

2006

Dynamically Formed Heterogeneous Robot Teams Performing Tightly Coordinated Tasks

E. Gil Jones
Carnegie Mellon University

Brett Browning
Carnegie Mellon University, brettb@cs.cmu.edu

M. Bernardine Dias
Carnegie Mellon University, bdias@andrew.cmu.edu

Brenna Argall
Carnegie Mellon University

Manuela M. Veloso
Carnegie Mellon University, veloso@cs.cmu.edu

See next page for additional authors

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

 Part of the [Robotics Commons](#)

Published In

.

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

E. Gil Jones, Brett Browning, M. Bernardine Dias, Brenna Argall, Manuela M. Veloso, and Anthony Stentz

Dynamically Formed Heterogeneous Robot Teams Performing Tightly-Coordinated Tasks

E. Gil Jones, Brett Browning, M. Bernardine Dias,
Brenna Argall, Manuela Veloso & Anthony Stentz

School of Computer Science
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Email: {egjones,brettb,mbdias,bargall,mmv,axs}@cs.cmu.edu

Abstract—As we progress towards a world where robots play an integral role in society, a critical problem that remains to be solved is the Pickup Team Challenge; that is, dynamically formed heterogeneous robot teams executing coordinated tasks where little information is known a priori about the tasks, the robots, and the environments in which they will operate. Successful solutions to forming pickup teams will enable researchers to experiment with larger numbers of robots and enable industry to efficiently and cost-effectively integrate new robot technology with existing legacy teams. In this paper, we define the challenge of pickup teams and propose the treasure hunt domain for evaluating the performance of pickup teams. Additionally, we describe a basic implementation of a pickup team that can search and discover treasure in a previously unknown environment. We build on prior approaches in market-based task allocation and plays for synchronized task execution, to allocate roles amongst robots in the pickup team, and to execute synchronized team actions to accomplish the treasure hunt task.

I. INTRODUCTION

The vision that drives this work is that teams of heterogeneous robots will dynamically solve complex tasks by efficiently joining their complementary capabilities. The research challenges in realizing this vision include robust operation across multiple environments, building capabilities applicable across multiple robot types, and building teams of robots that improve over time.

Many robotics research efforts have investigated multi-robot coordination for a variety of domains and tasks. Competitions, such as RoboCup, have been effective in focusing efforts to overcome some of these challenges [1]. However, these competitions focus on part of the overall problem and do not generally address teams formed in an ad hoc manner, complex environments beyond a well-defined soccer field, and the complexities of heterogeneous teams. How to effectively coordinate heterogeneous teams has been an ongoing challenge in multi-robot research (for example, [2]–[4]). However, the authors are unaware of any work focused explicitly on the principles underlying the building of such highly dynamic teams when the a priori interaction between individual robot developers is minimal. Much of the existing research implicitly assumes that the robot team is built by a group of people working closely together over an extended period of time. While some previous research within the software agents

community has addressed the coordination of simulated agents built by different groups [5], none has chosen to address this pickup challenge for the coordination of multiple robots. We believe this research direction of forming dynamic teams will greatly advance the science of multi-robot systems. Thus, the Pickup Team Challenge is to dynamically form teams of robots (and eventually humans) given very little a priori information. That is, team members may have only minimal prior knowledge of each other’s behavior, the tasks at hand, and the environments in which they operate, but are able to coordinate effectively.

There are several reasons why an increased understanding of pickup teams is needed. First, it is impractical to develop large teams or teams of expensive robots at the same site, at the same time. This currently hinders multi-robot research. Successful pickup teams will facilitate further research by allowing separate researchers to easily pool their robots to create teams for further study. Second, robots may be needed for emergency tasks where there may be insufficient time to hand-engineer the coordination mechanisms before task execution. Pickup teams enable robot teams to be formed on very short notice for such tasks. Third, as robots fail, get lost, or otherwise malfunction, it is often necessary to substitute or add new robots. Successful pickup teams will allow the integration of new robots into existing teams, and also enable teams of heterogeneous robots to perform efficiently under dynamic and uncertain conditions. Thus, the overall research challenge is to provide a principled methodology for creating pickup teams. This paper presents a first approach to address this challenge.

The reported work focuses on dynamically forming teams of heterogeneous robots to perform tasks that require tight coordination. The robots have limited individual capabilities, different sensing modalities, and can be assigned abstract tasks for execution. Robots can solve primitive tasks in different ways depending on their capabilities and prevailing environmental conditions. The implemented approach is demonstrated in a treasure hunt scenario. Thus, the contribution of this paper is threefold: defining the Pickup Team Challenge, introducing the treasure hunt domain, and implementing a first approach to enable coordination of pickup teams.

II. THE TREASURE HUNT DOMAIN

To investigate the conduct of pickup teams, we require a domain which will allow for dynamic and heterogeneous team formation, encourage coordination and tight coupling between team members, and provide a metric against which to compare team performances. The Treasure Hunt is a domain which provides for each of these characteristics.

The Treasure Hunt domain consists of robot teams competing in and exploring an unknown space. The teams are heterogeneous, where the particular capabilities necessary to accomplish hunt tasks are distributed throughout the robot team; a given capability will often be unique to a single member. The hunt tasks are executed towards the goal of locating the specified treasure within an unknown environment. Thus a representation of the world must be built as it is explored, and the treasure must be identified and then localized within the built representation. Team coordination follows as a direct necessity, as the abilities required to perform each of these tasks are distributed throughout the team members.

The ultimate goal with respect pickup team formation is speed and plasticity. Within this domain, not only are teams created quickly and on the fly, but each member has no prior knowledge about the abilities of its potential teammates; a robot knows only of its own capabilities. Inherent to the definition of a hunt task are the abilities necessary to accomplish it. Communication between potential pickup team members is therefore carried out at the time of team formation to ensure the satisfaction of all required capabilities.

This domain presents an adversarial environment in which to execute the hunt tasks. Currently a team competes against the clock with the intent of collecting as much treasure as possible within the allotted time. Eventually teams will compete against other dynamically formed heterogeneous pickup teams. This competitive nature of the domain provides a metric for team performance; the quantity and kind of treasure located. Future scenarios can include the requirement to amass the discovered treasure in a designated location. Additionally, an adversarial environment encourages enhanced team performance in terms of efficiency.

In our specific treasure hunt implementation, potential team members include Pioneer robots and the robotic Segway RMP platform provided by Segway, LLC 1. The Pioneer robots are each equipped with a SICK laser and gyroscope, and are therefore able to both construct a map of an unknown environment, and localize upon that map. The Segway robots have been outfitted with two cameras, which enable them to visually identify both the Pioneer robots and the treasure. The tasks are to search for treasure. Neither type of robot can accomplish the task by itself, but together they can search for treasure: the Pioneer can navigate and construct a map while the Segway follows and visually searches for treasure. When treasure is discovered the Pioneer is able to localize the and record the location of the treasure on the map. Team coordination between the robots during execution is accomplished via the visual identification of the Pioneer by

the Segway, and by communication between the two robots should this visual link be lost (in which case the Pioneer is commanded to pause, until the Segway can find it). The Segway additionally communicates treasure discovery to the Pioneer.

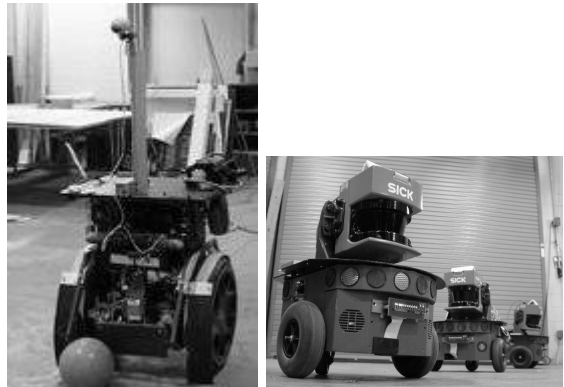


Fig. 1. The left figure shows a Segway robot, while the right figure shows the pioneer robots.

The treasure hunt domain satisfies the criterion set for the study of the performance of pickup teams. It offers a number of challenging aspects, including robust and efficient operation in unconstrained environments, tasks that require tight-coordination of heterogeneous teams, and ad hoc team formation. Efficient execution requires a coordinated search of the space and the maintenance of an accurate shared knowledge about the space. As such, this domain provides a rich environment in which to push the boundaries of adaptive, autonomous robotics.

III. COMPONENT TECHNOLOGIES

In this section we review the components of our current approach to teamwork – Skills, Tactics, and Plays (STP) for team coordination in adversarial environments, and TraderBots for efficient and robust role assignment in multi-robot tasks.

A. STP: Skills, Tactics, and Plays

Veloso et al. [6] introduce a STP for controlling autonomous robot teams in adversarial environments. In STP, teamwork, individual behavior, and low-level control are decomposed into three separate modules. Relevant to our implementation are plays which provide the mechanism for adaptive team coordination. Plays are the central mechanism for coordinating team actions. Each play consists of the following components: (a) a set of roles for each team member executing the play, (b) a sequence of actions for each role to perform, (c) an applicability evaluation function, (d) a termination evaluation function, (e) a weight to determine the likelihood of selecting the play.

Each play is a fixed team plan that describes a sequence of actions for each role in the team towards achieving the team goal(s). Each of the roles is assigned to a unique team member during execution. The role assignment is based on the believed state of the world and is dynamic (e.g. role A may start with

player 1, but may switch to player 3 as execution progresses). Note that the role assignment mechanism is independent of the play framework.

The concept of plays was created for domains where tight synchronization of actions between team members is required. Therefore, the sequence of tactics to be performed by each role is executed in lock step with each other role in the play. Hence, the play forms a fixed team plan whereby the sequence of activities is synchronized between team members.

As not all plans are appropriate under all circumstances, each play has a boolean evaluation function that determines the applicability of the play. This function is defined on the team's belief state, and determines if the play can be executed or not. Thus, it is possible to define special purpose plays that are applicable only under specific conditions as well as general-purpose plays that can be executed under much broader conditions. Once executed, there are two conditions under which the play can terminate. The first is that the team finishes executing the team plan. Each play includes an evaluation function that determines whether the play should be terminated. As with applicability, this evaluation function operates over the team's belief state. Hence, the second means of ending a play is if the termination evaluation function determines that the play should end, either because it has failed or is successful.

Team strategy consists of a set of plays, called a playbook, of which the team can execute only one play at any instant of time. A play can only be selected for execution if it is applicable. From the set of applicable plays, one is selected at random with a likelihood that is tied to the play's weight. The plays are selected with a likelihood determined by a Gibbs distribution from the weights over the set of applicable plays. This means the team strategy is in effect stochastic. This is desirable in adversarial domains to prevent the team strategy being predictable, and therefore exploitable by the opponent.

B. *TraderBots*

TraderBots, developed by Dias and Stentz [7] is a coordination mechanism, inspired by the contract net protocol by Smith [8], is designed to inherit the efficacy and flexibility of a market economy, and to exploit these benefits to enable robust and efficient multirobot coordination in dynamic environments. A brief overview of the *TraderBots* approach is presented here to provide context for the reported experimental results and analysis.

Consider a team of robots assembled to perform a particular set of tasks. Consider further, that each robot in the team is modelled as a self-interested agent, and the team of robots as an economy. The goal of the team is to complete the tasks successfully while minimizing overall costs. Each robot aims to maximize its individual profit; however, since all revenue is derived from satisfying team objectives, the robots self-interest equates to doing global good. Moreover, all robots can increase their profit by eliminating excess cost. Thus, to solve the task-allocation problem, the robots run task auctions, and bid on tasks in other robots task auctions.

If the global cost is determined by the summation of individual robot costs, each deal made by a robot will result in global cost reduction. Note that robots will only make profitable deals. Furthermore, the individual aim to maximize profit (rather than to minimize cost) allows added flexibility in the approach to prioritize tasks that are of high cost and high priority over tasks that incur low cost but provide lower value to the overall mission. The competitive element of the robots bidding for different tasks enables the system to decipher the competing local information of each robot, while the currency exchange provides grounding for the competing local costs in terms of the global value of the tasks being performed.

IV. IMPLEMENTATION

We address two main challenges in our implementation. The first is the efficient and dynamic formation of pickup teams given heterogeneous robots. We have adapted the *TraderBots* system to perform this function. Tasks are matched with plays consisting of a number of roles which, when performed, should satisfy the requirements of the task. Each role in a play contains a sequence of robot-executable action primitives and can require a certain set of capabilities. Robots only bid on roles that they have the capability to perform, thus accommodating the heterogeneity of the robots and providing an efficient way for new kinds of robots with different sets of capabilities to represent themselves to the system. *TraderBots* also requires that robots have the ability to estimate the cost of actions. For instance, a cost may be the total distance that a role requires a robot to move. By minimizing cost in performing role allocations we hope to not only get solutions that are feasible but also have high efficiency.

The second challenge we address is that of robustly executing tightly-coordinated tasks. The heterogeneity of the pickup teams demands that much care be taken during play execution - roles may depend on each other, and all robots need to do their part to actually discover, localize, and retrieve treasure. Thus once the allocation has been performed, the tight coordination subsystem must monitor and direct play execution. We use plays to accomplish this function.

The following describes our implementation of the subsystems for dynamic pickup team allocation and tight coordination. The first part introduces the main components of implementation, and the second part of the section illustrates system performance by describing the life cycle of a treasure hunt task as it moves through allocation to execution.

In our current implementation a number of different processes must be run on each of the different robots, as well as on a human operator workstation. However we anticipate reducing the necessary shared code for pickup teams in the future. Thus, in this paper we describe only those processes essential to the function of pickup teams.

The top layer of our implementation consists of Traders. Each agent, including the human operator, is assigned a Trader. A Trader is the agent's interface to the market. The Trader can introduce items to be auctioned to other Traders by sending a call for bids, can respond to these calls with bids, can

determine which bids are most beneficial for the auctioning agent, and can issue awards to bidders that have won auctions. In our system the human operator has a Trader known as an OpTrader. Each robot agent has a Trader called a RoboTrader.

The PlayManager forms the next component module. The primary role of the PlayManager is to select useful plays for a task and to coordinate the execution of activities in a play across a small sub-team to perform a won task. The PlayManager can select play to address a specified task, coordinate the execution of a play with the PlayManagers for the other assigned roles, execute the sequence of tasks for its particular robot and synchronize the activities, where required, between the different roles.

A final important component - the Robot Server - provides an interface between the PlayManager and the components responsible for controlling the robot. A strength of our system is that neither the PlayManagers nor the Traders need to know very much about how the control of the system is actually implemented - the Robot Server serves as the single point of contact. Thus the Robot Servers on the robots must understand a standard packet, cause PlayManager-commanded actions to be performed, and report action status; beyond that robot platform developers are free to develop their system in any fashion they wish.

A. The Treasure Hunt Task Life Cycle

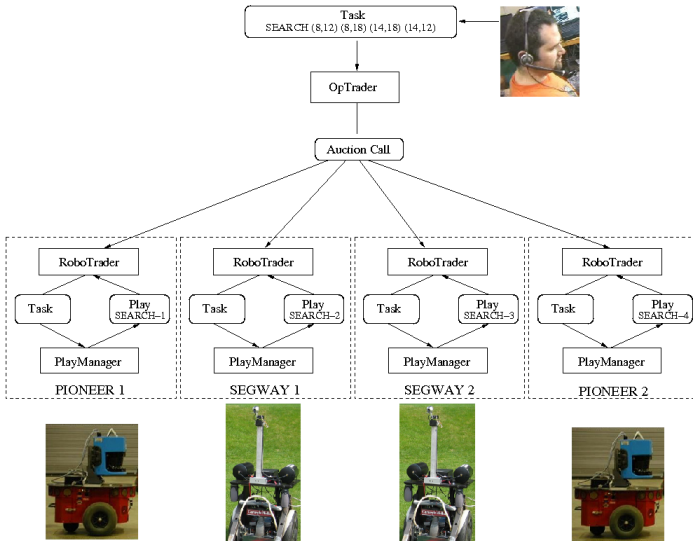


Fig. 2. Parts (a) and (b) of the task life-cycle as discussed in the Implementation section.

- a) **A task is issued and enters the Trading system:** Initially, the human operator designates a SearchArea task, embedding the points of a bounding polygon in the task data. The task is sent to the OpTrader. The OpTrader creates an auction call with the task data and sends it via the wireless network using UDP to all RoboTraders (see Figure 2).
- b) **Each RoboTrader receives the task auction call and gets a matching play from the PlayManager:** The RoboTraders receive the call for bids from the OpTrader and pass

the task specification to their individual PlayManagers. Each contacted PlayManager will compare the task string against the applicability conditions for each play in its playbook. It will then select a play stochastically amongst the set of plays that are applicable, and return this play to the RoboTrader.

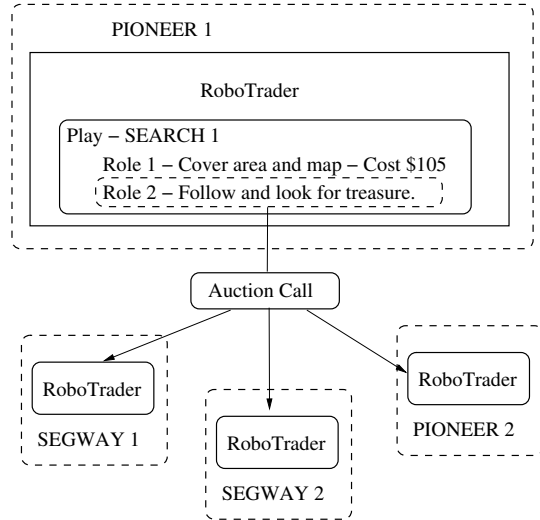


Fig. 3. The RoboTrader, upon receiving a play from the PlayManager, selects a play for itself and produces a cost estimate. It auctions the other role in the play to the other robots.

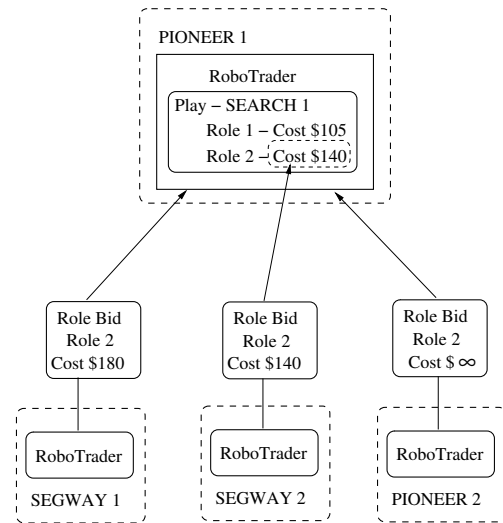


Fig. 4. Receiving the bids for the remaining role, the OpTrader selects the lowest cost bid.

- c) **The RoboTrader assesses the play:** The RoboTrader now has a play matching the task and a set of roles. It then must select one of the roles for itself. For each role in the play, the RoboTrader first considers whether or not it possesses all the capabilities required to perform that role. For all roles for which it is capable the Trader then performs a cost evaluation. In the treasure hunt we are largely concerned with minimizing the time it takes to accomplish the task - as our robots move roughly the same speeds during play execution, we try

to minimize distance travelled as an approximation of time. If robots were heterogeneous with respect to speed this should be reflected in their costing function. For most roles cost is computed as total metric path cost for goal points to be visited in the role. Costing in our system is modular, so additional or different costing functions can be added easily as required.

Once a cost has been assigned to each role, the RoboTrader selects the lowest cost role for itself. It then sends an auction call with all the remaining roles to the other Traders, as shown on the left in Figure 3.

d) The other RoboTraders bid on the role auction:

Upon receiving the call the other RoboTraders determine their own cost for each role in the call. Any role that requires capabilities the Trader’s robot does not possess are assigned an infinite cost. All roles that can be performed by the Trader’s robot are assigned a cost determined by the costing function. The role bids are then returned to auctioneering RoboTrader (shown in Figure 4).

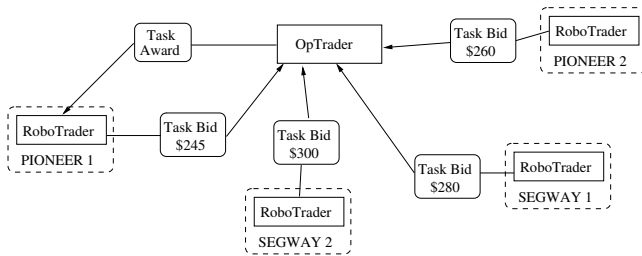


Fig. 5. The bids from all plays are returned to the OpTrader, who selects the lowest cost task bid and sends a task award to the winning bidder.

e) The RoboTrader bids on the task: Once all bids are received or a timeout has expired the auctioneer RoboTrader then determines the winner or winners of the role auction. All role bids are considered, and the lowest cost bid is selected. If that bid is non-infinite, the role is designated as assigned to the bidding Trader. As a Trader can only win a single role in a play, that Trader’s other bids are nullified. Additionally, all bids for that role from any other Trader are nullified. Then the lowest cost remaining bid is considered; this continues until all roles in the play are assigned or there are no remaining bids.

If all roles from the play have been assigned, the RoboTrader constructs a bid for the original task, with a cost that is summed over all assigned role cost estimates. The bid is sent to the OpTrader. This is shown in Figure 5.

f) Task and role awards are granted: Once the OpTrader has received bids from all RoboTraders or a call timeout has expired it awards the task to RoboTrader with the lowest cost bid. An award message is sent to the winning RoboTrader. All other RoboTraders are informed they lost the auction. The winning RoboTrader then sends role award messages to all RoboTraders that were assigned roles in the winning play. Note that at this point the winning RoboTrader still has not accepted the task award.

g) Awards are accepted and play execution begins: The initial role bids made by the RoboTraders were non-

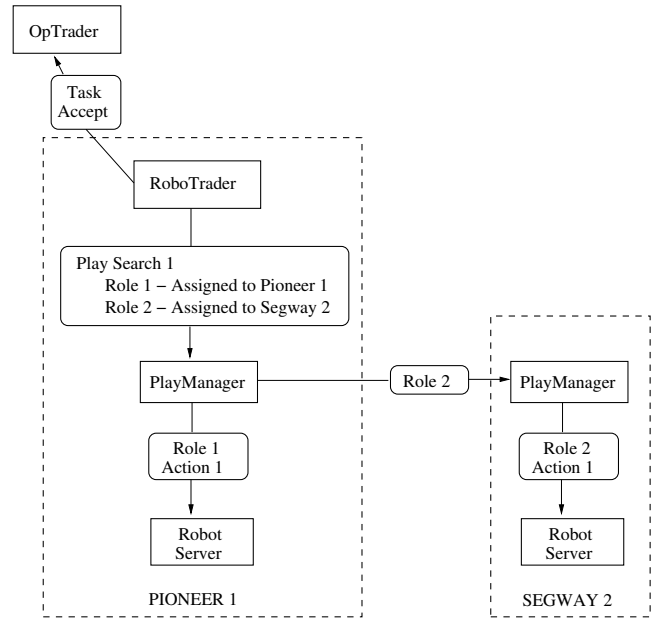


Fig. 6. After the RoboTrader has confirmed availability for all winning role bidders, it informs the OpTrader of task acceptance and signals the PlayManager to begin the tightly coordinated play. The PlayManager contacts other PlayManagers that have been given role assignments. Finally, action primitives are sent to Robot Servers to begin play execution.

binding. However, a role award is binding; once accepted, a role must be performed. If a RoboTrader has received no other role award in the interval between bidding and the arrival of this award, it accepts the role award. Otherwise it rejects the role award. If any of the role awards are rejected, or one of the Traders awarded a role does not respond to the award by a timeout, then the task award is rejected and the OpTrader must perform another auction.

If all role awards are accepted, then the RoboTrader sends a task acceptance message to the RoboTrader. Then it sends an execution message to the PlayManager detailing the final role assignments. This stage is shown in Figure 6. The RoboTrader’s role in the pickup team allocation is concluded except for relaying status information to the OpTrader when the play completes. Future work, however, will examine dynamic re-assignment of roles if and when required.

h) The PlayManager begins play execution: Once informed of the play to execute, and the list of assigned roles, the PlayManager forms the subteam to execute the play. It does this by contacting each of the subteam members RoleExecutors and transmits the play in compressed XML format to them. The RoleExecutor is responsible for executing the assigned role in the play. If any subteam members are essential and fail to acknowledge the play reception, or are not able to be contacted, the play is terminated and reported as such to the RoboTrader. Otherwise, execution proceeds by the PlayManager informing each RoleExecutor to start operation.

At this point, each RoleExecutor becomes loosely coupled to the PlayManager. The RoleExecutor will execute its sequence of actions and will only inform the PlayManager of ter-

mination (success or failure), or when it needs to synchronize with another role according to the play. To synchronize the RoleExecutor contacts the appropriate teammate's RoleExecutor and informs the PlayManager for status keeping purposes.

When each RoleExecutor reaches the end of its sequence of actions to perform, it informs the PlayManager of successful termination, and when the team is complete the PlayManager reports this to the RoboTrader. Alternatively, if a robot fails, or the time limit for execution is reached (as encoded in the play itself), the play is terminated and reported as such. Taken together, execution is distributed, and loosely coupled.

B. Current System

The indicated architecture has been implemented and tested on the Pioneer and Segway platforms. The Pioneers and Segways each have platform-specific Robot Servers. The RoboTraders and PlayManagers run on the platforms identically. The capabilities of a given robot are defined in a configuration file unique to them; the RoboTrader reads this file and uses the information in the role allocation process.

We have implemented and tested one of the main plays required for the treasure hunt domain: the covering of an area while searching for treasure. This play consists of two roles, one which explores and maps an indicated area and the other which follows the exploring agent while visually searching for treasure. We have tested this system from task issue to treasure identification.

While initial testing with multiple robots of each type have been carried out in simulation, we focus here on our first tests on physical robots which include one Pioneer and one Segway robot. In our tests, the search area was initially identified by the human via a graphical user interface. The OpTrader then dynamically allocated the two roles to our participating agents, the Pioneer and Segway. With the allocation completed, the PlayManager successfully enabled the tight coordination required of this team task. Each agent then executed their role until treasure was discovered, thus completing the task and its associated play. The appropriate robots then reported the completion of the task to the human.

To demonstrate the capabilities of the market-based allocation to dynamically form sub-teams, we created a tightly coordinated search play calling for two agents to map and explore an area together. The roles associated with this play designated leader and follower agents. We positioned four Pioneer robots in varying location for this task. Using a distance-based cost model, the two agents with starting positions closest to the search area were dynamically selected for the sub-team and subsequently executed the search pattern.

The realization of these plays utilized a few simplifying assumptions. A good allocation requires the existence of an accurate cost model. Necessary for this cost model, as well as for proper execution, is the notion of a shared coordinate space, as well as the ability to accurately localize within this coordinate space. Additionally, the environment was also visually simplified to allow higher probability of treasure discovery; both the Pioneer and the treasure were outfitted

with identifying visual fiducials. Work in progress includes implementing and testing treasure retrieval with human-robot participation, and with robots creating and allocating tasks to other robots and humans.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we presented the concept of pickup teams, where teams are formed dynamically from heterogeneous robots with little prior information about one another. We have also presented an appropriate domain for exploring the research issues related to pickup teams – multi-robot treasure hunts. The TraderBots allocation mechanism is coupled with play mechanisms to achieve dynamic sub-team formation, and efficient allocation of tasks and roles. Thus, we have proposed a new approach to address the research challenge of dynamic and efficient formation, coordination, and interaction of multi-robot pickup teams. A first version of this approach has been implemented and tested on two heterogeneous robots developed by two different research groups. Future work will enhance this preliminary implementation in several dimensions including improved robustness, more complex environments, teams and tasks, and learned adaptation to dynamic conditions.

ACKNOWLEDGMENT

This work is principally funded by the Boeing Company Grant CMU-BA-GTA-1. The content of the information in this publication does not necessarily reflect the position or policy of the Boeing Company and no official endorsement should be inferred. The authors would also like to thank the Boeing Company for their contributions to this project and this paper. 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).

REFERENCES

- [1] I. Noda, S. Suzuki, H. Matsubara, M. Asada, and H. Kitano, "RoboCup-97: The first robot world cup soccer games and conferences," *AI Magazine*, vol. 19, no. 3, pp. 49–59, Fall 1998.
- [2] P. Scerri, Y. Xu, E. Liao, J. Lai, and K. Sycara, "Scaling teamwork to very large teams," in *AAMAS'04*, 2004.
- [3] G. Kaminka and I. Frenkel, "Flexible teamwork in behavior-based robots," in *In Proceedings of the National Conference on Artificial Intelligence (AAAI-2005)*, 2005.
- [4] M. Koes, I. Nourbakhsh, and K. Sycara, "Heterogeneous multirobot coordination with spatial and temporal constraints," in *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI)*. AAAI Press, 2005, pp. 1292–1297.
- [5] D. V. Pynadath and M. Tambe, "An automated teamwork infrastructure for heterogeneous software agents and humans," *Autonomous Agents and Multi-Agent Systems*, vol. 7, no. 1-2, pp. 71–100, 2003.
- [6] M. Bowling, B. Browning, and M. Veloso, "Plays as effective multiagent plans enabling opponent-adaptive play selection," in *Proceedings of International Conference on Automated Planning and Scheduling (ICAPS'04)*, 2004.
- [7] M. B. Dias, "Traderbots: A new paradigm for robust and efficient multirobot coordination in dynamic environments," Ph.D. dissertation, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, January 2004.
- [8] R. G. Smith, "The contract net protocol: High level communication and control in a distributed problem solver," *IEEE Transactions on Computers*, vol. C-29, no. 12, pp. 1104–1113, 1980.