An Experiment in Mobile Robotic Cooperation

Lynne E. Parker[1]

Abstract

This paper describes the results of an experiment in real mobile robot cooperation which utilizes a fully distributed, cooperative software mechanism we have previously developed in simulation. This experiment — an artificial toxic waste cleanup mission — illustrates the ability of our software architecture to create robust and reliable teams of cooperative mobile robots that are able to function in dynamic, changing environments. By demonstrating this ability, we bring the goal of applying robots to real applications in challenging environments closer to realization.

1 Introduction

A key driving force in the development of robotic systems is their potential for reducing the need for human presence in dangerous work environments. The nature of many of these challenging work environments requires that such robotic systems be able to work fully autonomously in achieving human-supplied goals. One approach to developing these systems is to develop a single robot that can accomplish particular goals in a given environment. However, the complexity of many environments or missions may require a mix of robotic capabilities that is too extensive to design into a single robot. Additionally, time constraints may require the use of multiple robots working simultaneously on different aspects of the mission in order to successfully accomplish the objective. Thus, we must build teams of heterogeneous robots that can work together to accomplish a mission that no individual robot can accomplish alone.

Since such cooperative teams will often be working in dynamic and unpredictable environments, we must design the software control systems of these robot teams to allow the robots to respond robustly and reliably to unexpected events. These software control mechanisms must provide the means for robot teams to handle unexpected environmental changes and modifications in the robot team that may occur due to mechanical failure, the learning of new skills, or the addition or removal of robots from the team by human intervention.

[1]MIT Artificial Intelligence Laboratory, 545 Technology Square, Room 709, Cambridge, MA 02139. internet: parkerl@ai.mit.edu

Related work in cooperative mobile robotics (refer to [RIK, 1992] for a wide selection of papers in this area) has dealt predominantly with the study of large numbers (often called *swarms*) of robots, where the primary issue is how to obtain the desired emergent global behavior of the robot swarm from simple, local control laws. This approach to robotic cooperation is useful for non-time-critical applications involving numerous repetitions of the same activity over a relatively large area, such as cleaning a parking lot or collecting rock samples on Mars. Such approaches usually rely on mathematical convergence results that indicate the desired outcome over a sufficiently long period of time. However, many other real-world tasks require a more directed type of cooperation due to time constraints that are placed on the mission. This second type of mobile robotic mission usually requires that several distinct tasks be performed, often on a time-critical basis. Examples of this type of application include automated manufacturing, industrial/household maintenance, search and rescue, and security, surveillance, or reconnaissance tasks. Several researchers have addressed this second type of cooperative mission, developing software architectures that typically stress the allocation of tasks to robots through negotiation, but then ignore or only give brief treatment to the issues of robot performance of those tasks after they have been allocated. Such approaches usually assume the robots will eventually accomplish the task they have been assigned, or that some external monitor will provide information to the robots on dynamic changes in the environment or in robot performance. However, to realistically design a cooperative approach to robotics, we must include mechanisms within the software control of each robot that allow the team to recover from dynamic changes in the environment or in the robot team.

Our research addresses these needs for cooperating mobile robots by developing software mechanisms that allow groups of heterogeneous robots to work together to robustly and reliably accomplish their mission. In previous work [Parker, 1992], we have described an architecture that allows a team of robots to quickly adapt their actions to a dynamic environment, to modifications in the robot team composition, or to changes in the capabilities of the individual robot team members. In addition, we have developed a mechanism, described in [Parker, 1993], that allows these teams of robots to learn from their previous experiences with other robots, allowing them to better select their own actions on subsequent trials when working with "familiar" robots. This mechanism allows a human system designer to easily and quickly group the appropriate combination of robots together for a particular mission, since the robots need have no *a priori* knowledge of each other.

However, the results we presented in our previous work come from *simulated* robot experiments, such as a janitorial service application and a "bounding overwatch" (military surveillance) application. Previous experience in robot development [Brooks, 1990] indicates that approaches to robot control which work in simulated robot worlds are often not successful when applied to real mobile robots, due to unrealistic assumptions made in the simulations. In this paper, we describe the results of implementing and validating our architecture on a heterogeneous group of physical mobile robots performing an artificial toxic waste cleanup mission[2]. We show that the control mechanisms indeed allow the group

[2] By *artificial*, we mean that we pretend that the objects the robots manipulate are hazardous waste. We do not actually apply these robots to real toxic waste spills, since our robots are simply small laboratory research testbeds, and are in no way designed specifically for such missions.

of robots to adapt their actions to a changing environment, to modifications in the team composition, or to changes in robot capabilities.

Section 2 briefly reviews our cooperative robot architecture, followed by a description of our pool of heterogeneous mobile robots in section 3. We then, in section 4, describe the results of applying our architecture to an artificial hazardous waste cleanup mission using a subset of our pool of heterogeneous mobile robots. Section 5 offers some concluding remarks and describes our ongoing and future work.

2 Overview of the Cooperative Architecture

Our software architecture for heterogeneous robot cooperation is a fully distributed architecture that utilizes adaptive action selection to achieve cooperative control. The robots in this architecture possess a variety of high-level functions that they can perform during a mission, and must at all times select an appropriate action based on the requirements of the mission, the activities of other robots, the current environmental conditions, and their own internal states. Adaptive action selection is achieved through the selfish interests of individual robots, modified by their analyses of the current and previous performances of other team members. The *performance* of a robot is determined solely by how that robot affects the world, and is not dependent upon explicit, often artificial, skill metrics.

Individual robots in our system are designed using a behavior-based approach [Brooks, 1986]. Under this construction, a number of task-achieving behaviors are active simultaneously, each receiving sensory input and controlling some aspect of the actuator output. The lower-level behaviors correspond to primitive survival behaviors, such as obstacle avoidance, while the higher-level behaviors correspond to higher goals, such as map building and exploring. Extensions to this approach are necessary, however, when a robot must select among a number of competing actions — actions which cannot be pursued in parallel. Unlike typical behavior-based approaches, this architecture delineates several "behavior sets" that are either active as a group or are hibernating. Each behavior set corresponds to those levels of competence required to perform some high-level task, such as cleaning the floor or emptying the garbage. Due to conflicting goals, only one of these behavior sets are active at any point in time. However, other lower-level competences, such as collision avoidance, may be continually active regardless of the high level goal the robot is currently pursuing. Because of the alternative goals that may be pursued by the robots, they must have some means of selecting the appropriate behavior set to activate.

The primary mechanism for achieving adaptive action selection in this architecture is called the *motivational behavior*. Each motivational behavior receives input from a number of sources, including sensory feedback, inter-robot communication, inhibitory feedback from other active behaviors, and internal motivations (such as impatience and acquiescence). The output of a motivational behavior is the activation level of its corresponding behavior set, represented as a non-negative number. When this activation level exceeds a given threshold, the behavior set becomes active; once a behavior set is activated, other behavior sets are suppressed, so that only one behavior set is active in an individual robot at a time. This mechanism incorporates the ability for robots to ignore the communication of other robots if those robots do not demonstrate, through their effect on the world, that they can accomplish their task. It also provides robots with the ability to abandon their own current task if they sense that they are not successful in accomplishing it.

Intuitively, a motivational behavior works as follows. The motivation to perform a given behavior set is initialized to some number reflecting the priority of that behavior set. This priority indicates the relative importance of the behavior set among all behavior sets of the robot. Then, over time, the motivation for performing a behavior set increases as long as the corresponding task is not being accomplished, as determined from sensory feedback. Additionally, we want our robots to be responsive to the actions of other robots, adapting their task selection to the activities of team members. Thus, if a robot is aware that another robot is performing a particular task, the first robot should be satisfied for a time that that task is going to be performed even without its own participation in the task, and thus go on to some other applicable action. Of course, detecting and interpreting the actions of other robots is not a trivial problem for robots, and often requires perceptual abilities that are not yet possible with current sensing technology. Thus, to enhance the robots' perceptual abilities, this architecture utilizes a simple form of broadcast communication to allow robots to inform other team members of their current activities, rather than relying totally on sensing through the world. Each robot broadcasts a statement of its current actions at a pre-specified rate, which other robots may use or ignore as they wish. No two-way conversations are employed in this architecture.

Each robot is designed to be somewhat impatient, however, in that it is only willing for a certain period of time to allow the communicated messages of other robots to affect its motivation to activate a behavior set. Continued sensory feedback indicating that a task is not getting accomplished thus overrides the statements of another robot that it is performing that task. This characteristic allows robots to adapt to failures of other robots, causing them to ignore the activities of a robot that is not successfully completing its task. In this manner, impatience allows robots to adapt to dynamic changes in the environment or in the actions of other robots.

A complementary characteristic in these robots is that of acquiescence. This characteristic allows a robot to give up its current activity after a certain period of time if another robot has begun performing the same task. Just as the impatience characteristic reflects the fact that other robots may fail, the acquiescence characteristic indicates the recognition that a robot itself may fail. As a robot performs a task, its willingness to give up the task increases over time as long as the sensory feedback indicates the task is not being accomplished. As soon as some other robot indicates it has begun that same task, the unsuccessful robot gives up that task in an attempt to find an action at which it is more productive. In this way, a robot adapts its actions to its own failures.

The design of the motivational behaviors also allows the robots to adapt to unexpected environmental changes which alter the sensory feedback. The need for additional tasks can suddenly occur, requiring the robots to perform additional work, or existing environmental conditions can disappear, and thus relieve the robots of certain tasks. In either case, the motivations fluidly adapt to these situations, causing robots to respond appropriately to the current environmental circumstances.

We have also added a learning capability to this software mechanism that allows robots to change their behavior from trial to trial as they learn about other team member capabilities in order to achieve even greater performance improvements. Refer to [Parker, 1993] for a formal mathematical model of this architecture, including the learning component.

Figure 1: Three types of robots are being used in our heterogeneous robot cooperation experiments. From left to right, these robots are Genghis-II, R-1, and R-2.

3 Our Pool of Heterogeneous Robots

One of the goals of our research is to allow a human system designer to create new teams of cooperative robots by selecting, from a pool of available robots, those robots which have the proper mix of capabilities for the current application. This newly composed team can then set out to successfully perform its mission even without *a priori* knowledge of other team members' abilities. However, as the robots perform their missions, they learn about the abilities of other robots and adapt their actions accordingly on subsequent trials in order to perform their mission more efficiently whenever familiar robots are present.

To enable us to demonstrate this capability, we have composed a pool of three types of mobile robots, from which we can create various teams with differing group capabilities. Shown in figure 1, our pool of robots consists of three types of mobile robots — Genghis-II, R-1, and R-2. Although our laboratory has 20 of the R-1 type of robot, 5 of the R-2 type of robot, and 1 Genghis-II, we do not expect to use more than 5 to 8 total robots during any given experiment.

The first type of robot, Genghis-II, is a legged robot with six two-degree-of-freedom legs. Its sensor suite includes two whiskers, force detectors on each leg, a passive array of infrared heat sensors, three tactile sensors along the robot belly, four near-infrared sensors, and an inclinometer for measuring the pitch of the robot. The second robot type, the R-1, is a four-wheeled vehicle with a magnet and gripper for grasping small metallic objects. Contact switches at the tip of the gripper allow the robot to detect collisions. Six infrared sensors on the gripper allow the robot to detect objects in front of the gripper, below the gripper, and between the two fingers of the gripper, for use in finding and grasping objects. The third type of robot, the R-2, has two drive wheels arranged as a differential pair, and a two-degree-of-freedom gripper for grasping objects. The sensor suite consists of a color sensor in the fingers for detecting object color, plus eight infrared sensors and seven bump sensors evenly distributed around the front, sides, and back of the robot.

Note that even though we have duplicate copies of the R-1 robot type and the R-2 robot type, only rarely do robots of the same type possess the same sensory and effector capabilities, due to significant variations in the sensitivity and accuracy of their sensors and effectors. Thus, we can compose a heterogeneous robot team consisting of only one type of robot, since the robot behavior varies noticeably from robot to robot.

A radio communication system is used to allow the robots to communicate with each other. This radio system consists of a transceiver unit attached to each robot, plus two base stations for use in triangulating the robot positions and in sending and receiving the robot messages. The positioning system is accurate to about $\frac{1}{2}$ foot, and is useful for allowing robots to know their own position with respect to their environment and with respect to other robot team members.

4 An Experiment in Cooperation

4.1 The Application: Hazardous Waste Cleanup[3]

Imagine that there has been a hazardous waste spill in an enclosed rectangular room of a building that must be cleaned up. In our case, the spill consists of a number of small cylindrical objects clustered in one area of the room. Our robot team must locate the spill and move it to the far end of the room, where we assume the waste material can be dealt with more easily by humans. The robot team must also periodically report its progress to humans monitoring the system. Since we assume that radio transmission will not work through the walls of the room, this status report must be performed by having a robot representative return to the room entrance occasionally to radio the team's current mission completion status to the human monitor located outside the room. If the monitoring human does not hear from the robots after a certain period of time, the human will assume the robots are having trouble, and will send in more robots to help with the mission.

A difficulty in this mission is that the human monitor does not know the exact location of the spill in robot coordinates, and can only give the robot team qualitative information on the initial location of the spill and the final desired location to which the robots must move the spill. In this case, the robots are told that the initial location is in the center of the front third of the room, and that the desired final location of the spill is in the back, center of the room, relative to the position of the entrance. We assume that the two base stations for use in robot positioning are easily placed inside the room at the entrance. The robot team's ultimate goal is to complete this mission as quickly as possible without needlessly wasting energy.

4.2 The Robot Team

In this experiment, we compose our team of two R-2 robots, RED and BLUE. Since these robots are of the same type, they have the potential of maximum redundancy in capabilities. However, even though these two robots were originally designed to be identical, mechanical drift and failures have caused them to have quite different actual abilities, in terms of the effect they can have on the world. For example, RED has use of its side infrared (IR) sensors, which

[3]Of course, when working with real mobile robots, one is always limited in the applications that can be demonstrated by the physical limitations of the available robots. Thus, this experiment has been designed specifically with the capabilities of our robots in mind, as described in the previous section.

allow it to perform wall-following, whereas BLUE's side IR sensors have become dysfunctional. The learning system gives these robots the ability to determine from trial to trial which team member is best suited for which task.

The robots have been preprogrammed to have the following behavior sets, which correspond to high-level tasks that must be achieved on this mission: *find-initial-final-locations-methodical*, *find-initial-final-locations-wander*, *move-spill*, and *report-progress*. A low-level *obstacle-avoidance* behavior is active at all times in these robots, except during portions of the *move-spill* task, when it is suppressed to allow the robot to pick up the spill object. Additional lower-level behaviors that are used at various times in one or more behavior sets are *wall-following*, *go-to-location*, and *search-for-spill-object*.

Two behavior sets are provided which both accomplish the task of finding the initial and final spill locations: *find-initial-final-locations-methodical* and *find-initial-final-locations-wander*. The methodical version is much more reliable, and involves the robot following the walls of the room back to the start location while tracking the minimum and maximum x and y positions it reaches. It then uses these x, y values to calculate the coordinates of the center of the front third of the room (for the initial spill location), and the back center of the room (for the final spill location). The wander version does not require the use of the robot's side IR sensors; instead, it involves wandering in each of the four directions (east, north, west, and south) for a certain time period, tracking the minimum and maximum x and y positions as in the methodical version, and then calculating the desired initial and final locations.

4.3 Results

Figure 2 shows the actions selected by each robot on a typical trial of this experiment. Prior to this trial, the learning system in each robot has allowed both RED and BLUE to determine that BLUE cannot successfully accomplish the task corresponding to *find-initial-final-locations-methodical*. At the beginning of the mission, RED has the highest motivation to perform behavior set *find-initial-final-locations-methodical*, causing it to initiate this action. This causes BLUE to be satisfied for a while that the initial and final spill locations are going to be found; since no other task can currently be performed, BLUE sits waiting for the two locations to be found. (Recall that the robots periodically broadcast a statement of their current actions to the other team members using their radio systems, which allows the team members to determine the current actions of other robots.) However, BLUE will not sit forever waiting on the locations to be found. It becomes more and more impatient over time, which can cause it to activate its own *find-initial-final-locations-wander* if RED does not successfully locate the spill. Note that BLUE will not activate its own *find-initial-final-locations-methodical* because it has learned in previous trials that that action does not achieve the desired effect. Indeed, BLUE did overtake RED at this task when we intentionally interfered with RED's progress on another trial. However, in this trial, RED does indeed complete the task, and reports the two locations to the rest of the team.

At this point, the environmental feedback and knowledge of the initial and final spill locations indicate to both robots that the *move-spill* behavior set is applicable. Since this is a task that can be shared, both robots begin searching for a spill object in the initial spill area and taking that object to the desired final spill location. In the meantime, their motivations to report their progress are increasing, so that once a robot has delivered a spill object to the destination, that robot (in this case, BLUE) becomes motivated to report the team's progress,

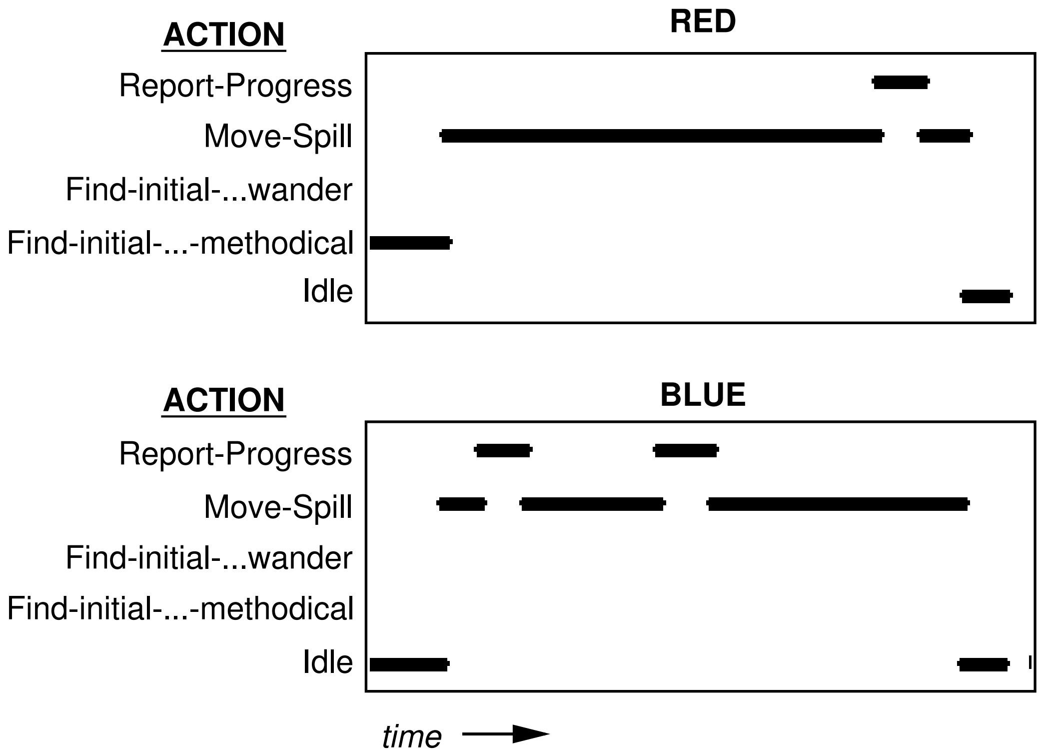

Figure 2: Robot actions selected during experiment.

and thus activates the *report-progress* behavior set. This satisfies the rest of the team (namely, RED) that the progress will be reported, so RED then re-activates the *move-spill* behavior. This series of actions is repeated until all of the spill is moved, and the mission is complete.

4.4 Discussion

This experiment illustrates the primary characteristics we consider important in developing cooperative robotic teams. First of all, the cooperative team is robust, in that robots are allowed to continue their actions only as long as they demonstrate their ability to have the desired effect on the world. This was illustrated in our example by BLUE becoming gradually more impatient with RED's search for the spill. If RED did not locate the spill in a reasonable length of time, then BLUE would overtake that task, with RED acquiescing the task to BLUE. Secondly, the cooperative team is able to respond autonomously to many types of unexpected events in either the environment or in the robot team without the need for external intervention. At any time in this mission, we could disable one of the robots, causing the remaining robot to perform those tasks that the disabled robot would have performed. Clearly, we could also have easily increased or decreased the size of the spill during the mission, and the robots would not be adversely affected. Third, the cooperative team need have no *a priori* knowledge of the abilities of the other team members to effectively accomplish the task. However, our learning system does allow the team to improve its efficiency on subsequent trials whenever familiar robots are present. This was illustrated in BLUE's willingness to allow RED to attempt to find the spill because of BLUE's knowledge that RED was superior in this task. In a similar vein, the learning system would also allow BLUE and RED to learn about BLUE's improved performance in finding the spill if BLUE's faulty sensors were repaired between missions.

5 Conclusions and Future Work

The primary goal in our research is to create a software architecture that allows a group of heterogeneous robots, with different but overlapping skills, to work cooperatively, robustly, and reliably to accomplish a mission in a challenging environment. Although in earlier work we had developed an architecture that meets this goal, we had previously only validated it in simulation. In this paper, we have described the results of an artificial toxic waste cleanup experiment on actual robots which illustrates the success of our architecture in meeting our design goal.

One possible shortcoming of this approach is that it assumes that the cooperative mission is composed of loosely coupled subtasks with very little dependence amongst tasks. Thus, an ongoing topic of research for us is to extend this software architecture to apply to missions composed of more highly coupled subtasks. Additionally, we are examining this architecture analytically to better determine its expected performance in a wide variety of applications.

Acknowledgements

Support for this research was provided in part by the University Research Initiative under Office of Naval Research contract N00014-86-K-0685, in part by the Advanced Research Projects Agency under Office of Naval Research contract N00014-85-K-0124, and in part by the Mazda Corporation.

References

[Brooks, 1986] Rodney A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, RA-2(1):14–23, March 1986.

[Brooks, 1990] Rodney A. Brooks. Challenges for complete creature architectures. In Jean-Arcady Meyer and Stewart W. Wilson, editors, *Proceedings of the First International Conference on Simulation of Adaptive Behavior*, pages 434–443. MIT Press, 1990.

[Parker, 1992] Lynne E. Parker. Adaptive action selection for cooperative agent teams. In Jean-Arcady Meyer, Herbert Roitblat, and Stewart Wilson, editors, *Proceedings of the Second International Conference on Simulation of Adaptive Behavior*, pages 442–450. MIT Press, 1992.

[Parker, 1993] Lynne E. Parker. Learning in cooperative robot teams. In *Working Notes of the IJCAI Workshop on Dynamically Interacting Robots*, Chambery, France, August 1993.

[RIK, 1992] RIKEN; The Institute of Physical and Chemical Research. *Proceedings of the International Symposium on Distributed Autonomous Robotic Systems*, Wako, Saitama, Japan, September 1992.