TraderBots: A Market-Based Approach for Resource, Role, and Task Allocation in Multirobot Coordination

M. Bernardine Dias and Anthony (Tony) Stentz

CMU-RI-TR-03-19 2

TraderBots: A Market-Based Approach for Resource, Role, and Task Allocation in Multirobot Coordination

M. Bernardine Dias
Anthony (Tony) Stentz

CMU-RI -TR-03-19

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

August 2003

Abstract

The problem of efficient multirobot coordination has risen to the forefront of robotics research in recent years. Interest in this problem is motivated by the wide range of application domains demanding multirobot solutions. In general, multirobot coordination strategies assume either a centralized approach, where a single robot/agent plans for the group, or a distributed approach, where each robot is responsible for its own planning. Inherent to many centralized approaches are several difficulties. The key advantage of centralized approaches is that they can produce globally optimal plans. While most distributed approaches can overcome the obstacles inherent to centralized approaches, they can only produce suboptimal plans. This work presents the philosophy and traces the development of "TraderBots": a market-based architecture that is inherently distributed, but also capable of opportunistically forming centralized sub-groups to improve efficiency. Robots are self-interested with the primary goal of maximizing individual profits. The revenue/cost models and rules of engagement are designed so that maximizing individual profit has the benevolent effect of moving the team toward the globally optimal solution.

Table of Contents

Table of Figures

1. Introduction

In this digital age, the demand for technological solutions to increasingly complex problems is climbing rapidly. With this increase in demand, the tasks which robots are required to execute also rapidly grow in variety and difficulty. A single robot is no longer the best solution for many of these new application domains; instead, teams of robots are required to coordinate intelligently for successful task execution. For example, a single robot is not an efficient solution to automated construction, urban search and rescue, assembly-line automation, mapping/investigation of unknown/hazardous environments, and many other similar tasks. Multirobot solutions are paramount for several reasons:

1. **A single robot cannot perform some tasks alone**, a team is required for successful execution. While in many cases it may be possible to design a single robot capable of executing all tasks, many problems are better suited to team-execution. For example, a single robot can accomplish moving heavy objects if the robot is designed appropriately. However, in many cases, it is simpler to design a team of robots that cooperate to move the heavy objects efficiently. Other application domains such as robotic soccer require a team of robots and cannot be executed with a single robot.

2. **A robot team can accomplish a given task more quickly** than a single robot can by dividing the task into sub-tasks and executing them concurrently in application domains where the tasks can be decomposed. Application domains such as mapping of unknown areas and searching for landmines require careful coverage of a large area. Problems such as these can be easily decomposed into components such that a team of robots can divide the workload and execute sub-portions of the task concurrently, thus completing the overall task more efficiently.

3. **A team can make effective use of specialists** designed for a single purpose (for example, scouting an area, picking up objects, or hauling payload), rather than requiring that a single robot with versatile capabilities be a generalist, capable of performing all tasks. This allows more flexibility in designing the robots since a robot that needs to haul heavy payloads can be built with a heavy base for stability and strength, while a robot required to provide visual feedback can be designed to be more agile and move around with greater speed.

4. **A team of robots can localize themselves more efficiently** if they exchange information about their position whenever they sense each other. This allows more robust localization capabilities. In an environment where a single robot would have to rely on landmarks of some sort for localization, a team could have the added advantage of being able to benefit from the localization information of their teammates.

5. **A team of robots generally provides a more robust solution** by introducing redundancy, and by eliminating any single point of failure as long as there is overlap between the robots' capabilities. For example, a team of robots equipped with cameras, will be a more reliable system than a single robot with a camera for constructing vision-based maps of a dynamic environment because the failure of a single one of the robots in the team will not jeopardize the entire mission.

6. **A team of robots can produce a wider variety of solutions** than a single robot can, and hence a team can opportunistically respond to dynamic conditions in more creative and efficient ways. Even if a team of robots does not overlap entirely in terms of specialization, the collective resources of the group can be used in creative ways to solve problems. For example, if a diagnostic robot loses its camera during operation, another robot with a camera could aid the diagnostic robot to complete its tasks by providing visual feedback [28]. Similarly, if a rover gets stuck in the mud, one or more of its teammates can assist the stuck robot by pushing it out of the mud [28].

Thus, for many applications, a team of robots can be used more effectively than a single robot can. Introduced below are some of the more prominent application domains that would benefit from efficient coordination of multirobot systems:

- **Autonomous robot colonies for operations in remote locations:**

  Many applications in the future (for example extra-planetary exploration) will require a colony of robots to autonomously execute complex tasks, while humans intervene remotely from time to time to alter the procedure of operations, remedy a situation beyond the capabilities of the robots, or coordinate with the robots to accomplish additional goals.

- **Robotic aids for urban reconnaissance:**

  Military Operations in Urban Terrain pose fierce constraints and the use of biochemical threats against both land forces and indigenous population in urban settings is an increasing likelihood. These conditions place humans in a highly dangerous environment. Robots can enable minimally invasive and precise operations that reduce these risks. Potential tasks for robotic systems include minesweeping, reconnaissance, monitoring, and providing communications infrastructure.

- **Robotic aids for urban search and rescue:**

  Urban Search and Rescue (USAR) workers have forty eight hours to find trapped survivors in a collapsed structure, otherwise the likelihood of finding victims still alive is nearly zero. Also, the mechanics of how large structures collapse often prevent rescue workers from searching buildings due to the unacceptable personal risk and the added risk to survivors from further collapse of the building. Furthermore, both people and dogs are frequently too big to enter voids. Robots can make a significant impact in this domain if made capable of aiding humans in USAR efforts.

- **Intelligent environments:**

  Intelligent Environments are spaces in which computation is seamlessly integrated to enhance ordinary activity. Many familiar environments such as office buildings and schoolrooms are highly likely to incrementally evolve into intelligent environments. In these environments, automated agents will oversee optimized utilization of the resources, resolve conflicts regarding resource utilization, and keep track of maintenance requirements for all resources.

- **Automated construction:**

  Automated construction involves the assembly of large-scale structures, such as terrestrial buildings, planetary habitats, or in-space facilities. Such domains need heavy lifting capabilities, as well as precise, dexterous manipulation to connect parts together. Future space facilities, characterized by their immense size and the difficulties of human construction in space will be assembled in part by groups of autonomous heterogeneous robots.

- **Robotic educational and entertainment systems:**

  Robotic toys, educational tools, and entertainment systems are rapidly gaining popularity. Many of these systems will require coordinated efforts by multiple robots. An example in this domain is robotic soccer.

- **Automated production plants:**

  A growing trend in production plants is automation. In order to increase production, decrease labor costs, improve efficiency, increase safety, and improve quality in general, more and more industries are seeking to automate their production facilities. This trend demands efficient and robust coordination of heterogeneous multirobot systems.

- **Robotic exploration of hazardous environments:**

  Exploration of hazardous environments has long been a problem demanding robotic solutions. Some examples in this category are exploration of extra-planetary regions, exploration of volcanic regions, exploration of disaster zones, and exploration of minefields.

- **Robotic cleanup of hazardous sites:**

  Robots continue to play an important role in cleanup of hazardous sites. Some examples in this domain are robotic minesweeping, robotic cleanup of nuclear waste, and robotic cleanup of disaster zones.

- **Agricultural Robots:**

  Many groups involved with agricultural work are now seeking automated solutions to their labor problems. Due to the long hours, hard physical work in rough conditions, and tedious and repetitive nature of some of the tasks in this domain, there is a growing decline in the available labor. Spraying fields, harvesting, moving plant-containers (potted plants), and sorting plants are some such examples. For many of these tasks, coordinated teams of robotic agricultural machines could provide efficient solutions.

Thus, for many applications, multirobot systems can improve efficiency. However, simply increasing the number of robots does not improve efficiency in itself. Thus, the problem of efficient multirobot coordination has risen to the forefront of robotics research in recent years. The wide range of application domains demanding multirobot solutions motivates interest in this problem. In general, multirobot coordination strategies span a spectrum of methodologies ranging from fully centralized systems to fully distributed reactive systems. A centralized approach entails a single robot planning for the entire group based on state information gathered from the group, whereas a distributed approach is where each robot is responsible for its own planning. Inherent to many centralized approaches are difficulties such as intractable solutions for large groups, sluggish response to changes in the local environment, heavy communication requirements, and brittle systems with single points of failure. The key advantage of centralized approaches is that they can produce globally optimal plans. While most distributed approaches can overcome the obstacles inherent to centralized approaches, they typically produce sub-optimal plans. Moreover, distributed coordination schemes are often fortuitously cooperative and hence do not allow for explicit coordination. This limits the capability of the system since the robots are prevented from performing more tightly coordinated tasks.

To realize the best characteristics from both approaches, we developed a market-based approach, "TraderBots", that is inherently distributed, but can also opportunistically form centralized sub-groups to improve efficiency, and thus approach optimality ([31], [10], [11]). Robots are self-interested agents, with the primary goal of maximizing individual profits. The revenue and cost models and rules of engagement are designed so that maximizing individual profit has the benevolent effect of moving the team towards the globally optimal solution. This architecture inherits the flexibility of market-based approaches in allowing cooperation and competition to emerge opportunistically. The outlined approach is ideally suited to solve the multirobot coordination problem for autonomous robotic colonies carrying out complex tasks in dynamic environments where it is desirable to optimize to whatever extent possible.

2. Illustrative Scenario[1]

A complex multirobot scenario is the problem of robotic exploration of Mars. For the foreseeable future, mobile robots will serve as the remote sensors and data collectors for scientists. To create an outpost for such long-term exploration, the robots need to assemble solar power generation stations, map sites, collect science data, and communicate with Earth on a regular basis. Envision the scenario illustrated in Figure 1 where on the order of ten robots are sent to Mars, many with different capabilities. Some of the robots specialize in heavy moving and lifting, some in science data collection, some in drilling and coring, and some in communication. The rovers have different, but overlapping, capabilities.

The rovers cooperatively search for a location suitable in size and terrain for a base station. Once such a location is found, rovers with appropriate capabilities form several teams to construct the base station capable of housing supplies and generating energy. Two rovers carry parts, such as solar panels, that are too large for a single rover. Complementary capabilities are exploited.

Meanwhile, other rovers begin general exploration of the region. To start, several scouting robots (perhaps joined by aerial vehicles) quickly survey the region. Scientists on Earth (and perhaps the rovers

[1] Motivation for this scenario was obtained from the FIRE team's proposal to NASA for the "Heterogeneous Multi-Rover Coordination for Planetary Exploration" program.

themselves) identify sites within the region that have high likelihood to contain interesting science data. Rovers with specialized sensing instruments are sent to investigate. If a particular subtask requires more intensive scrutiny, additional rovers with appropriate capabilities are brought in. Rover failures are addressed by dispatching a rover with diagnostic capabilities. The diagnostic rover can use its cameras to view the failed robot to see if it can be aided in the field, or it may drag the rover back to the base station to be repaired by replacing its failed modules. In the meantime, another robot with the same (or similar) capabilities can be substituted, so as to complete the original task with minimal interruptions.

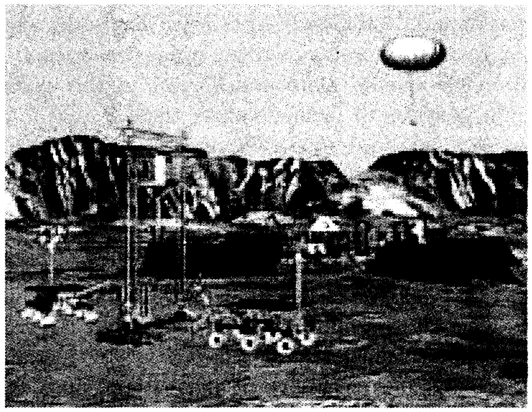

Figure 1: Conceptual Illustration of a Multirobot Martian Outpost (Illustration produced by courtesy of Jet Propulsion Laboratory)

At any given time, different teams of rovers may be involved in exploration, base-station construction/maintenance, and rover diagnosis/repair. Many tasks will be time critical, requiring execution within hard deadlines or synchronization with external events. The teams form dynamically, depending on the task, environment, and capabilities and availability of the various robots to best meet mission requirements over time. The rovers negotiate their individual roles, ensure safety of the group and themselves, and coordinate their actions, attempting as a group to avoid unnecessary travel-time, to minimize reconfiguration and wait-time, and to prefer more reliable alternatives in cases of overlapping capabilities. The challenge is to keep all the robots healthy and busy in appropriate tasks in order to maximize the scientific data collected.

A scenario like this is illustrative of a challenging multirobot application that demands high quality performance. Hence, a multirobot coordination scheme suitable for an application domain such as this must fulfill many requirements. The following characteristics describe these requirements for a coordination mechanism able to successfully execute the motivational scenario:

1. **Robustness**: *Robust to robot failure, or no single point of failure for the system.*
   This is an important characteristic since many applications rely on continued progress even if some components in the system fail. The motivational scenario expects that several robots will malfunction or be destroyed during operation, and still require the overall mission to be completed in the best way possible given the remaining resources.
2. **Dynamic Conditions**: *Opportunistically optimized response to dynamic conditions.*
   This characteristic is desirable in general, and required in some domains. Since the scenario described above involves dynamic conditions, the ability to opportunistically optimize the system response to these conditions is necessary for efficiency and success.
3. **Speed**: *Quick response to dynamic conditions.*
   Often in dynamic environments, a key to successful task execution is the ability to respond quickly to the dynamic conditions. If information always needs to be channeled to another location for plan modification, conditions can change too rapidly for the planning to keep up.

4. **Extensibility**: *Easily extendable to accommodate new functionality.*
   A key characteristic to building a generalized system that can evolve with the needs of the different applications is the ability to easily add and remove functionality as needed. This is identified as extensibility.
5. **Communication**: *Ability to deal with limited and imperfect communication.*
   In general, many application domains cannot realistically guarantee perfect communication among all robots at all times. Hence, any generalized coordination architecture should be robust to communication failures and limits in range of communication.
6. **Resources**: *Ability to reason about limited resources.*
   The ability to reason about the limited resources available in a robotic system is very important for optimization purposes. For example, in the illustrative scenario, it is undesirable to use the only robot with some very costly science sensor to perform a simple but risky scouting task. Also, when a robot is assigned a task, the planner must understand the resource requirements for that task in order to allow efficient task-allocation. The planner must also take into account events scheduled to occur in the future and the resources that will be required for those events before committing to any new tasks.
7. **Task Allocation**: *Efficient allocation of tasks.*
   A key difficulty in coordinating multiple robots is deciding who does what. Thus, the task allocation mechanism is an important factor in the architectural design. Factors such as robot capabilities and resources, risk involved with different tasks, and task constraints need to be considered in order to maximize the efficiency of the task allocation.
8. **Heterogeneity**: *Ability to accommodate heterogeneous teams of robots.*
   Many architectures assume homogeneity for ease of planning. The coordination problem is more difficult if the robots are heterogeneous. A successful architecture will be able to accommodate any team regardless of its homogeneity or heterogeneity.
9. **Roles**: *Efficient adoption of roles.*
   In many architectures robots are restricted to being able to play only a single role in the team at any given time, even if they possess the resources to be able to play multiple roles simultaneously. Efficient role adoption will enable robots to play as many roles as required at any given time based on resource availability, and also allow robots to change in and out of different roles as conditions change. Some roles may require the robots to work in tight coordination with other robots.
10. **New Input**: *Ability to dynamically handle new tasks, resources, and roles.*
    In many dynamic application domains, the demands on the robotic system can change during operation. Hence, it may become necessary to assign new tasks, change existing tasks, add new resources, or introduce new roles. All of this would ideally be supported by the architecture.
11. **Flexibility**: *Easily adaptable for different applications.*
    Since different applications will have different requirements, a general architecture will need to be easily configurable for the different problems it proposes to solve. Instructions and advice on how to reconfigure the architecture for different applications will also be useful.
12. **Fluidity**: *Easily able to accommodate the addition/subtraction of robots during operation.*
    Several applications could require the ability to introduce new robots into the system during operation. Conversely, robots can exit or malfunction during task execution. A successful architecture will be able to support such events gracefully.
13. **Learning**: *On-line adaptation for specific applications.*
    While a generalized system is often more useful, its application to specific domains usually requires some parameter tuning. The ability to tune relevant parameters automatically in an on-line fashion is thus a very attractive feature that can save a lot of effort.
14. **Implementation**: *Implemented and proven on physical system.*
    As with any claim, a proven implementation is far more convincing. Moreover, successful implementation of an architecture on a robotic system requires discovering and solving many details that are not always apparent in simulation and software systems.

3. Related Work

Many research groups ([6], [21], [5], [27]) have implemented centralized approaches for multirobot coordination. The principal advantage of such centralized approaches is that optimal plans can be produced. The leader can take into account all the relevant information conveyed by the members of the team and generate an optimal plan for the team. However, centralized approaches suffer from several disadvantages as detailed in the Introduction.

Local and distributed approaches address the problems that arise with centralized, globally coordinated methods by distributing the planning responsibilities among all members of the team. Many research efforts have modeled distributed systems inspired by biology ([1], [4]). Others have designed systems based on fluidics and similar physics-based concepts ([2], [7]). Some have chosen to pursue rule-based, heuristic-based and model-based approaches ([9], [18], [33]). Economy-based models have inspired still others ([14], [25], [30]). However, the principal drawback of distributed approaches is that they often result in highly sub-optimal solutions because all plans are based solely on local information.

Smith [30] first introduced the concept of using an economic model to control multi-agent systems as the Contract Net protocol. Many groups have since adopted similar strategies for controlling multi-agent systems. Work done by Krovi et al. [23], Faratin et al. [14], Jung et al. [22], Brandt et al. [3], Wellman and Wurman [36], Smith [30], Gibney et al. [16], Collins et al. [8], Jennings and Arvidsson [20], Sandholm [25], Tumer, Agogino, and Wolpert [35], and Sycara and Zeng [32] are examples of economy-based sofware-agent systems. In contrast, work done by Simmons et al. [28], Dias and Stentz [10], Gerkey and Matarić [15], and Golfarelli et al. [19] are examples of economy-based coordination approaches applied to multirobot systems. Many characteristics differentiate software-agent domains from situated-agent (robotic) domains. Some principal differences between robotic systems and software systems are highlighted next.

Tasks assigned to robotic agents can vary significantly from tasks in software domains. Also, robotic agents often deal with more restricted resources and robotic systems often have to deal with more restricted communication. Failures occur with higher frequency and in a wider variety in robotic systems. Furthermore, robotic systems have to be able to accommodate larger error bounds in performance since they often deal with faulty sensors and interact with real-world environments. Finally, robotic systems often require more creative solutions to recover from faults (for example, one robot pushing another robot that is stuck, two robots cooperating to lift a heavy obstacle, etc.). Thus, controlling multirobot systems can be a significantly different problem compared to controlling multiple software agents.

Hence, a multirobot coordination approach has to take into account many robot-specific details. Our market-based coordination mechanism aims to satisfy all of these requirements.

4. The TraderBots Approach

Stentz and Dias [31] first introduced the concept of using a market approach to coordinate multiple robots to cooperatively complete a task, building on the contract net protocol by Smith [30], its extension by Sandholm and Lesser [26], and the general concepts of market-aware agents developed by Wellman and Wurman [36]. This work introduced the methodology of applying market mechanisms to intra-team robot coordination (i.e. in typically non-competitive environments) as opposed to competitive multirobot domains and competitive inter-agent interactions in domains such as E-commerce. Simulation results using this approach were produced by Dias and Stentz [10], and proven robot results were presented by Thayer et al. [34], and Zlot et al. [37]. A few other groups have also published research relevant to market-based multirobot coordination. Golfarelli and Rizzi [19] proposed and implemented a swap-based negotiation protocol for multirobot coordination that restricted negotiations to task-swaps, and Gerkey and Matarić [15] developed the MURDOCH publish/subscribe mechanism that includes a single-round auction for task distribution. Rabideau et al. [24] published a comparison study of three multi-rover coordination mechanisms that included a contract-net-based approach. However, to date, no other group has explored, in detail, a market-based approach to multirobot coordination. A brief summary of our approach is presented next.[2]

Consider a team of robots assembled to perform a particular set of tasks. Consider further, that each robot in the team is modeled 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 (which can translate to minimizing individual cost where possible); however, since all revenue is derived from satisfying team objectives, the robots' self-interest equates to doing global good. Moreover, all robots can only increase their profit by eliminating unnecessary waste (i.e. excess cost). Hence, if the global cost is determined by the summation of individual robot costs, each deal made by a robot (note that robots will only make profitable deals) will result in global cost reduction. 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.

4.1 Revenues, Costs, the Role of Price and the Bidding Process

Appropriate functions are needed to map possible task outcomes onto revenue values and to map possible schemes for performing the task onto cost values. As a team, the goal is to execute some plan such that the overall profit (the excess of revenue over cost) is maximized. Furthermore, these functions must provide a means for distributing the revenue and assessing costs to individual robots.

Thus, robots receive revenue and incur costs for accomplishing a specific team-task, but the team's revenue function is not the only source of income. A robot can also receive revenue from another robot in exchange for goods or services. The *price* dictates the payment amount for the good or service. A common approach is to *bid* for a good or service in order to arrive at a mutually acceptable price.

4.2 Cooperation vs. Competition

Two robots are *cooperative* if they have complementary roles; that is, if both robots can make more profit by working together than by working individually. Conversely, two robots are *competitive* if they have the same role; that is, if the amount of profit that one can make is negatively affected by the presence of the other robot. The flexibility of the market-model allows the robots to cooperate and compete as necessary to accomplish a task efficiently.

4.3 Self Organization, Learning and Adaptation

Conspicuously absent from the market approach is a rigid, top-down hierarchy. Instead, the robots organize themselves in a way that is mutually beneficial. Since the aggregate profit amassed by the individuals is directly tied to the success of the task, this self-organization yields the best results.

Moreover, the robot economy is amenable to learning new behaviors and strategies as it executes its complex global task. An added strength of the market approach is its ability to deal opportunistically with dynamic environments.

4.4 Architectural Format

The envisioned structure for the architecture is shown below in Figure 2. The illustration is tailored to the distributed mapping application and shows the architectural format envisioned for each robot in the team. It is organized in layers. In the bottom layer are the resources under the robot's control, such as sensors, computers, and communication devices. These resources are available to the robot to perform its tasks—some unused resources can be leased to other robots in the team if there is a demand for them. For example, if a robot is not using its entire computing capacity, it can do another robot's data processing for a fee. The next layer consists of the agent's roles for accomplishing tasks. Roles are application-specific

[2] For a detailed description of this market approach please see Dias and Stentz [11].

software modules that implement particular robot capabilities or skills, such as acting as a communication router or generating optimal plans for the team as a leader. The roles utilize resources in the layer below to execute their tasks. Roles execute tasks that match their specific capabilities. They receive assignments from the trader and could be monitored by an executive. As they execute their tasks they may generate other tasks or subtasks to be bid out to the other robots. These new tasks will be communicated to the trader.

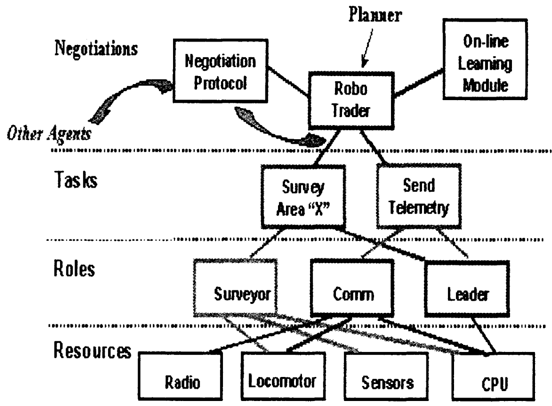

Figure 2: Architectural management of resources, tasks, and roles on a single robot

At the top layer in the architecture, the RoboTrader coordinates the activities of the agent and its interactions with other agents. All of the planning is carried out at this top layer. The trader bids on tasks for the robot to perform and offers tasks for sale. It passes on tasks it wins to an executive who matches tasks to roles, schedules the roles to run, and resolves any contention for resources. The trader could be equipped with an on-line learning module that enables it to perform better over time by adapting to the specific application and environment. But the architecture for a single robot does not complete the picture. Figure 3 below illustrates potential high-level interaction between a group of robots and two users:

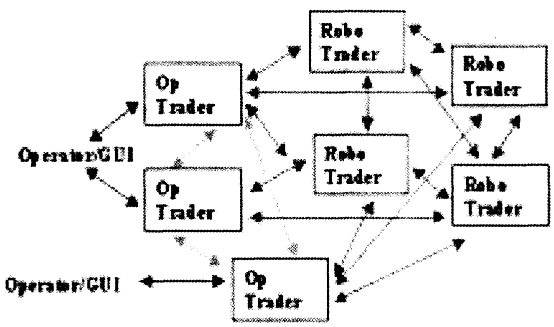

Figure 3: Interaction between robots and operator

As shown above, the operators can communicate high-level tasks to the interface agents known as the "Operator Traders" (or the OpTraders). The OpTraders then interpret the operator's commands and translate them into tasks that the robots can recognize. Next these tasks are bid out to the RoboTraders on the robots within communication range. The OpTraders could also negotiate amongst themselves.

4.5 Satisfying the Identified Requirements

The TraderBots approach addresses the identified requirements of a successful multirobot coordination approach as follows:

1. **Robustness:**
   Since the TraderBots approach has no single leader it has no single point of failure. Note that although a single OpTrader could represent the operator, if one OpTrader fails for some reason, a new one could be initiated and it could gather information from the RoboTraders to assess their current operational status. Thus, although some information may be lost with the failure of agents or robots, the system performance will degrade gracefully and the robot team will always aim to complete the assigned tasks with the functioning resources if possible.
2. **Dynamic Conditions**:
   The TraderBots approach will respond to dynamic conditions by accounting for these conditions during negotiations. For example, if new conditions arise which make an assigned task no longer profitable for a robot it will try to sell that task to another robot who may find it profitable due to/in spite of the new conditions.
3. **Speed**:
   Since each robot in this approach will make decisions for itself, the system as whole can respond more quickly to dynamic conditions than if new information had to be conveyed to a different agent and the robot had to wait for the new plan before acting.
4. **Extensibility**:
   The TraderBots approach allows for the addition and subtraction of different levels of functionality in a modular fashion since tasks and resources and roles are described in a modular fashion.
5. **Communication**:
   The TraderBots approach does not assume any guarantees about communication. Robots make trades with any robot within communication range.
6. **Resources**:
   The available resources on the robots are always taken into account when determining costs and savings for tasks being traded in this approach. Furthermore, the TraderBots approach allows for an executive that can monitor the addition and depletion of resources and roles on the robots and notify the trader of changes to the robot capability so that the current capabilities of the robot are taken into account during negotiations.
7. **Task Allocation**:
   In the TradeBots approach, actors such as robot capabilities, risks, and task constraints can be considered when determining costs and savings for tasks during trading. Note that different cost and revenue functions can result in different allocations of tasks. Hence, designing appropriate cost and revenue functions can be very important in this approach. However, imperfect cost and revenue functions could be altered to some extent via learning.
8. **Heterogeneity**:
   The TraderBots approach makes no assumptions about the heterogeneity or homogeneity of the team.
9. **Roles**:
   In this approach, robots are not restricted to being able to play a limited number of roles at any given time. Robot capabilities are derived at any time from available resources at that time. If a robot needs to switch out of a current role, it can decide to do so by selling the task that requires it to play that role or deciding to default on its commitment and paying a penalty.
10. **New Input**:
    Adding new resources and introducing new roles can be handled by the TraderBots approach as long as the robots are able to detect the changes in resources and roles when evaluating their current capabilities. New task assignments can be handled if they can be communicated to a capable robot, and changes to existing tasks can be handled as long as the change is communicated to the relevant robot in time for the change to be made before execution.

11. **Flexibility**:
    The TraderBots approach is not specifically geared to a single application domain. While the modularity of the TraderBots approach allows the approach to be applied to different task domains, an on-line learning capability could enhance the flexibility of the approach by autonomously tuning the market parameters to adapt to the application domain.
12. **Fluidity**:
    If a new robot enters the team, it can join the team activities by participating in any on-going trading. If the failure of a robot can be detected (by means of a heart-beat or occasional pinging of the robot) all tasks assigned to that robot could be re-auctioned to other robots thereby assuring the completion of tasks as long as the necessary resources are available.
13. **Learning**:
    A learning module could allow the robots to autonomously tune the market parameters according to the prevailing conditions, and thereby improve efficiency.
14. **Implementation**:
    The TraderBots approach has been implemented and tested in simulation and on robot teams to different extents. These implementations and tests are summarized next.

5. Implementations

Previously published work includes an implementation of the market-based architecture that was developed and tested for a distributed sensing task in a simulated interior environment ([10] and [34]) and also on a group of Pioneer II-DX robots [37].

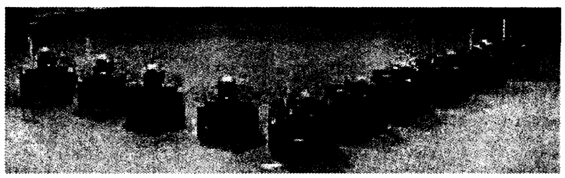

Figure 4: Team of Pioneer II-DX robots

In this implementation, the TraderBots market approach seeks to maximize benefit (information gained) while minimizing costs (in terms of the collective travel distance), thus aiming to maximize utility. The system is robust in that exploration is completely distributed and can still be carried out if some of the team members lose communications or fail completely. The effectiveness of this approach was demonstrated through successful mapping results obtained with the team of robots. Zlot et al. [37] found that by allowing the robots to negotiate using the market architecture, exploration efficiency was improved by a factor of 3.4 for a four-robot team.

The authors implemented an initial version of the leader role as a combinatorial exchange, and tested, in simulation, the advantages of multi-party and multi-task negotiations in the TraderBots approach [12]. Results show a significant advantage to be gained from trading in clusters of tasks. Results further show the TraderBots approach capable of producing task allocations within 10% of the optimal allocation for simple distributed sensing problems.

A third implementation focuses on the space application domain, and more specifically, presents simulation results for market-based coordination of a group of heterogeneous robots engaged in information gathering on a Martian outpost.

In this implementation the market-based, multi-robot planning capability, is designed as part of a distributed, layered architecture for multi-robot control and coordination[3]. More specifically, this architecture is an extension to the traditional three-layered robot architecture (illustrated in Figure 5) that

[3] See Goldberg et al. [17], and Simmons et al. [29] for more details about this layered architecture.

enables robots to interact directly at each layer – at the behavioral level, the robots create distributed control loops; at the executive level, they synchronize task execution; at the planning level, they use the TraderBots approach to assign tasks, form teams, and allocate resources.

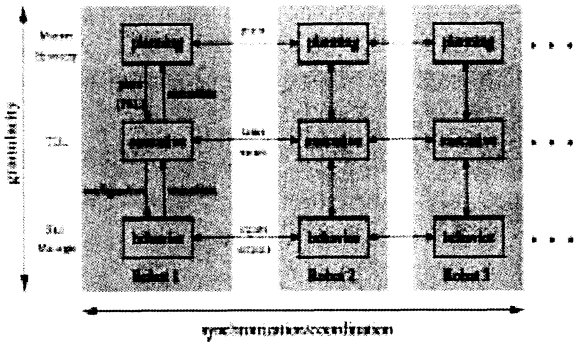

Figure 5: Extended three-layer architecture

This implementation is tested using a 3D graphical simulator developed for the project (see Figure 6).

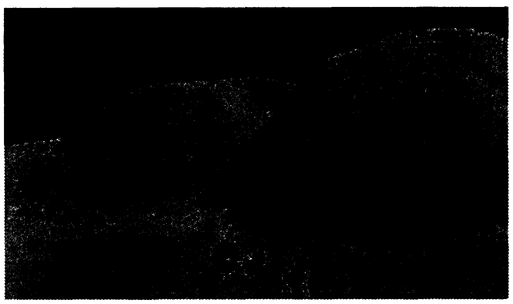

Figure 6: Screen shot from 3D graphical simulator

The market-based planning layer of each robot has two main components: a "trader" that participates in the market, auctioning and bidding on tasks, and a "scheduler" that determines task feasibility and cost for the trader, and interacts with the executive layer for task execution. The focus of the development and testing of the current system has been on a characterize task that will fit within the broader scenario of the Martian outpost. In this task, a user/scientist specifies a region on the Mars surface, indicating that rocks within that region are to be characterized with an appropriate sensing instrument. The scientist may also specify the locations of rocks, if known.

In a fourth implementation, Zlot and Stentz [37] investigate task abstraction using tree structures within the market framework. The participants in the market are permitted to bid on nodes representing varying levels of task abstraction, thereby enabling distributed planning, task location, and optimization among the robot team members. Results in simulation demonstrate that this approach can introduce a significant improvement on the total solution cost for the team.

6. Conclusion and Future Work

The authors are currently implementing a comparative study between three multirobot coordination schemes that span the spectrum of coordination approaches: a fully centralized approach that can produce optimal solutions, a fully distributed behavioral approach with minimal planned interaction between robots, and the TraderBots approach which sits in the middle of the spectrum.[4]

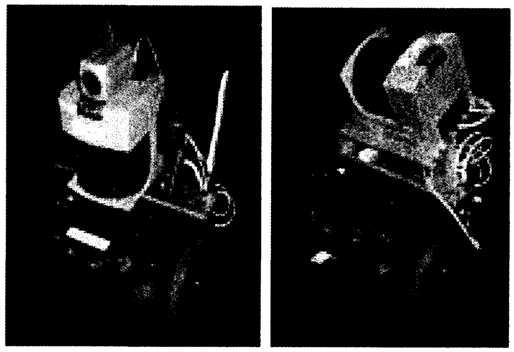

Figure 7: Pinoeer II-DX robots enhanced with vertically or horizontally mounted Sick lasers

A team of researchers at Carnegie Mellon University is currently working on extending the implementation of the TraderBots approach on the Pioneer II-DX robot team. The robots have been enhanced by adding vertically mounted Sick lasers on some of them, and horizontally mounted Sick lasers on others (see Figure 7). This group aims to extend the implementation in many ways including the addition of an executive to manage resources and roles on each robot, the introduction of robots capable of playing a variety of roles in complex scenarios, and the inclusion of tasks that require tight-coordination between robots. The goal of this work is to produce a fully functional market-based coordination mechanism capable of efficient and robust multirobot coordination.

7. References

[1] Arkin, R. C., "Cooperation without Communication: Multiagent Schema-Based Robot Navigation", Journal of Robotic Systems, Vol. 9, No.3, pp. 351-364, 1992.

[2] Böhringer, K., Brown, R., Donald, B., Jennings, J., and Rus, D., "Distributed Robotic Manipulation: Experiments in Minimalism", Proceedings of the International Symposium on Experimental Robotics (ISER), 1995.

[3] Brandt, F., Brauer, W., and Weiß, G., "Task Assignment in Multiagent Systems based on Vickrey-type Auctioning and Leveled Commitment Contracting", Cooperative Information Agents IV, Lecture Notes in Artificial Intelligence, Volume 1860, pp.95-106, 2000.

[4] Brooks, R. A., "Elephants Don't Play Chess", Robotics and Autonomous Systems, Vol. 6, pp.3-15, 1990.

[5] Brumitt, B. L., Stentz, A., "Dynamic Mission Planning for Multiple Mobile Robots", Proceedings of the IEEE International Conference on Robotics and Automation, No. 3, pp. 2396-2401, 1996.

[6] Caloud, P., Choi, W., Latombe, J-C., Le Pape, C., and Yim, M., "Indoor Automation with Many Mobile Robots", IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.67- 72, 1990.

[7] Chevallier, D., and Payandeh, S., "On Kinematic Geometry of Multi-Agent Manipulating System Based on the Contact Force Information", The 6th International Conference on Intelligent Autonomous Systems (IAS-6), pp.188-195, 2000.

[4] First comparison results have been submitted for publication [13].

[8] Collins, J., Jamison, S., Mobasher, B., and Gini, M., "A Market Architecture for Multi-Agent Contracting", Technical Report 97-15, University of Minnesota, 1997.

[9] Decker, K. S., and Lesser, V. R., Designing A Family Of Coordination Algorithms, Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95), pp. 73-80, 1995.

[10] Dias, M. B. and Stentz, A., "A Free Market Architecture for Distributed Control of a Multirobot System", proceedings of the 6th International Conference on Intelligent Autonomous Systems (IAS-6), 2000.

[11] Dias, M. B. and Stentz, A., "A Market Approach to Multirobot Coordination", Technical Report, CMU-RI -TR-01-26, Robotics Institute, Carnegie Mellon University, 2001.

[12] Dias, M. B. and Stentz, A., "Opportunistic Optimization for Market-Based Multirobot Control", IEEE/RSJ International Conference on Intelligent Robots and Systems, 2002.

[13] Dias, M. B. and Stentz, A., "A Comparative Study between Centralized, Market-Based, and Behavioral Multirobot Coordination Approaches", submitted to IEEE/RSJ International Conference on Intelligent Robots and Systems, March 2003

[14] Faratin, P., Sierra, C., Jennings, N. R., "Negotiation decision functions for autonomous agents", International Journal of Robotics and Autonomous Systems, Volume 24(3-4), pp. 159-182, 1997.

[15] Gerkey, B. P. and Matarić, M. J., "Sold! Market methods for multi-robot control", IEEE Transactions on Robotics and Automation Special Issue on Multi-Robot Systems, submitted March 2001.

[16] Gibney, M. A., Jennings, N. R., Vriend, N. J., and Griffiths, J. M., "Market-Based Call Routing in Telecommunications Networks Using Adaptive Pricing and Real Bidding", 3rd Inernational Workshop on Multi-Agent Systems and Telecommunications (IATA-99), pp.50-65, 1999.

[17] Goldberg, D., Cicirello, V., Dias, M. B., Simmons, R., Smith, S., Smith, T., and Stentz, A., "A Distributed Layered Architecture for Mobile Robot Coordination: Application to Space Exploration", Proceedings of the 3rd International NASA Workshop on Planning and Scheduling for Space, 2002.

[18] Goldman, C. V., and Rosenschein, J. S., "Emergent Coordination through the Use of Cooperative State-Changing Rules", Proceedings of the Twelfth National Conference on Artificial Intelligence, pp.408-413, 1994.

[19] Golfarelli, M., Maio, and D., Rizzi, S., "A Task-Swap Negotiation Protocol Based on the Contract Net Paradigm", Technical Report, 005-97, CSITE (Research Centre For Informatics And Telecommunication Systems), University of Bologna, 1997.

[20] Jennings, B., and Arvidsson, Å., "Co-operating Market/Ant Based Multi-agent Systems for Intelligent Network Load Control", Proceedings of the 3rd International Workshop on Intelligent Agents for Telecommunication Applications (IATA), LNAI, Vol. 1699, pp. 62-75, 1999.

[21] Jensen, R. M., Veloso, M. M., "OBDD-based Universal Planning: Specifying and Solving Planning Problems for Synchronized Agents in Non-Deterministic Domains", Lecture Notes in Computer Science, No. 1600, pp. 213-248, 1999.

[22] Jung, H., Tambe, M., and Kulkarni, S., "Argumentation as Distributed Constraint Satisfaction: Applications and Results", Proceedings of the 5th International Conference on Autonomous Agents, 2001.

[23] Krovi, R., Graesser, A. C., and Pracht, W. E., "Agent Behaviors in Virtual Negotiation Environments", IEEE Transactions on Systems, Man, and Cybernetics, Volume 29, pp.15-25, 1999.

[24] Rabideau, G., Estlin, T., Chien, S., Barrett, A., "A Comparison of Coordinated Planning Methods for Cooperating Rovers", Proceedings of AIAA Space Technology Conference, Albuquerque, NM, 1999.

[25] Sandholm, T., "An Implementation of the Contract Net Protocol Based on Marginal Cost Calculations", Proceedings, Eleventh National Conference on Artificial Intelligence (AAAI-93), pp. 256-262, 1993.

[26] Sandholm, T., and Lesser, V., "Issues in Automated Negotiation and Electronic Commerce: Extending the Contract Net Framework", Proceedings of the first International Conference on Multiagent Systems (ICMAS-95), 1995.

[27] Simmons, R., Apfelbaum, D., Fox, D., Goldman, R. P., Haigh, K. Z., Musliner, D. J., Pelican, M., and Thrun, S., "Coordinated Deployment of Multiple, Heterogeneous Robots", Proceedings of the Conference on Intelligent Robots and Systems (IROS), Takamatsu Japan, 2000.

[28] Simmons, R., Singh, S., Hershberger, D., Ramos, J., and Smith, T., "First Results in the Coordination of Heterogeneous Robots for Large-Scale Assembly", Proceedings of the International Symposium on Experimental Robotics (ISER), 2000.

[29] Simmons, R., Smith, T., Dias, M. B., Goldberg, D., Hershberger, D., Stentz, A., and Zlot, R. M., "A Layered Architecture for Coordination of Mobile Robots", Multi-Robot Systems: From Swarms to Intelligent Automata, Proceedings from the 2002 NRL Workshop on Multi-Robot Systems, 2002.

[30] Smith, R., "The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver", IEEE Transactions on Computers Vol. C-29, No. 12, 1980.

[31] Stentz, A. and Dias, M. B., "A Free Market Architecture for Coordinating Multiple Robots", Technical Report, CMU-RI-TR-99-42, Robotics Institute, Carnegie Mellon University, 1999.

[32] Sycara, K., and Zeng, D., "Coordination of Multiple Intelligent Software Agents", International journal of Intelligent and Cooperative Information Systems, Volume 5(2-3), pp.181-211, 1996.

[33] Tambe, M., "Towards Flexible Teamwork", Journal of Artificial Intelligence Research, Vol. 7, pp.83-124, 1997.

[34] Thayer, S. M., Dias, M. B., Digney, B. L., Stentz, A., Nabbe, B., and Hebert, M. 2000. Distributed robotic mapping of extreme environments. Proceedings of SPIE: Vol. 4195: Mobile Robots XV and Telemanipulator and Telepresence Technologies VII.

[35] Tumer, Kagan, Agogino, Adrian K., and Wolpert, David, "Learning sequences of actions in collectives of autonomous agents", First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), pp.378-385, 2002.

[36] Wellman, M. P., and Wurman, P. R., "Market-Aware Agents for a Multiagent World", Robotics and Autonomous Systems, Volume 24, pp.115-125, 1998.

[37] Zlot, R., Stentz, A., Dias, M. B., and Thayer, S. 2002. Multi-Robot Exploration Controlled By A Market Economy. Proceedings of the IEEE International Conference on Robotics and Automation.

[38] Zlot, Robert and Stentz, Anthony, "Market-based Multirobot Coordination Using Task Abstraction", International Conference on Field and Service Robotics, to appear in July 2003.