelines for extending sliding autonomy for peer-to-peer human robot teams. The proposed guidelines are used to implement a system that enables a team of humans and robots to accomplish a treasure hunt task.

1. Human-Robot Teams

First let's look at what we mean by human-robot teams. In most of the previous work in human-robot interaction, the human has a role as either a supervisor [6] or user[9] when interacting with a single robot, or in some cases, with multiple robots or agents [10]. 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.

We are interested in ***Peer-To-Peer human-robot teams*** where humans and robots can assign tasks to each other through direct requests or through an auction-based task allocation system. Several research efforts are emerging in this area [7] with applications ranging from lunar construction [4] to soccer [2]. We are also interested in ***Pickup teams*** [5] 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 will improve over time as the strengths and weaknesses of different members are discovered and accounted for in planning, task allocation, and task execution.

Scholtz[12] presents a theory of human-robot interaction with five roles based on the level of interaction: *supervisor*, *operator*, *mechanic*, *peer* or *teammate*, and *bystander*. We found this categorization to be somewhat cumbersome in our work and propose instead that four roles are sufficient to categorize *both humans and robots* in a peer-to-peer team setting. These roles are ***decision maker***, ***executor/actor***, ***coordinator***, and ***bystander***. The decision maker role maps to Scholtz's supervisor role and allows one or more humans and/or robots to make high-level decisions for the team and to monitor the team performance towards the team goals. The executors/actors are the humans and robots that execute actions to complete tasks. They can work independently or in sub-teams depending on the task requirements, available resources, capabilities, and level of expertise. The executors will also make decisions at a lower level to allow them to execute tasks. The most common role for team members will be the executor role. Hence, in pickup teams, it will especially be important to gauge and adapt to the level of expertise of the new team members when they join the team. The coordinators are the humans or robots (or software agents) that allocate tasks to different members of the team and monitor the execution of sub-team performance during task execution. Finally, the bystanders map directly to Scholtz's bystander role where a human or robot is extraneous to a given task and interacts with other robots or humans purely based on directly observable feedback (avoiding collisions for example).

With this clearly defined understanding of what we mean by human-robot teams, we can now explore the challenges of sliding autonomy in these peer-to-peer teams.

2. Sliding Autonomy

In the agents and robotics literature many define the term "autonomy" in relation to the system dependence on human involvement. For example, a fully autonomous system (or "pure autonomy") is said to require no human intervention to complete a task [6]. Sliding or adjustable autonomy is thus defined similarly in terms of the system ability to incorporate human intervention when needed (and to otherwise operate independently) [10], thus sliding between lower and higher levels of autonomy as necessary to accomplish a task. 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.

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

We adopt the definition of autonomy presented by Maheswaran et al [9] where the ability to decide transfer of control governs the level of autonomy. Thus, we adapt the definition by Maheswaran et al [9] where adjustable autonomy means that each member of the team (humans, robots, and software agents) can decide if and when to transfer control to another member of the team. In our work, we assume that the team members are heterogeneous and some team members may not be capable of making their own decisions. Hence, our notion of sliding autonomy is that the decision-making control can shift between different members of the team as needed, without requiring that all team members possess this capability. We also allow prioritization of different team members such that higher priority members can seize control from lower priority members if deemed necessary. In our current implementation we allow two levels of priority where humans can override non-human team members, and "Play Managers" can override individual team members participating in a "Play" (group task).

Several challenges manifest when implementing sliding autonomy in a peer-to-peer human-robot team. Our work in this area primarily builds upon 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].

Sellner et al [6] discuss three major issues that affect human awareness in multi-agent teams: requesting help, maintaining coordination, and gaining situational awareness. In a peer-to-peer system we have several additional challenges to overcome. The need to request help is still present in peer-to-peer teams since no single team member is necessarily aware of the entire team state. There may, however, be situations where a team member is not capable of asking for help or assisting in its recovery process from a failure [8]. 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. Maintaining coordination also remains a similar challenge in peer-to-peer teams. Some additional challenges are that pickup teams require that coordination not only continue despite interventions, but also that coordination strategies adapt dynamically to accommodate new members with potentially new or different (and sometimes unknown or uncertain) capabilities and resources.

Gaining and maintaining situational awareness is perhaps the biggest challenge in peer-to-peer teams especially when multiple humans are a part of the team. It is no longer sufficient to create a single GUI since not only must we capture state information of the different robots in the team and make this state transparent to the humans in the team, but the opposite also becomes true. That is, we must now capture the state of the humans and the dialog and gestures that are a natural part of communication between the humans, and make all of this sufficiently transparent to the non-human team members. Furthermore, in pickup teams we must be able to accommodate new capabilities and resources as new 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 such teams that include tools such as GUIs, 3-D interactive environments, dialog, and gestures ([1], [3], [4], [6], [11]). However, there is still much to be done in this area of research.

Bruemmer and Walton [11] discuss a few other necessary capabilities for robots in mixed-initiative teams. We adapt three of these issues -- granularity, prioritization, and learning -- for discussion in the context of peer-to-peer human-robot teams. The issue of granularity is encountered especially in the context of situational awareness. State information will most likely need to be presented at several levels of granularity to allow effective comprehension by different team members, and this level of granularity will need to be dynamic and modular (that is, a member can choose to increase or decrease the granularity of information separately for different system components). Additionally, this information granularity will need to be customizable to different user (human or robot) contexts. Granularity can also affect interactions where certain team members might need to tightly coordinate to accomplish a task, whereas others may be able to largely interact loosely.

Explicit prioritization is important in peer-to-peer teams because one cannot assume an inherent prioritization as in hierarchical teams. Thus, if the safety of certain team members needs to be prioritized, or if specific team members are known to be better decision-makers, these prioritizations need to be made explicit. A key point to note here is that these prioritizations will not necessarily be consistent over all tasks and situations. For example, human members might be prioritized in safety considerations, but a robot with powerful computing capability might be prioritized for planning tasks. Furthermore, these priorities can change when the team composition changes, and also due to other dynamic conditions that may arise. Finally, 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 performance and other knowledge gathered over time.

In summary, we have discussed the challenges in sliding autonomy for peer-to-peer human-robot teams in the areas of asking for help, maintaining coordination, establishing effective situational awareness, dealing with different granularities of information and interaction, prioritization of team members, and learning to adapt strategies based on interactions over time. Next, we will describe how we have started to incorporate this methodology into a coordination system that enables peer-to-peer human-robot teams to engage in complex tasks.

3. Approach and Implementation Details

Our approach supports two methods for tasking human-robot teams. In the first method a high-level task objective is issued to the system. The system must take the high-level objective and determine a multi-agent, potentially tightly-coordinated plan that, if allocated and executed, will address the objective. The system must then create a 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 must coordinate during execution, handle errors, and report back status on the high-level objective. Occasionally, errors will 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-level commands to the robot participants; the low-level command pathway is the second method of tasking human-robot teams. In this section we will describe our approach and implementation that supports these two tasking pathways.

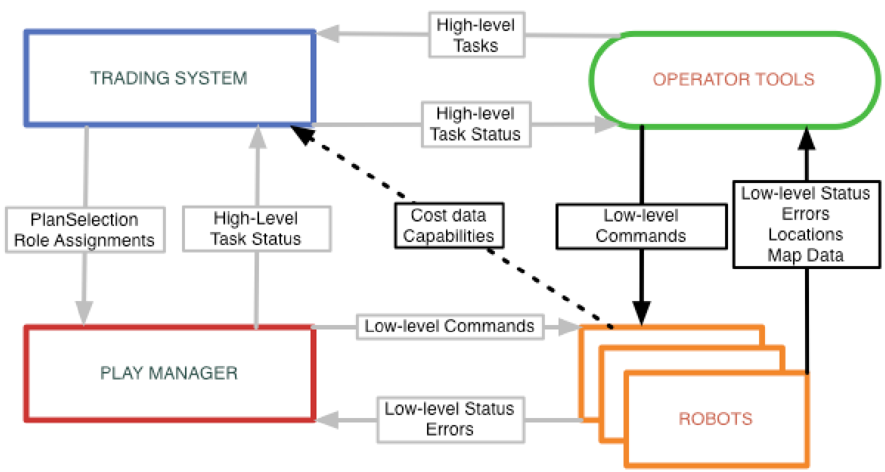


Figure 1: The four components of our system, with arrows indicating the flow of information. The pathway for high-level tasks and resulting high-level status is shown in grey arrows. The pathway for low-level commands directly to robots for error recovery is shown by black arrows. The dotted arrow represents 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 support generation of both high-level task objectives, which are passed to the Trading System component, and to issue low-level commands to particular robots. It can additionally display status for high- and low-level tasks, and locations and maps produced by robots with localization and mapping capabilities.

The Trading System, receives high-level tasks from the operator tools and tries to determine a plan and a 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 different 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 is 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. [6] for more details).

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 plan 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. Human intervention to address unhandled errors can take two primary forms – physical interaction and direct robot command. When resolving errors through physical interaction 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-level 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. By support sliding autonomy for error recovery within our system we hope to increase the robustness and adaptability of the system.

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. To this end 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 form then our system can be easily integrated as a pickup team member, participating in sub-team formation and execution of task objectives.

4. Experiments in the Treasure Hunt Domain

In order to investigate sliding autonomy in human-robot pickup teams we require a domain in which heterogeneous agents must dynamically form tightly-coupled sub-teams in order to maximize a performance metric. We call our chosen domain

“Treasure Hunt”. In Treasure Hunt a human robot team must locate and retrieve a specified treasure items (visual fiducials) in an unknown environment. This requires a combination of exploring and mapping an unknown environment, searching for and localization treasure within the environment, acquiring the treasure, and returning it to a “home” location. We use a heterogeneous combination of platforms in this domain: including Pioneer DXII robots equipped with SICK Laser range finders and fiber optic gyros, Segway RMPs equipped with cameras, and ER1s equipped with cameras. The latter can only be teleoperated in our current set up. Our team has orthogonal capabilities meaning that no one robot can complete the task alone. Pioneers are able to explore, build maps, and stay localized but cannot find treasure or identify teammates (see Figure 3 for an example map). Segways can find treasure and identify teammates, but have only odometry and hence must rely on relative localization from their teammates to localize in the same global frame. ER1s must be teleoperated, and otherwise have similar capabilities to the Segways. Finally, human members are restricted to following robots, retrieving previously localized treasure items and operating with the team via a GUI, which provides both situational awareness and a command interface. The task is run in a large, complex, cluttered, and dynamic environment (The Newell Simon Hall, “Highbay” – see Figure 2. For a more exhaustive description of the Treasure Hunt domain see Jones et al. [6].

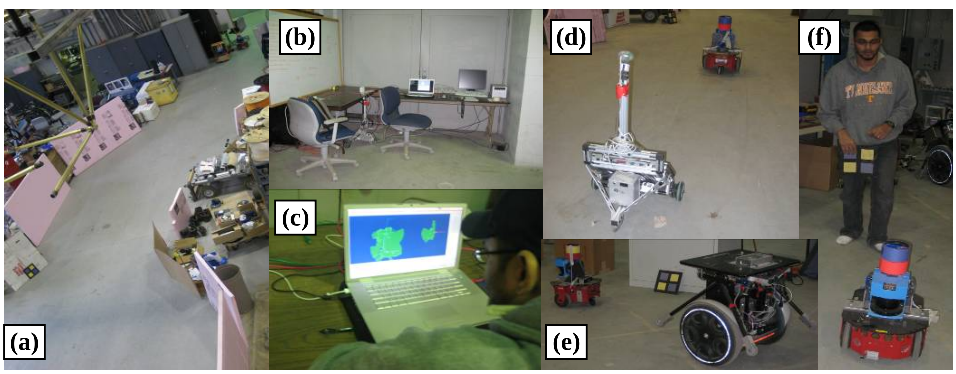
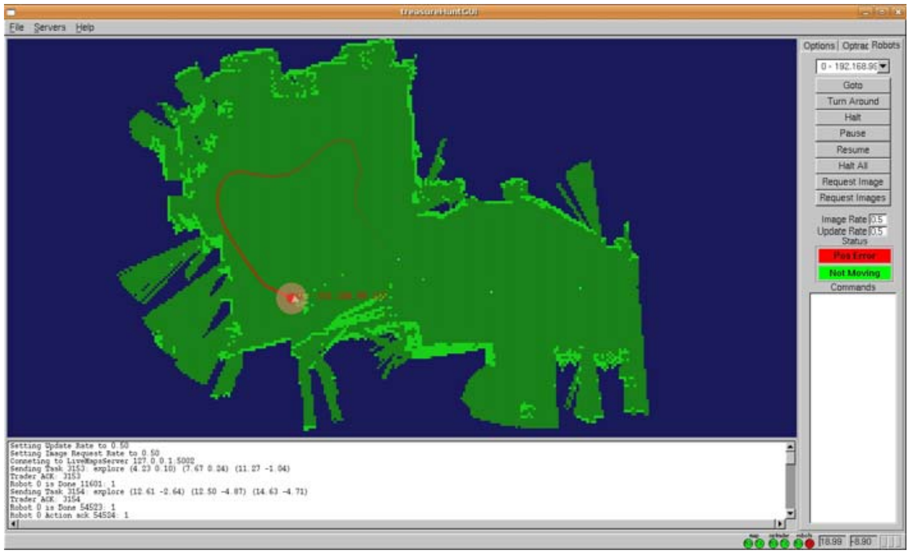


Figure 2: Execution of the Treasure Hunt task. From left to right and top to bottom, the images show (a) overhead view of the operating environment where 7 “treasure items” are randomly placed, (b) the secluded “home” location, (c) a human team member observing the map being built by the laser data sent back 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.

Following [6], we evaluate the system performance in two different settings: with sliding autonomy enabled, with sliding autonomy disabled. In the context of our system, that means we evaluate the performance with the full system with error handling and notification enabled and compare that against a team that is artificially constrained by disabling all error handling and notification. Errors typically are quite infrequent, hence we artificially induce a number of errors to evaluate the impact on the system in a consistent way (true errors may still occur randomly).


File Servers Help
Options Optrac Robots
0 - 192.168.9
Goto
Turn Around
Halt
Pause
Resume
Halt All
Request Image
Request Images
Image Rate 0.5
Update Rate 0.5
Status
Pos Error
Not Moving
Commands
Setting Update Rate to 0.50
Setting Image Request Rate to 0.50
Conneting to LiveMapsServer 127.0.0.1:5002
Sending Task 3153: explore (4.23 0.10) (7.67 0.24) (11.27 -1.04)
Trader ACK: 3153
Robot 0 is Done 11601: 1
Sending Task 3154: explore (12.61 -2.64) (12.50 -4.87) (14.63 -4.71)
Trader ACK: 3154
Robot 0 is Done 54523: 1
Robot 0 Action ack 54524: 1
18.99
-8.90

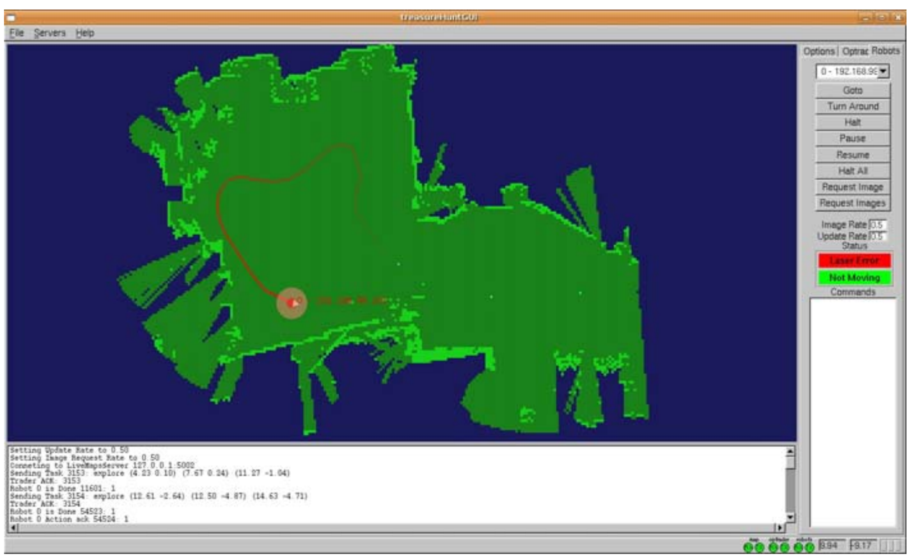

File Servers Help
Options Optrac Robots
0 - 192.168.9
Goto
Turn Around
Halt
Pause
Resume
Halt All
Request Image
Request Images
Image Rate 0.5
Update Rate 0.5
Status
Laser Error
Not Moving
Commands
Setting Update Rate to 0.50
Setting Image Request Rate to 0.50
Conneting to LiveMapsServer 127.0.0.1:5002
Sending Task 3153: explore (4.23 0.10) (7.67 0.24) (11.27 -1.04)
Trader ACK: 3153
Robot 0 is Done 11601: 1
Sending Task 3154: explore (12.61 -2.64) (12.50 -4.87) (14.63 -4.71)
Trader ACK: 3154
Robot 0 is Done 54523: 1
Robot 0 Action ack 54524: 1
8.94
-9.17

Figure 3: Laser map of Highbay area with different treasure configuarations

To measure performance we fix the number of treasure items and evaluate how many the team recovers within a fixed time period. To provide for comparable evaluations, we use the same treasure locations for each setting (error handling, no error handling). Finally, to ensure the experiments cover at last some of the parameter space, we use three different treasure sets, where treasure items are distributed differently. Figure 3 shows a map of the area built by a Pioneer, with the three different treasure sets, which we call environment configurations. The home location is also shown. The human operator can only interact with the team via the GUI and is prevented from being able to directly see the team operate. A second human performs any necessary treasure retrievals as tasked by the system.

5. Results and Discussion

The first set of experiments was performed for a team consisting of 2 humans, 1 pioneer and 1 Segway robot. The following table encapsulates the results from different runs. Each run is conducted over a fixed time period of 15 minutes with a total of 7 treasures scattered throughout the building. For all the runs the pioneers act as explorers, search leaders and treasure retrievers, where Segway is used as followers and humans for pickup. The artificial human-generated (H) errors were randomly generated.

Run	Treasure seen (recovered)	Errors	Generated	Error per Robot
T_1	4 (2)	**Total: 5** [L(1), A(2), P(2)]	S(5)	R1(2), R2(3)
T_2	3 (2)	**Total: 6** [L(4), A(1), P(1)]	H(2) S(4)	R1(2), R2(4)
T_3	2 (0)	**Total: 2** [P(1), L(1)]	H(2)	R1(1), R2 (1)

Table 1: This table shows the experimental results for 3 runs where 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)	Errors	Generated	Error per Robot
T_1	2 (2)	**Total: 1** [L (6.5 min)]	H(1)	R1(1)
T_2	1 (1)	**Total: 2** [P (2 min), L (5 min)]	H(1)	R1(1), R2(1)
T_3	0 (0)	**Total: 1** [P (7.5 min)]	H(1)	R1(1)

Table 2: This table shows the experimental results for 3 runs where sliding autonomy was disabled. Type of Errors – Arc (A), Laser (L), and Pose (P). Additionally, the time of the first error is indicated to give an idea of the amount time and resources wasted when it can not recover from that error. # 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)	Errors	Generated	Error per Robot
T_1	TBA	**Total: 1** [L (6.5 min)]	H(1)	R1(1)
T_2	TBA	**Total: 2** [P (2 min), L (5 min)]	H(1)	R1(1), R2(1)
T_3	TBA	**Total: 1** [P (7.5 min)]	H(1)	R1(1)

Table 3: This table shows the experimental results for 3 runs where a different sub-team of humans, a pioneer, and an ER1, was used for the treasure hunt. Different humans with different skill levels were the primary operators of the ER1 robot. Type of Errors – Arc (A), Laser (L), and Pose (P). Additionally, the duration of each run is given because two runs ended early due to low-battery levels in the Pioneer robot. # Error Generated Type – Artificial/manually induced (H) or occurring as part of the system/environment (S) # Robots – R1: leader/explorer pioneer, R2: retriever pioneer.

6. Conclusions and Future Work

This should be easy to write
So much to do!

Acknowledgements

The authors wish the acknowledge Victor Marmol's valuable assistance in conducting the experiments reported in this paper. The authors also wish to thank Thomas Harris, Alexander Rudnicky, and Sarah Belousov of Carnegie Mellon University, Scott Smith, Ron Provine, Rick Wojack, and Mike Kerstetter from the Boeing Company, and the members of the rCommerce Laboratory and the CORAL group, for their support and contributions which made this work possible.

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.
[12] J. Scholtz, "Theory and Evaluation of Human Robot Interactions," Proceedings of the 36th Annual Hawaii International Conference on System Sciences, 2003.