INEEL/EXT-01-00232

Command And Control Architectures For Autonomous Micro-Robotic Forces

FY-2000 Project Report

D. D. Dudenhoeffer
D. J. Bruemmer

April 2001

Idaho National Engineering and Environmental Laboratory
Bechtel BWXT Idaho, LLC

INEEL/EXT-01-00232

Command and Control Architectures for Autonomous Micro-Robotic Forces

FY00 Project Report

D. D. Dudenhoeffer
D. J. Bruemmer

Published April 2001

Idaho National Engineering and Environmental Laboratory
Human Systems Engineering and Sciences Department
Idaho Falls, Idaho 83415

Prepared for the
U.S. Department of Energy
Defense Advanced Projects Research Agency
Under DOE Idaho Operations Office
Contract DE-AC07-99ID13727

ABSTRACT

Advances in Artificial Intelligence (AI) and micro-technologies will soon give rise to production of large-scale forces of autonomous micro-robots with systems of innate behaviors and with capabilities of self-organization and real world tasking. Such organizations have been compared to schools of fish, flocks of birds, herds of animals, swarms of insects, and military squadrons. While these systems are envisioned as maintaining a high degree of autonomy, it is important to understand the relationship of man with such machines. In moving from research studies to the practical deployment of large-scale numbers of robots, one of critical pieces that must be explored is the command and control architecture for humans to re-task and also inject global knowledge, experience, and intuition into the force. Tele-operation should not be the goal, but rather a level of adjustable autonomy and high-level control. If a herd of sheep is comparable to the collective of robots, then the human element is comparable to the shepherd pulling in strays and guiding the herd in the direction of greener pastures. This report addresses the issues and development of command and control for large-scale numbers of autonomous robots deployed as a collective force.

ACKNOWLEDGMENTS

A special thanks is extended to the following people who provided valuable insight and direction in this research project: Dr. Lynne Parker of Oak Ridge National Laboratory and Dr. Douglas Gage of the Defense Advanced Research Projects Agency. Additionally, the efforts of the following people were instrumental in the completion of this project: Dr. Susan Hill, Ph.D., Mr. Jim Byers, Mr. Mark McKay, and Mr. Matthew Anderson, of the Idaho National Engineering and Environmental Laboratory. Donovan Bramwell provided technical writing support for the report.

This work is supported through DARPA contracts J932 and J933, with additional support provided through the INEEL Long-Term Research Initiative Program under DOE Idaho Operations Office Contract DE-AC07-99ID13727.

CONTENTS

FIGURES

TABLES

ACRONYMS

DARPA	Defense Advanced Research Projects Agency
DOD	Department of Defense
DOE	Department of Energy
DR	Distributed Robotics
INEEL	Idaho National Engineering and Environmental Laboratory
ITO	Information Technologies Office
LDRD	laboratory-directed research and development
SDR	Software for Distributed Robotics
UAV	Uninhabited Aerial Vehicle
UCAV	Uninhabited Combat Air Vehicle
USAF	United States Air Force

Command and Control Architectures for Autonomous Micro-Robotic Forces FY00 Project Report

1. INTRODUCTION

Advances in robotics will soon give rise to the development of a diverse array of small to miniature robots capable of autonomous travel through air, in water, and on land. Coupled with advanced sensor and transmission technologies, these units have tremendous potential in intelligence gathering applications, especially in filling current gaps in intelligence collection, both in times of peace and times of conflict. A key element in the transition from tabletop development to field deployment is the role of the human operator and the necessary interaction between the robotic force and human during the mission. The words that General Bruce C. Clarke stated in 1959 hold true with regard to the introduction of robotics in military and non-military applications: “The truth is that the most expensive weapon that technology can produce is worth not an iota more than the skill and will of the man who uses it” (National Research Council 1997, p.22). The successful use of large numbers of robots in field applications depends on the ability for human operators to interact with the robot force in exchanging information, providing direction, and gaining an understanding of the intent and operations at both the microscopic and the macroscopic levels.

Researchers at the Idaho National Engineering and Environmental Laboratory (INEEL) are engaged in a research project that addresses this need. The objective of the project is to identify, develop, and evaluate various command and control architectures that permit continuous, real-time human user interaction with large-scale micro-robotic forces as a collective entity (including the capacity to task and query), rather than requiring the human operator to interact with each and every individual robot. These control architectures must address situations in which the number of autonomous units makes the individual control of units by a single point neither feasible nor desirable. This project has conducted the initial design and prototyping of a command and control structure based on hierarchical design. The command and control system provides the capability to inject human-supplied domain knowledge to supplement the autonomous nature of the robots. Providing on-the-fly ad hoc tasking and re-tasking, the system allows operator-specified levels of abstraction to permit interactions at the individual, group, or collective level. Additionally, the system provides a set of tools for assisting the operator in maintaining situational awareness. The command and control system has been implemented in *AgentSim*, a simulation environment developed at the INEEL. This document is the final report for the first year’s tasking.

1.1 Program Purpose

This INEEL project is sponsored by the Information Technologies Office (ITO) directorate of the Defense Advanced Research Projects Agency (DARPA) under the Software for Distributed Robotics (SDR) program. The stated mission of this program is to “develop the missing software needed to enable the employment and control of large numbers of small, distributed, mobile robots so as to achieve *large scale results from many small scale robots*.” The technical challenges and goals associated with this mission are the development of the required software technologies to enable:

1. A large collection of micro-robots that can move, communicate, and work collectively to achieve a collective goal.

2. Human/robot interface technologies that permit the human to interact with the robots as a group (including the capacity to task and query), rather than requiring the human operator to interact with each and every individual robot.

This project reflects INEEL's contribution to address the second goal of human robot interface technologies, specifically in the development of command and control concepts.

1.2 Background

This effort represents one of the first research projects that examines and attempts to develop a multi-robot human control for large numbers of robots. Dr. Douglas Gage (1992) initially explored the area of large-scale multi-robot command and control, but very little has been done since. Several command and controls systems have been developed to support small numbers of robotic forces (i.e., 4 to 15). The most notable systems include MissionLab, Demo III, and the Uninhabited Combat Air Vehicle command system.

MissionLab, a multi-robot mission specification system developed at Georgia Tech under the direction of Dr. Ronald Arkin, is probably the best known example (Arkin 1998). It has been used as part of the Tactical Mobile Robots program at DARPA. While providing the necessary control elements with small numbers of robots, MissionLab, does not facilitate control of large numbers of robots. Specifically, it does not support levels of force abstraction in its command and control scheme. The ability to create abstract levels of control, i.e. groups and units, is a key element in permitting the human operator "one-to-many" control of large numbers of robots.

Demo III is a program sponsored by the Department of Defense (DOD) to develop and demonstrate small autonomous ground vehicles. The primary operator interface is through Operator Control Units. The system facilitates mission planning, task execution, and re-tasking through a Windows-like map-based interface (Morgenthaler et al. 2000). A drawback, however, is that like MissionLab, Demo III does not provide a mechanism for command over large numbers of units. An operator can control at most four vehicles from one Operator Control Unit.

The objective of the Uninhabited Combat Air Vehicle (UCAV) command system developed by the United States Air Force (USAF) is to provide a human operator an interface mechanism for controlling multiple UCAVs in coordinated mission execution. The operator is responsible for establishing mission goals, monitoring system status, and refining task execution. Operational studies on this system have revealed that a single operator can manage at most four UCAVs simultaneously, as long as the mission execution does not deviate significantly from the original plan (Barbato 2000).

MissionLab, Demo III, and the USAF UCAV command and control systems are sufficient for directing small numbers of autonomous vehicles, but they do not have the capability to handle large numbers of robots as envisioned by the SDR program. Within the SDR and the Distributed Robotics (DR) programs, additional "command and control" mechanisms exist or are being developed, but their primary focus is directed at facilitating laboratory development, not field deployment. In contrast, the focus of our research project is a field-deployable system with soldiers and sailors as the primary end users. As laboratory development progresses toward manufacturing fieldable units, we hope that the command and control concepts developed by the INEEL can be integrated to create a deployable system.

AgentSim is a simulation environment developed by the INEEL. It allows for the evaluation of individual and collective behaviors for large numbers of autonomous robots. Additionally, it provides a test interface for the development and evaluation of command and control concepts for use with large numbers of autonomous robots. The simulation package was originally developed as part of an internal research project at the INEEL to evaluate individual and collective behaviors for large robotic forces. Further enhancements have been added as part of this DARPA project to explore human interface and control issues.

1.3 INEEL Concurrent Research and Investment

This DARPA project has leveraged significantly from an internal INEEL laboratory-directed research and development (LDRD) project being conducted, entitled "Behavioral and Control Modeling for Large Scale Micro-Robot Systems."

The objective of the LDRD project is to research, develop, and determine the applicability of different behavioral and control systems models for use in deploying large (upwards of 1000) numbers of micro-robots in collective tasking. Specifically, this research centers on developing models for self-organization of the units into specific formations or groups and developing the human operator control mechanisms necessary to manipulate them. This multi-year research program at the INEEL represents a $240,000 investment to date.

2. INEEL PROJECT TASKING

INEEL-specific tasking includes the development and evaluation of various command and control architectures for use by humans in the deployment of large-scale micro-robotic forces. Specific areas to be examined include shared control by multiple users, arbitration of control between users, and collaboration and cooperation between autonomous units. The control architectures will address the situation in which the magnitude of autonomous unit numbers makes the individual control of units by a single point neither feasible nor desirable. The goal is to develop a control system for the collective tasking and control of a large-scale force of mobile autonomous units. The system will be based on innate goal-directed behaviors with the capability to inject human-supplied domain knowledge and direction.

2.1 INEEL Project Tasks

The scope of the initial INEEL work is to study and evaluate a hierarchical-based command and control structure. Three specific tasks of the project are:

1. Develop the concepts and control structures for adapting a hierarchical model in the command and control of large-scale forces (>1000) of autonomous micro-robots.

2. Incorporate the design concepts developed in Task (1) into a computer simulation of a deployed force of autonomous micro-robots. Use the results of the simulation to evaluate and refine the control structure and to identify the advantages and limitations of the command structure.

3. Present the research findings, via the deliverables described in the following subsection.

2.2 Project Deliverables

The two stated deliverables for this project are:

1. A final report detailing the command and control architecture(s) evaluated by this project. The report will include any control algorithms developed as part of the research. Additionally, the report will discuss the advantages and limitations of each control schema evaluated. This document is the deliverable report.

2. A system demonstration of a command and control system using computer modeling and simulation of the control structures. The demonstration was presented at a program review held in Knoxville, TN in October 2000. The command structures were incorporated within *AgentSim*, a simulation environment developed at the INEEL as part of a related, internally funded research project.

3. COMMAND AND CONTROL SYSTEM REQUIREMENTS

Recent conferences and program meetings have revealed a general misunderstanding on the nature of command and control, specifically in its application to military operations. The DOD defines command and control as:

> The exercise of authority and direction by a properly designated commander over assigned and attached forces in the accomplishment of the mission. Command and control functions are performed through an arrangement of personnel, equipment, communications, facilities, and procedures employed by a commander in planning, directing, coordinating, and controlling forces and operations in the accomplishment of the mission. The command and control process is a continuous sense, assess, decide and act cycle executed in order to accomplish an assigned mission (DOD Dictionary of Military Terms).

The misconception that often arises in discussions with people who lack military experience is that the design of command and control systems for robotic forces is equivalent to a global planning program or a computer graphical-user-interface for issuing control signals. The core of the command and control process is decision making, and the focus is not on the robots, but on the operator, who must integrate the robotic force in dynamic mission tasking. Unlike environments such as MissionLab, Saphira, TeamBots 2.0, Allyu, and Charon, where command must be implicitly wrought through *a priori* development of a static control architecture, *AgentSim* seeks to meet operational needs by emphasizing continuous, real-time command at several levels of abstraction.

In a recent review of the philosophical and practical challenges related to the command and control process, the National Research Council offered the following observation:

> Commanders must take relevant knowledge and combine it with their judgment – including difficult-to-quantify aspects of human behavior (such as fatigue, experience level, and stress), the uncertainty of data, and the plausible future states resulting from actions by both their own force and the enemy – to make decisions about future actions and how to convey those decisions in ways to facilitate their proper execution. (National Research Council 1999, p.31)

DOD training on the subject of command and control describes the process as both an art and a science. Table 1 (Source: Combined Arms and Service Staff School) provides a summary that separates the aspects of mere control from those of decision-making support.

Table 1. The art and science of command and control.

Art (Command)	Science (Control)
• Commander's Domain	• Staff's domain
• Visualizing the Battlefield	• Determining requirements
• Ranking priorities	• Allocating means
• Formulating Concepts	• Defining limits
• Assigning Missions,	• Monitoring status
• Leading, guiding, and motivating the organization	• Developing specific guidance from general guidance

In short, one of the fundamental requirements of a command and control architecture for deploying a large-scale force of mini-robots is that it must accommodate and enhance the decision-making role of its human operators.

3.1 Current Research into Autonomous Robot Command and Control

Although great strides have been made in technology, the introduction of autonomous robotic forces into military applications has yet to be realized. Radio controlled, tether controlled, and some semi-autonomous robotic platforms have been used by the military for surveillance and intelligence. The interaction with human operators has been primarily on a one to one or a one to several (<10) robot level.

The potential for a large numbers of robots deployed as a collective force represents tremendous capability in terms of area coverage, redundancy, and time savings. By the same token, however, it presents a nightmare in terms of control and monitoring of the collective.

This problem was identified some time ago and remains an ongoing concern (Gage 1992; Lee 2000). Although much work in the past few years has explored the utility of distributed control concepts, little research has been conducted to develop robust, scalable command and control tools for interfacing humans with a large-scale force of robots. If distributed approaches are to be deployed across military, humanitarian, and commercial domains, there is an acute need for further consideration of human factors. Recent Defense Science Board studies indicate that one of the greatest obstacles to the inclusion of autonomous mobile surrogates within the battlespace is the need for operator confidence. This confidence can be achieved only by enabling modes of operator involvement that allow humans to do what they are best at (e.g., formulating concepts, assigning missions, ranking priorities, determining requirements, allocating means, defining limits, monitoring status, etc.) while offloading the onerous burden of mentally tracking and processing vast data. Within the context of distributed robotics, the DARPA/INEEL project described in this report represents groundbreaking research in this area.

The monitoring and control of hundreds to thousands of mobile robots demands significant effort in terms of cognitive workload, specifically in the area of maintaining battle-space or situation awareness. Within the greater sphere of command and control, much research has been done to understand the need for situation awareness : "the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning and the projection of their status in the near future" (Endsley 1987). The operator must understand the robotics force not only in terms of where the robots are, but also in terms of what they are doing now and what they will likely do in the near future. Situation awareness is a critical element in decision making, especially in highly dynamic circumstances that are outside of normal operations. One specific goal of the command and control system should be to provide tools designed to minimize the operator's cognitive workload in developing and maintaining situation awareness of the battlespace. A loss of situation awareness will likely result in slower detection times, slower reaction times, and higher likelihood of decision errors as the operator struggles to re-orient with the operational parameters.

The envisioned robotics force can act autonomously and yet must be responsive to user-control at a variety of levels. Mission planning is the most important aspect of the successful deployment of a robotics force. However, once the force is deployed, the majority of operator interaction is devoted to monitoring its status and conducting minor changes to the original mission plan as the system operates autonomously. Research has indicated that in activities with high level of automation in which an operator serves mainly in a monitoring role, situation awareness may be negatively impacted. It has been hypothesized that this may result from: (a) a loss of vigilance as the operator assumes a monitoring role, (b) the shift from the operator being an active processor of information to that of a passive recipient, or (c) a loss or change in system feedback concerning the state of the system (Endsley and Kiris 1995). As the degree of automation increases, it becomes more difficult for the operator to understand the underlying state of the system. In the case of the robots, this lack of understanding is evident by the “Now why are they doing that?” response.

Gawron (1998) identified some of these problems with the deployment of uninhabited aerial vehicles (UAVs). Some of the relevant human-user interface problems identified are:

1. Data link drop-outs were not always apparent to the operator, and the UAVs traveled beyond the data link and control range of the operators.

2. Operators had trouble maintaining vigilance over long periods of time during UAV missions of 3.5 to 40 hours.

3. Humans could process imagery exploration on only a single data stream at a time, but some of the individual UAVs each collected two simultaneous data streams.

4. Operators had difficulty controlling vehicles when the systems experienced significant time delays in the control system.

A focus of this work has been to develop mechanisms that can alleviate these problems. Ideally, the system should have some capacity to monitor itself, detecting and reporting error, mortality, and task progress. Perhaps most importantly, the system should supply the user with meta-knowledge – an understanding of the relevance and certainty of what is known and reported by the system.

3.2 Necessary Command and Control Elements

Command and control for large numbers of autonomous robots represents a unique situation for a human operator. In some instances, it resembles an air traffic controller trying to monitor and coordinate the movements of a large number of aircraft. In other cases, the operator assumes a role much like that of a sonar operator on a submarine, who in monitoring a vast array of sensors is constantly trying to optimize the sonar system’s performance to identify that one piece of information in a vast ocean of noise.

The functional requirements for autonomous systems control were discussed at a 1998 national technical workshop sponsored by the Department of Energy (DOE) and the DOD. Figure 1 illustrates the roles / functions that a supervisor of an autonomous system must meet. It also indicates some of the necessary functional elements for command and control systems (DOE 1998).

Of these functional requirements, planning is the most critical, because of the autonomous nature of the robots. After deployment, however, most of an operator’s time will be spent in a monitoring mode.

3.2.1 Command and Control Considerations

The following list presents some of the prerequisite elements for such a command and control system. This list is not necessarily inclusive and may not address all situations.

1. Selectivity – The ability to select individual or groups of robots. The basis of selection may be for command issuance or for information retrieval. The ability to create abstractions such as groups is a key element for managing large numbers of autonomous robots. Systems without this capability reduce the human operator’s effectiveness at maintaining operational control.

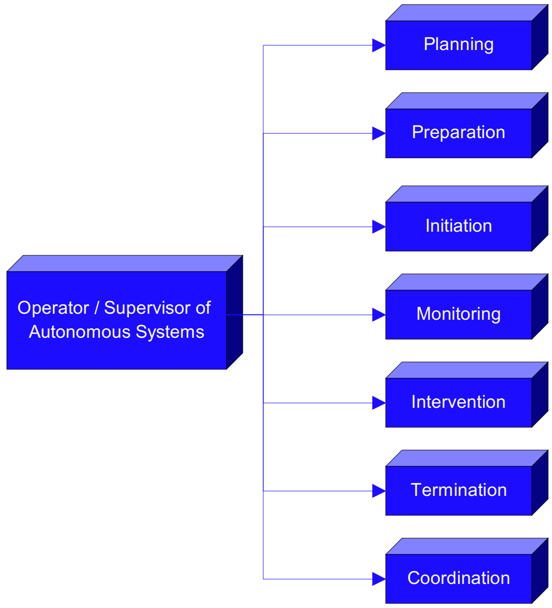

Figure 1. Multiple roles are necessary to supervise an autonomous system.

2. Command Issuance – A method for issuing commands or influencing the behavior of selected entities.

3. Acknowledgment – This reflects the ability of the command and control system and the robot to acknowledge that a command has been issued and received, respectively. This could be represented as a message exchange or perhaps even as a color change on the icon representing the entity of concern. In the absence of positive confirmation that a message has been received, the tendency is to re-transmit the message. This consumes resources in terms of both bandwidth and time, and additionally, diverts attention for more important tasking. "System designers must thus provide for easy and convenient ways to check on the status of messages without clogging message queues on either the sending or the receiving end" (National Research Council 1999, p.207).

4. Visualization – The operator must be able to visualize the environment as perceived by the robot and understand the actions of the collective. Visualization includes sensor fusion and multi-modal / multi-perspective representations of the robots, environment, and task. Visualization should not only provide sensory feedback, but also support an understanding of the behavioral principles behind the operation of the robots.

5. Arbitration – A mechanism must be available to arbitrate between competing priorities of the command and control system and the behavioral characteristics of the robot. Arbitration mediation is also needed in the case of shared control where multiple operators may control different functions within the same task force. An example is the case where a forward-deployed Army Ranger is tasked with ensuring that the robots are in position to analyze an area for the possible presence of weapons of mass destruction. Another operator at a command center may be in charge of sensor operation and the geometric configuration of the task force. It is conceivable that competing orders may be issued to the robots. Some mechanism must exist to resolve such conflicts.

6. Suspend / Activate – an ON / OFF mechanism to suspend and activate a robot. This mechanism would allow an operator to suspend selected robots and conserve power until the situation required activation.

7. Distress Call – The distress call represents a mechanism by which the "system" recognizes that it is faced with a previously unknown situation or a predefined situation that requires external assistance to resolve. The distress call mechanism can exists on two levels. One level is that of the individual robot having self-awareness of its current state and the environment. The second level is that of the command and control system examining the collective state and (for example) identifying areas of previously undefined or unexpected activity among multiple robots. On an individual level, the robots may not sense a need for assistance, but their collective actions may dictate the need for user intervention.

8. System Status Report – This represents a mechanism for reporting a robot's state or health status. This may constitute a simple display of the robot's state variables, or it may involve an abstract quality for fitness or health based upon a self-test. The quality of health could be termed as "excellent", "good", "fair", etc., the assessment being based either on a preprogram or on operator designated criteria.

9. Task Awareness – Operators must receive feedback not only on the robots' health, but also on their ongoing task performance. This information should be relayed at several levels. Using the visualization feedback, the operator might be able to intuit certain measures of success. In addition, progress can be reported empirically through measures such as percent of task completed, estimated time-until-completion, ground coverage, etc. Such metrics can prove useful for optimizing collective behavior either manually or through machine learning techniques that exploit an automated critic. Task awareness might also include measures of confidence regarding this information. Mine field penetration and marking is a mission often linked to future micro-robotic forces. In this case the command and control system operator, based upon feedback from the task force, must assess how successful the system has been at locating and marking mines. The operator must confidently determine whether the area is sufficiently marked to allow for safe personnel and equipment passage.

The command and control system elements listed above are designed to assist the operator or commander in decision making and task execution. The command structure should leverage the tactical commander, through whom intelligence, experience, and intuition might direct action beyond that dictated by the simple situational awareness of individuals or the social collective. Additionally, the command and control system must direct thousands of independent robots where individual robot control is beyond the capabilities of a single operator.

3.2.2 Tactical Display Considerations

As previously stated, research in the area of command and control for large scale autonomous robotic forces has been limited. The design guidelines for tactical displays, however, are still applicable and provide a basis for system attributes. A recent study by the National Research Council on tactical displays for soldiers listed the following as important design considerations:

1. The tactical display should minimize the cognitive workload placed on the operator. This is accomplished through:

    - providing integrated information (e.g. fusing information from different sources)

- providing easy user input of information (e.g., menus)
- minimizing memory requirements
- reducing extraneous information
- simplifying the format of the information presented
- minimizing operator tasks
- presenting information in a task-oriented sequence and grouping
- providing information in the needed format (e.g., egocentric maps).

2. The display should enhance situation awareness by providing salient cueing, directing attention to the most important information.

3. The display should minimize complexity and avoid high levels of automation.

4. The system should provide new capabilities needed by the soldier, such as integrating information (as needed for decision making), comparing information to pertinent goal states, allowing a projection of future states, and providing support for human memory.

5. The display design should allow for easy sharing of information between team members and between field and headquarters.

6. Continuing from the example above, the command and control elements necessary for successful deployment of autonomous micro-robots mirror the process by which a sergeant issues an order to a soldier or group of soldiers. This analogy is very appropriate, since the micro-robots constitute intelligent (albeit artificial) autonomous entities and are not merely a remotely controlled extension of the human operator (National Research Council 1999).

4. INEEL COMMAND AND CONTROL ARCHITECTURE DEVELOPMENT

4.1 The Hierarchical Model of Command and Control

A hierarchical system of command and control was selected as the first model for evaluation. The hierarchical system consists of an organizational structure with various levels of control between units within the organization. A “chain of command” exists within the organization, which dictates the relationship between levels units. The military is the most common example of a hierarchical structure. In this case, the individual soldiers constitute the base element of the structure. Soldiers are grouped into units, each of which possesses a unit leader; multiple units constitute a squad, etc. Command and control systems using a hierarchical framework reduce system complexity by allowing the user to interface with an individual soldier at a high level of abstraction. Command and control objectives are achieved along organizational lines and not by a one to one correspondence. Two methods exist for implementation of a hierarchical-based structure. The first involves the specific designation of group membership for individuals. The second involves designation and control through a leader around which a following develops.

4.1.1 Group Abstractions

Hierarchical structures can be developed through operator designation of groups. Here the operator designates the hierarchy by explicitly choosing group membership. The operator may make this choice based on proximity or based upon functionality. The operator may want to establish a certain functional capability among a non-homogenous collection of robots. The key is that the operator specifies group membership.

This type of hierarchical structure has strong roots in biological societies that exhibit a dominance ordering among members. This dominance order consists of a set of sustained aggressive-submissive behaviors among members of the society. In the simplest form this consists of rule by one individual, despotism. In many cases, however, it consists of a hierarchy among members with rank distinction. Here an alpha individual dominates the society; a beta dominates all but the alpha, down to the omega at the bottom of the line. In these societies, dominance is normally based on size, strength, and aggressiveness (Wilson 2000). Recently, a study at the University of Southern California showed that a group of robots could efficiently arbitrate task conflicts by enacting pre-wired aggression behaviors (Vaughan et al. 2000).

Using *AgentSim* (described in more detail in Section 4.2), the user can specify a hierarchy between individuals and/or groups. As an example, an operator deploying 500 robots over a field designates two groups. Group Alpha contains all the robots on the north half of the field. Group Bravo contains all the robots on the south half of the field. The user can now interface with the robotic force on a group level of abstraction, (i.e. Alpha or Bravo). User interface with individual robots is still possible, but not necessary for high-level commands.

In the same way, groups may be organized into collections called units. In this example, the user may designate groups Alpha and Bravo to be members of a unit called Field1. Commands to a unit will flow down to groups, which will flow down to individuals. This structure may be visualized as a branching tree diagram. Each node represents a level of abstraction. The links between the vertices represent the relationships between the abstractions.

4.1.2 Group Leadership

The second method of implementing a hierarchical structure is through dynamic group formation. In this model, the human user does not select the group, but rather selects individual leaders from among the masses. Commands are issued to these leaders who in turn invoke a following among the collective. Group membership is not predefined, but is a function of the "charisma" of the leader.

Leadership in the animal kingdom commonly refers to the simple act of physically leading other group members during movement from one location to another. In this case, the movement leader is not necessarily the dominant member of the group. The leader of the group may change as circumstances warrant such as the discovery of a predatory threat (Wilson 2000). This type of control has applications when combined with a subsumption-type layering of behavior. In this regard, a leader may attract a following of "unemployed" robots. Other robots engaged in meaningful activity would not be compelled to follow the leader. Additionally, followers could break off from the group if stimulated to perform a higher level action.

Once a leader is designated, the question is how to communicate and instill group action? Nature again provides some interesting examples. Birds commonly use a combination of body gestures and audible sounds to signal intent. The honeybee does the waggle dance to indicate the direction and distance to a target. Another form of leadership found in honeybees produces an autocatalytic reaction. This form of action initiation is called the buzzing run, the breaking dance, or *Schwirrlauf*, which honeybees use to induce swarming. In this method "...one or several bees begin to force their way through the throngs with great excitement, running in a zigzag pattern, butting into other workers, and vibrating their abdomens and wings ..."(Wilson 2000, p.213). This action incites other worker bees to perform in the same manner and soon most of the collective is affected. After about 10 minutes, the bees nearest the opening depart and the frenzied collective follows.

The ways by which a leader can influence the collective can therefore be grouped into four styles.

1. Leadership by example – The imitation of the leader's actions by the collective, i.e. follow the leader.

2. Tasking by explicit order – The issuance of direct communication from the leader to subordinates to signify action.

3. Tasking by a preprogrammed response to a leader's or other member's actions – Not necessarily an imitation behavior, but a response to actions by the leader, so as to produce a cascading effect.

4. Any combination of the above.

4.2 Implementation of Command and Control in *AgentSim*

AgentSim is a simulation environment developed by the INEEL. It allows for the evaluation of individual and collective behaviors for autonomous robots. Additionally, it provides a test interface for the development and evaluation of command and control concepts for use with large numbers of autonomous robots. The simulation package was originally developed as part of an internal research project at the INEEL to evaluate individual and collective behaviors for large robotic forces. It was developed using Microsoft Visual C++ on a *Wintel* machine. A window platform was chosen for development to facilitate the distribution of code among potential users. The Windows family of operating systems is the DOD and DOE company standard. No additional software or hardware is required to run the program on a “company standard” machine.

4.2.1 Development Scenario

Rather than design an all-encompassing generic command and control system, this project focuses on a particular class of applications scenario: the use of a large number of mini-micro-robots to characterize a potentially hazardous area. For example, a train with several cars containing hazardous material has derailed. A subsequent explosion has scattered debris around the crash site. The extent of the area of contamination and the status of the cars carrying the hazardous material are not known. Due to the dangerous nature of the scattered material, a task force of micro-robots is selected for site evaluation before human entry into the area. An uninhabited aerial vehicle (UAV) drops a group of 2000 micro-robots equipped with chemical sensors over the debris field and in the vicinity of the overturned cars. As the robotics coordinator, your mission is to use the micro-robots to map the contamination levels in the area and assess the hazards before humans enter.

This scenario represents a practical and seemingly simple application of micro-robots. However, it reveals many of the technical challenges involved in deploying large numbers of micro-robots. These challenges include the following questions.

- Once the robots have deployed, how do you direct the masses in coordinated motion in the direction of interest?

- During the formation and sensor sweep, how do you identify and then adjust for the inevitable “deaths” (unit failures) in order to ensure complete coverage?

- How do you assess the extent to which the area has been fully searched? Do coverage gaps exist?

- At what point do you have enough confidence in the sensory feedback to send in a human response team?

4.2.2 Hierarchical Structure Design Implementation

The hierarchical command and control structure provides commanders a means to communicate, task, and restructure resources without interaction at an individual robot level. Hierarchical command and control in *AgentSim* is accomplished primarily through the use of group abstraction. It provides the operator the capability for “on-the-fly,” ad hoc designation and control of groups of robots. The system provides three levels of control. The first level of control consists of an interface with individual robots. This level is necessary to permit the operator to control or evaluate the state of individual robots. The next control level is that of a group. Groups are designated and modifiable by the operator. This provides a level of abstraction for interface and control. Groups consists of a collection of operator designated robots. The operator has the ability to create and modify the content of groups. The highest level of control and abstraction is that of the unit. A unit is similar in structure to a group. Its membership can consist of groups and/or other units. Again, the operator designates membership. Figure 2 illustrates this command relationship.

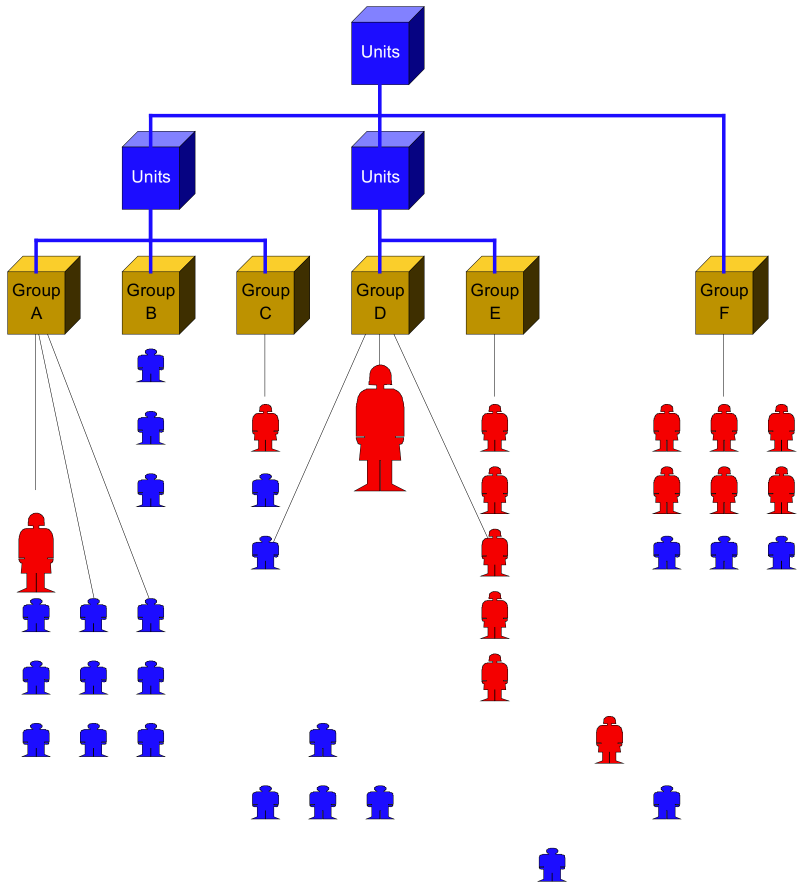

Figure 2. Flexibility is built into the *AgentSim* command relationship.

Groups are operator-designated and may be formed by creating a specific functional mix among a heterogeneous collection of robots, or by selecting all the robots in a specific area. As illustrated in Figure 2 (Groups D and E), one individual may be a member of several groups. An example may be a group of robots, all of which are equipped with sonar beacons. At the same time these robots may individually be dispersed in other groups based upon geographical location. Group control facilitates the simultaneous activation of sonar beacons and also facilitates motion control. Unit formulation allows the operator to designate forces on the basis of the functional requirements of specific mission tasking.

The designation of an organizational structure within the collective also promotes the ability to design layers of access control among multiple operators. Assume three operators are given distributed control over a collective of 5000 robots. Shared control among the operators can be allocated via group and unit assignment. Operator 1 can be assigned control over Unit 1 consisting of Groups A through C. Operator 2 is assigned control of Unit 2 with Groups D and E. Operator 3 is the overall coordinator and has control over Unit 3, which encompasses both Units 1 and 2.

In the implementation of *AgentSim*, lower-level commands subsume higher-level commands. A command to a Group will supercede the Unit command. Likewise a command to an individual robot will supercede a previous command from the Group to which it is a member. This represents only one method for control interaction between the command levels. Other methods for arbitration and conflict resolution should be evaluated.

4.2.3 Prototype System Development

A rapid prototype of a hierarchical command and control structure was developed within the *AgentSim* simulation environment. The objective was to refine command and control requirements and explore possible mechanisms for meeting those requirements. Programming was conducted using Visual C++. The user interface is based on standard Windows-type displays and controls (i.e., a menu-driven mouse interface). Pictures showing the interface are presented in Appendix B. Figure 3 shows the basic window, with what represents a batch distribution of a sizeable number of robots (348). Each robot is constructed as an autonomous software object with a unique system-state and behavior set. Simulation currently drives the system variable, but work is in process to tie the system-state to the output of actual robotic platforms.

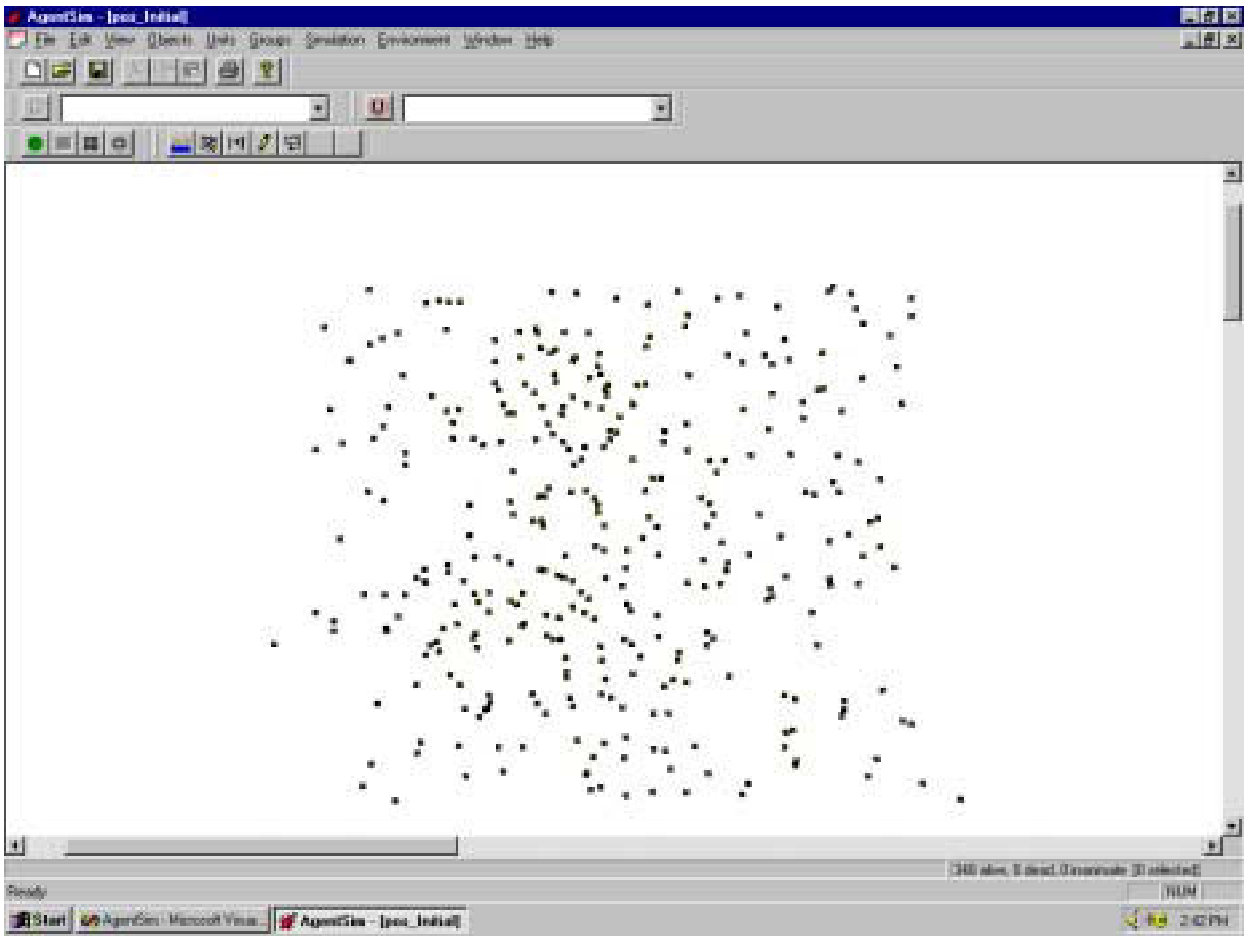

Figure 3. Basic *AgentSim* interface window.

4.2.3.1 Individual Level Control. Figure 4 illustrates the ability to query and view individual robot states. In this case robot 287 and robot 340 are in view. The robot window displays the system states, including robot type, position, heading, wander direction, and state. The time of death is part of the simulation, based on an exponential life expectancy function. The operator is presented two sets of robot controls. The first type of control does not affect the robot, but provides a means for the operator to view and organize robot information. Two separate controls of this type exist:

1. A control that allows the operator to remove the robot information from the viewing screen. This allows the operator to customize a view for focus on a certain area or task while minimizing distracting information.

2. A control that traces the movement path of the robot, to assist the operator in understanding behavior and also to track robot coverage.

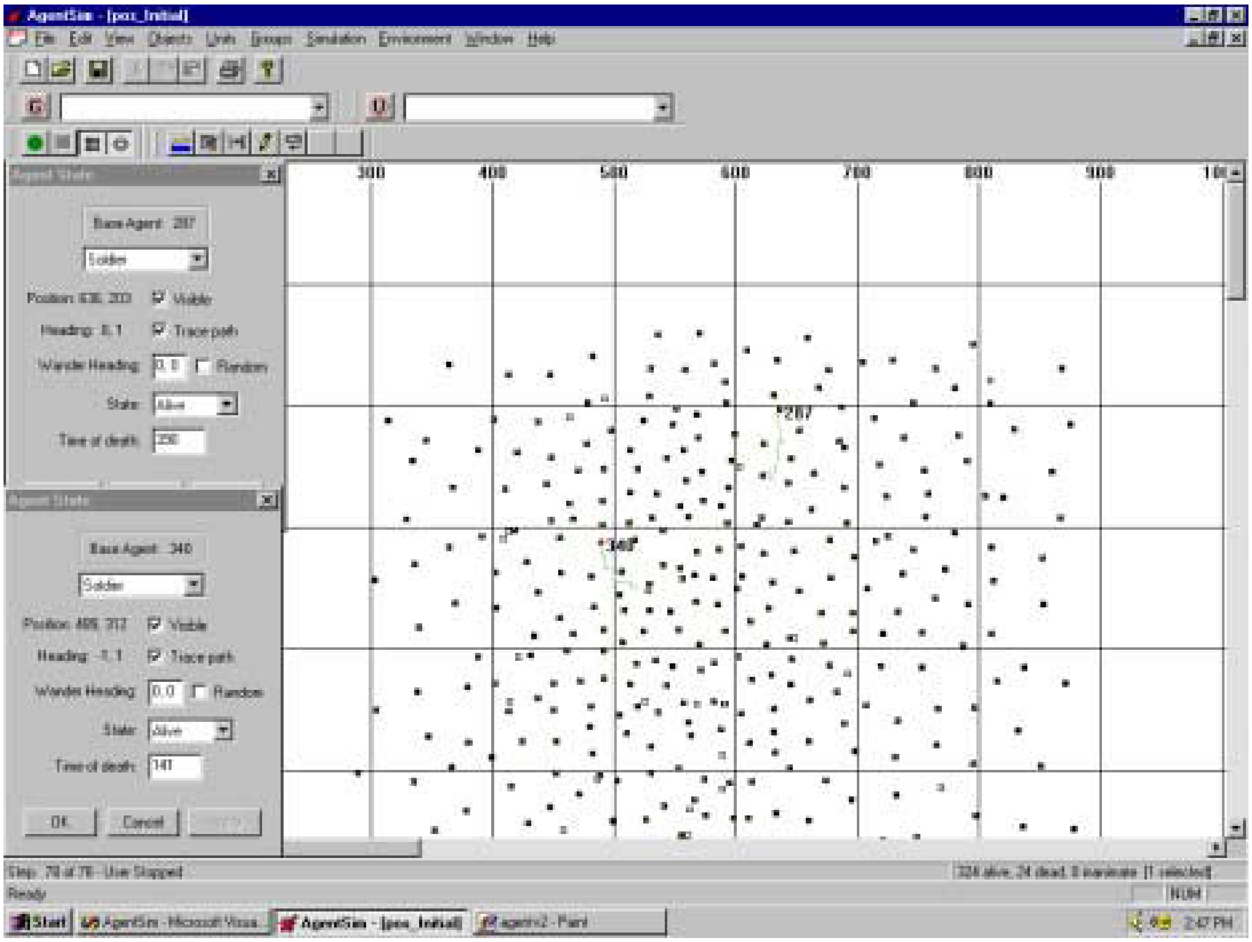

Figure 4. Query and control of an individual robot (individual level).

The second type of control consists of commands to the robot for behavior modification. These include:

1. Robot type selection. As stated earlier, hierarchical structures can be implemented in multiple ways. One method is by operator grouping; another is by leadership designation. This control allows the operator to designate a robot as a soldier or as a commander. The change of state to a commander modifies the way neighboring robots react. In this case, a commander imparts a greater attractive force than that exerted by a soldier. Thus a commander is able collect a following of robots.

2. Wander heading. This displays the robot's desired heading subject to the effects of external stimuli. This may be predetermined or a random function.

3. Robot state. This allows the operator to suspend or activate a robot.

Figure 4 also illustrates two visual aids designed to enhance the operator's situation awareness and understanding. These include an adjustable grid overlay to provide the operator a geospatial sense of individual and collective movements. Another feature is the display of "dead" robots. In this case a dead robot represents a robot known to been terminated, or with which communication is lost and cannot be re-established. Each of the open squares on the display represents a "dead" robot. The operator may wish to investigate or avoid an area with a high mortality rate. Although this system emphasizes the need for appropriate user input, the system is designed to support a variety of machine learning approaches that can permit autonomous adaptation. For instance, co-evolutionary learning capabilities could permit members of a group to self-adjust their behavior, responding online to significant events such as catastrophic loss of members or physical areas of high mortality.

4.2.3.2 Group Level Control. The level of abstraction immediately above individual control is group control. The interface for Group control and designation is illustrated in Figure 5. Groups permit the designation, selection, tasking, and re-tasking of multiple robots by a single operator. Group membership is identified by a colored ring surrounding the robot. The operator uses the mouse to designate membership in a group. Selection is made individually or by circling a collection of robots. Group controls are similar to those for individual robots and include display (visualization) and operational controls.

Visualization controls allow the operator to custom-configure the information presentation to best fit his/her needs. These controls include:

1. Group name. The operator can give the group a meaningful name beyond the default name.

2. Group radius and color. The operator can modify the appearance of the group by specifying the ring color and size.

3. Group visibility. Additionally, the operator can remove a group representation from the viewing screen. This permits the operator to focus attention on pertinent information while temporarily removing possibly distracting data from the display.

Behavior modification controls include:

1. Group formation behavior. The ability to enable or disable group formation behavior among group members. Removal of this behavior removes the potential field effect between neighboring robots. Instead of being influenced by an adjacent robot, individual motion is along the goal heading, unless otherwise modified.

2. Goal heading designation. A spin wheel allows the user to specify the group's goal heading by selecting a direction arrow. The geospacial alignment is North at the top of screen.

Note that robots can be members of multiple groups simultaneously. This feature promotes flexibility for groups of multiple functionality and also for distributed control among multiple operators.

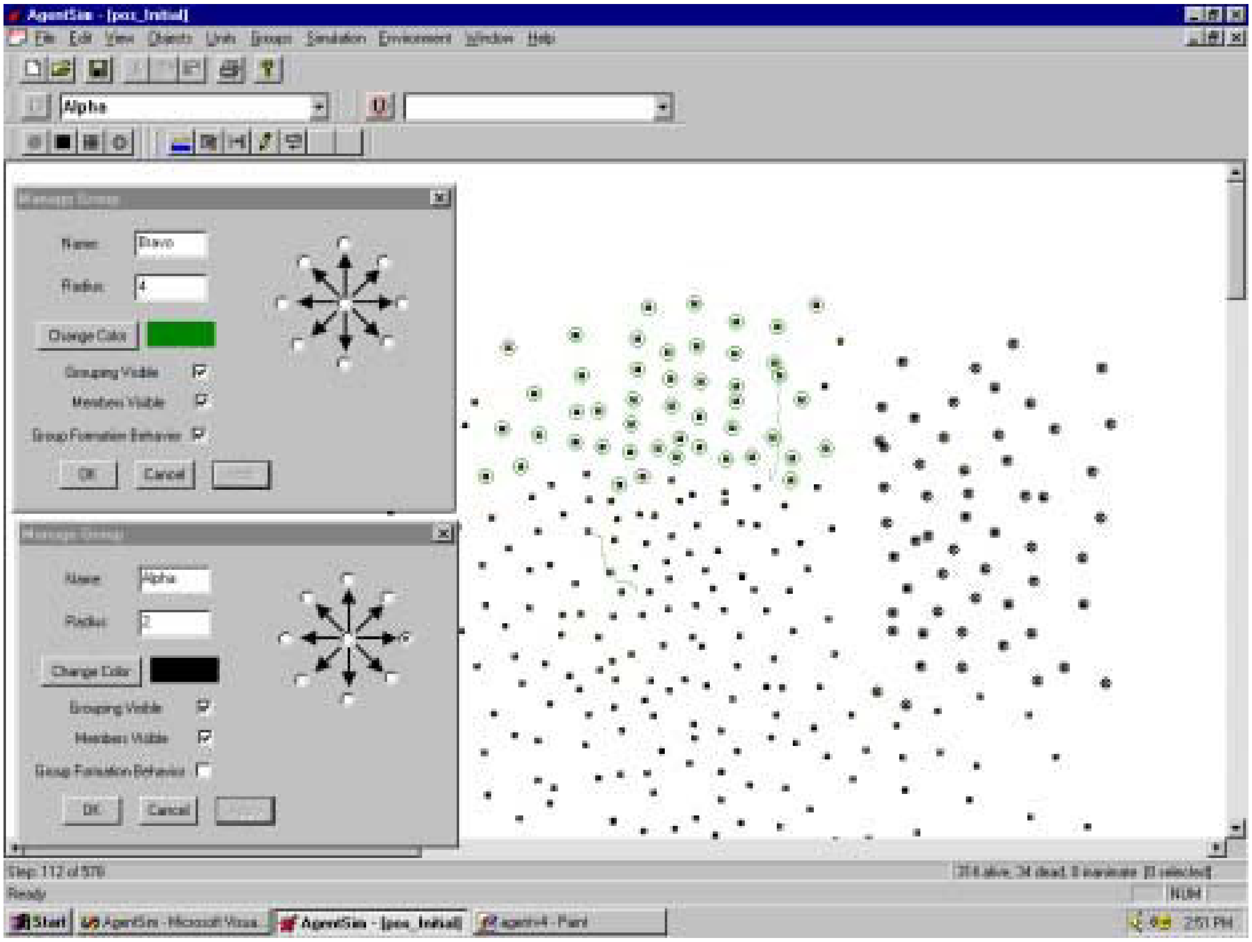

Figure 5. Control at the group level.

4.2.3.3 Unit Level Control. Unit designation represents the highest level of control abstraction. The interface for Unit control and designation is illustrated in Figure 6. This level of abstraction gives commanders flexibility in designating and re-assigning assets to meet specific mission requirements. Membership can consist of groups and/or other units. The display and operational controls are similar to those for group and individual control and include:

1. Unit name
2. Unit color
3. Unit visibility.

Behavior modification controls include:

1. Unit formation behavior; the ability to enable or disable this behavior among Unit members
2. Goal heading designation.

Where as groups may be based on a collection of robots with similar capability, the Unit may be designed to contain a specific capability mix of groups to support specific mission profiles. Within the interface it is easy to dynamically create and modify the Unit's composition.

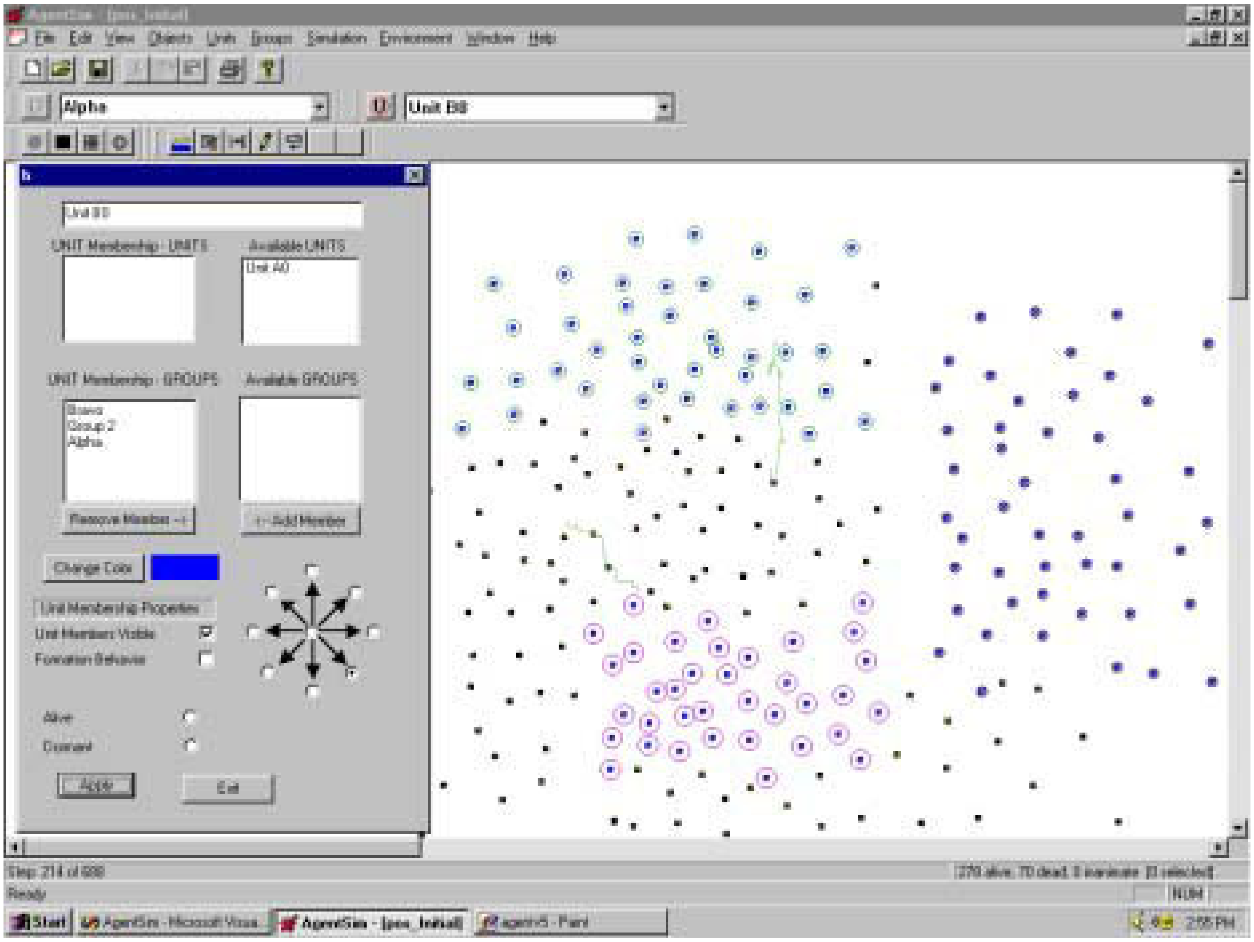

Figure 6. Control at the unit level.

4.3 Initial Findings

The objective of this joint INEEL and DARPA project has been to develop and evaluate human robot interface technologies that will permit the human to interact with the robots collectively rather than requiring the human operator to interact with each and every individual robot. Specific attention has been given to studying command and control attributes and structures necessary to make this human operator interaction possible. This project has conducted an initial investigation into a hierarchical type control structure to facilitate one-to-many, human-to-robot interactions. In fact, this project represents some of the initial research in this area.

The goal to date has not been to develop a full-scale command and control system, but to develop and model some initial concepts using a rapid prototype system. The hierarchical control model as demonstrated in *AgentSim*, while not intended to provide full functionality, has proven very beneficial in exploring command and control issues for large-scale robotic forces. The ability to dynamically designate, task, and re-task robots through a group level abstraction provides great flexibility to an operator in managing large numbers of robots. A key element here is the dynamic nature of this resource allocation. Very rarely does a mission plan execute as expected. This command and control system supports rapid "on the fly" modifications to accommodate changing environmental conditions and changing mission priorities. Through levels of abstraction and various visualization aids, this system also supports the commander's ability to understand and project individual and collective behavior of the robots.

5. PATH FORWARD

This first year's research has provided insight into issues related to both visualization and control. The results so far show promise in enhancing command and control support. We anticipate addressing these issues in greater depth, along with other issues, as the project continues in subsequent years.

5.1 Further Research

Areas for further research include:

1. Advanced visualization methods for representing collective / individual robot states. The objective is to enhance operator situation awareness and provide a greater understanding of the entire system, while at the same time reducing the operator's cognitive workload.

    a. Iconic representation of group and units. Instead of displaying individual robots, the operator could display groups and/or units as individual icons. The icon might display information such as the number of robots within the group. The operator could then shift between an expanded group view of the individual robots and an iconic representation of the collective.

    b. Area of uncertainty bounds for collectives / groups. Given possibly covert missions, limited bandwidth, and communications limitations, it is unlikely that constant contact will always be maintained with individual robots. In the absence of precise location or sensor coverage of the robots, an area of uncertainty could bound the most probable location and coverage area. The area of uncertainty would grow over time until positive communication is established, at which time the area of uncertainty would shrink. Such a display tool would assist the operator in assessing robot placement and the periodicity requirements for communications and data links.

    c. Density graphs (robot dispersing, sensor coverage). Instead of relying on a naked eye perception of area coverage, a density graph using a color or shading scheme would allow the operator to quickly identify areas of low sensor coverage.

    d. Containment graphs representing boundary strength. When containment and not area density is the issue, a containment graph representing the bounding area covered by the robots provides the operator a visual aid in assessing boundary strength.

    e. 3D visualization. While 2D visualization is often sufficient for ground-based vehicles, 3D visualization is necessary for water and air-based platforms. Additionally, 3D visualization would enhance an operator's understanding in situations where ground-based units are coordinating with water / air-based units or if ground-based units are operating at multiple levels, such as in surveying a multi-story building.

2. Coordinated Control and Enhanced Planning.

    a. Event scripting capability. Scripting provides the capability to sequence and track coordinated actions between teams of robots and also between the robots and other resources. Time-lining the events also provides a means to assess progress and identify potential problems.

b. Seamless transition from predictive (simulation) and operational modes. A simulation mode based upon constantly updated real world parameters would greatly assist the operator in assessing future action and responses. To further smooth the transition between simulation and real-world, machine learning techniques can be used to perform online parameter adjustment.

c. Multi-user domain with level-of-control assignment (distributed command). In the age of network centric warfare, distributed command will be an essential element of any emerging command and control system. Multiple operators must be able to simultaneously access and control the same collection of assets. This type of control will require arbitration and collaboration schemes to coordinate the actions of multiple users.

d. Distributed simulation merging real and simulated assets (web-based). Pre-mission planning and execution training constitutes probably the most important aspect of large-scale autonomous force deployment. It will be highly critical to conduct remote training of potential distributed forces before commencing a mission. Distributed simulation will be a vital aspect of this training.

e. Currently, *AgentSim* models robot death using an exponential life expectancy function. Further work will give the system the ability to model catastrophic failure where whole groups or large percentages of groups are unexpectedly taken offline. This could be due to a loss of communication or robot death. How should the system respond to such losses? Templates could provide the system with the characteristics of each robot, allowing the system to report on the capabilities lost and the resulting impact on the task.

3. Integration of actual robot control into the current program. This project has concentrated on the development of command and control concepts and has conducted an initial simulation-based evaluation. The next step for evaluation is the integration and control of actual robotic platforms. The INEEL plans to pursue this effort using a team of 10 to 15 Growbot™ robots by Parallax, Inc. Figure 7 shows part of our team.

As INEEL implements *AgentSim* on small-scale robots, efforts continue to face two fundamental problems:

a. The need for accurate, reliable positioning

b. The need for efficient robot deployment.

To address these problems, INEEL is working to develop robot templates that permit *AgentSim* to orchestrate heterogeneous robots. *AgentSim* already includes the capability to specify and task a command robot. Current research is investigating the possibility that a physical parent robot can augment the capabilities of a swarm, deploy the smaller robots, provide positioning support, and facilitate command. In addition to GrowBots, the implementation of additional robot templates will enable *AgentSim* to be used with other platforms developed or used under the auspices of the DR/SDR programs. The goal is to show that *AgentSim* can be used for command and control across a variety of distributed robots and applications. Rather than increasing the complexity of the system, the use of templates within *AgentSim* can render heterogeneity transparent to the operator.

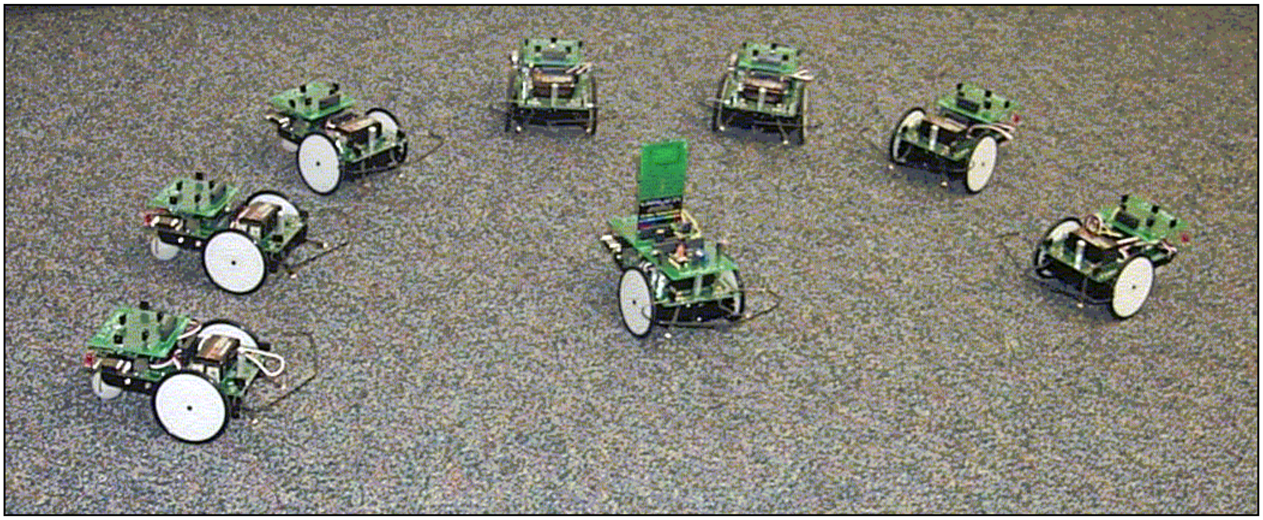

Figure 7. A few of the robots with which INEEL will integrate command and control.

4. Evaluation of additional command and control structures. The hierarchical structure represents only one of several possible command and control structures. Others exist, for example, air traffic control models. An evaluation of additional models is needed. Mission requirements will dictate the exact nature of the needed command and control system.

5.2 Project Collaboration

This project represents the SDR's sole high-level command and control research project for large-scale forces of autonomous robots. One of the over-arching goals of this project is the integration and development of a command and control system to support the deployment of robotic platforms developed by other SDR and DR programs at DARPA. Discussions are currently in progress with Dr. Jason Janet of Nekton Technologies, Inc. in exploring applications and command and control requirements in the support of submarine-based applications for the MicroHunter™ platform.

The University of Minnesota's Scout and Ranger system is another DR project with potential for collaboration. Like the MicroHunter™, this system is close to field deployment, but ultimate integration into mission scenarios will depend on a robust command and control system suitable for a nominal operator's use.

The insights gained and the command structures developed in this project are also planned for integration into remote surveillance and monitoring systems in support of the Long Term Environmental Stewardship mission at the INEEL.

5.3 Technology Transition

The technologies and insight developed with this project are being used in conjunction with internal research conducted by the INEEL for developing cornerstone capabilities in cooperative robotics and application venues in support of the DOE's Robotics and Intelligent Machine initiative (DOE 1998). The control concepts and human interaction requirements are applicable to a diverse force of larger autonomous robots, which are also of interest to other government agencies.

6. PUBLICATIONS

The following conference papers and presentations represent additional and ongoing aspects of this project.

Dudenhoeffer, D. D., Jones, M. P. "A Formation Behavior For Large-Scale Micro-Robot Force Deployment," *Proceedings of the 2000 Winter Simulation Conference 2000*, (WSC '00), Orlando, FL, Dec 5-8, 2000.

Dudenhoeffer, D. D., McKay, M. D., Anderson, M. O., Bruemmer, D, "Development and implementation of large-scale microrobotic forces using formation behaviors", *Proceeding of SPIE Vol. 4364*, Orlando, FL April 16-20, 2001.

Hallbert, B. P., D. D. Dudenhoeffer, D. J. Bruemmer, M. L. Davis, and G. J. Khoury. 2001. Human Interface Concepts for Autonomous/Distributed Robot Control, INEEL/EXT-2001-00232, Idaho National Engineering and Environmental Laboratory, Idaho Falls, ID.

7. REFERENCES

Arkin, Ronald C., 1998, *Behavior-Based Robotics*. Cambridge, Massachusetts: The MIT Press.

Barbato, Greg, 2000, "Uninhabited Combat Air Vehicle Controls and Displays for Suppression of Enemy Air Defenses", *CSERIAC Gateway*, XI (1): 1-4.

Department of Defense, "DOD Dictionary of Military Terms", http://www.dtic.mil/doctrine/jel/doddict/.

Department of Energy, 1998*, Guidance Document and National Technical Workshop Report: Final Report – Guidelines for the Development of Training Programs for the Operation and Maintenance of Robotic Equipment*, DOE Office of Science and Technology.

Endsley, M.R., 1987, *SAGAT: A Methodology for the Measurement of Situation Awareness* (NOR DOC 87-83). Hawthorne, CA: Northrop Corp.

Endsley, M.R. and E.O. Kiris, 1995, "The out-of-the-loop performance problem and level of control in automation," *Human Factors* 37(2): 381-394.

Gage, Douglas W., 1992, "Command and Control for Many-Robot Systems," In *Proceedings of AUVS-92*. Reprinted in *Unmanned Systems* 10 (4): 28-34.

Gawron, V.J., 1998, "Human Factors Problems Associated with Uninhabited Aerial Vehicles (UAVs)," *Proceeding of the Human Factors and Ergonomics Society 42nd Annual Meeting*, 1998, p. 1600.

Lee, J., 2000, "Cognitive engineering challenges of managing swarms of self-organizing agent-based automation," *Proceedings of the IEA 2000/HFES 2000 Congress:* 1.568-1.571

Morgenthaler, Matthew, Alan Dickinson, and Betty Glass, 2000, "XUV/Demo III Multi-Vehicle Operator Control Unit", *Proceedings of SPIE, Unmanned Ground Vehicle Technology II*, 4024: 148-158.

National Research Council, 1997, *Tactical Display for Soldiers: Human Factors Considerations*, National Academy Press, Washington D.C.

National Research Council, 1999, *Realizing the Potential of C4I: Fundamental Challenges*, National Academy Press, Washington D.C.

Vaughan, R., K. Stoy, G. Sukhatme, M. Mataric, 2000, "Go Ahead Make My Day: Robot Conflict Resolution by Aggressive Competition*," Proceedings of Simulation of Adaptive Behavior Conferenc, 2000*, p. 491.

Wilson, Edward O., 2000, *Sociobiology: The New Synthesis (Twenty-Fifth Anniversary Edition)*, Cambridge, Massachusetts: The Belknap Press of Harvard University Press.

Appendix A

Basic Agent Model

Basic Agent Model

The model of an autonomous micro-robot is constructed by building upon a base autonomous agent object. The basic model of the agent can be thought of as simply a physical shell. In abstract programming terms it may also be thought of as an object with general capabilities. The basic agent possesses only locomotion as an innate capability. The agent exists in one of three states: dead, alive, or dormant. The only core capability possessed by the agent is motion, which is further restricted by speed and endurance limitations. We make a distinction between the agent's motion capability and a behavior designed to direct or use that capability of motion.

This basic agent serves as the platform on which additional capabilities (i.e., sensors) and individual behaviors are layered. Sensors are added to the agent model by "plugging in" sensor models. The sensors query the environment model to perceive objects or conditions of interest. The agent receives input from these sensors to increase its basic capabilities.

Similarly, new behaviors may be added to take advantage of additional sensory capability. However, it should be noted that sensors and behaviors are not the same thing, nor is it necessary to have a one-to-one correspondence between sensors and behaviors. Sensors provide a means for perceiving environmental states or conditions, while behaviors are the actions the agent takes based on the perceptions it makes.

Behaviors may rely on multiple sensory input (stimuli). For example, a robot's next move may be based on the input it receives from multiple neighbor detection sensors. Similarly an individual stimulus is not necessarily unique to one behavior. Neighbor position information may be used in both a group formation behavior and a collision avoidance behavior. When behaviors conflict or compete for resources, an arbitrating mechanism usually dictates the agent's reaction. In this way behaviors are layered.

Sensor Model

Sensors are modeled as encapsulated object classes. The agent uses a fixed set of input and output parameters to communicate with each sensor. Consequently, multiple types and qualities of sensors may be evaluated with complete transparency to the agent model. The agents presented in this model possess two types of sensors, a Nearest Neighbor sensor and an Object Detection sensor.

This project distinguishes itself from much of the past research in this area by the attention dedicated to modeling realistic sensor capabilities. The premise behind agent interaction is that one agent can "see" his neighbor. The ability to detect and identify neighboring agents cannot be taken for granted. Adjacent agents can be identified via two methods. The first method consists of an active broadcast in which agents broadcast position information. Neighbor position may be derived from a relative coordinate system or by strength and direction of the signal. Omni-directional position data are possible.

The second method involves passive detection without open communication between agents. Neighboring agents are detected through passive sensors. Infrared sensors are an example of this type of sensor. Sensor coverage is directly tied to the number and arrangement of sensors. Detection is further dependent on the sensors' accuracy.

The agent model uses passive detection. Each agent possesses an array of five sensors for detecting neighboring agents. Each sensor has a coverage spread of 45 degrees. Figure A-1 illustrates the sensor configuration used in the model. Three additional sensors could have been added for complete 360-degree

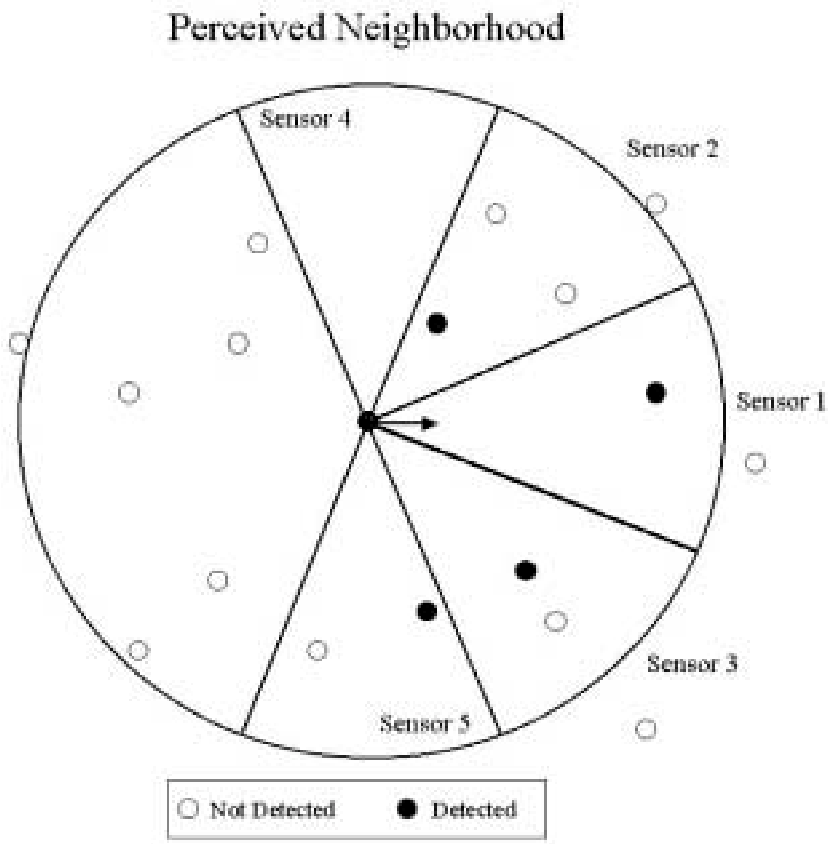

Figure A-1. Perceived robot neighborhood.

coverage. The decision not to add those sensors was based on the need to conserve the resources that would have been consumed by each additional sensor. These resources include power consumption, physical space, and computing (CPU) time. For this study's purposes, 360-degree coverage was not necessary, because an agent does not care who is behind it.

The sensors detect the nearest agent within the sensors' coverage area. The sensor returns the relative bearing, range, and type of the neighbor agent detected. Neighbor type is important because neighbor type will determine the agent's reaction to the detection. Figure A-1 represents the neighborhood of the agent. The perceived neighborhood, represented by the black dots, consists of only those neighbors correctly detected by the agents. Note that only the nearest neighbor is detected if multiple neighbors exist within the same sector.

Two types of errors are modeled for each sensor, an inherent offset error and a detection error. The offset error accounts for the imperfect angular alignment of a sensor with respect to the intended relative positioning. This error is constant. The detection error represents the imperfection of the sensor and the degradation of the detection probability as a function of the detection range. The detection function is based on an exponential distribution with a mean detection range of forty inches.

The Object Detection sensor determines whether an obstruction exists along the intended path of the agent. Specifically, an obstruction is detected only if it is immediately in front of the agent. This sensor returns a signal indicating an obstacle was detected. Within the model, no errors are associated with this sensor.

Behavior Model

The most important element of agent construction is the behavior set. In developing concepts and models for individual agent and collective behavior schemas, we examined biological entities and examples from nature for insight. Important to this research project were the relationships that birds and fish exhibit in flocking and schooling behaviors. Birds and fish have the ability to form and maintain collective patterns. These patterns are formed by the animals' ability to balance the desire to remain close to the flock (or school) and the desire to avoid collision (Shaw 1975). Within the flock, the bird does not possess universal knowledge (i.e., knowledge of the positions of all other birds in the flock), but it adjusts its position based on the perception of the locations its immediate neighbors. Reynolds used this framework to develop his ground-breaking animation work on Boids (Reynolds 1987). These two principles of flocking and local perception provide the basis for the development of the agent's behavior.

Subsumption Architecture

Once a set of individual behaviors has been developed, a framework or architecture must be constructed to initiate behavioral responses and coordinate multiple behaviors. The subsumption architecture (Brooks 1986) provides the basis for behavioral coordination within the micro-robot agent model.

In simplistic terms, the subsumption architecture is based on layering reactive behavior sets on top of each other. These behaviors concurrently react to the perceived environment. A key tenet is that reaction is based on perception and not on planning. Coordination among behaviors involves a hierarchical scheme where higher-level behaviors suppress or inhibit lower-level behaviors. In this same way, successively more complex behaviors can seamlessly be layered onto the existing behavior set. (Arkin 1998)

Figure A-2 illustrates the micro-robot agent's behaviors in order of their priority. The priority goes from Collision Avoidance (highest) to Wandering (lowest).

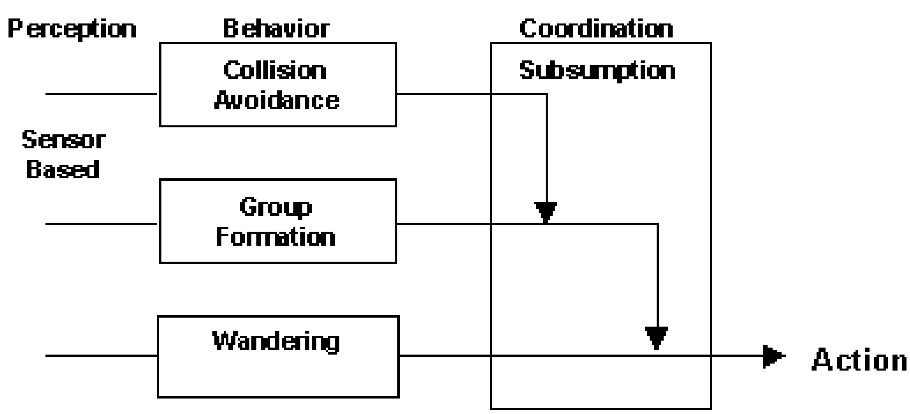

Figure A-2. Behavior architecture.

Wandering Behavior

The Wandering Behavior reflects the agent's desire to move about when not under other influences. The wandering may be a random walk or motion in a predetermined direction.

Group Formation Behavior

The Group Formation Behavior seeks to establish a specific spatial relationship between adjacent neighbors. The work by Reif and Wang' (1999) on Social Potential Fields provides the basis for establishing and maintaining this spatial relationship between agents within the model. In addition to the work by Reif and Wang, the work by Dudenhoeffer and Jones (2000) incorporates the concept of a neutral zone within the social potential field. The neutral zone permits the Wandering Behavior to activate and promotes expansion of the collective in a specific direction.

Social Potential Fields are based on the underlying concept that an agent is influenced by his immediate neighbors. A force vector is used to represent the influence exerted by an agent's neighbors. The nature of the force can be attracting or repelling, depending on the distance between agents. The sign and magnitude of the force is represented by the force function (Reif and Wang 1999):

$$f(d) = \frac{-c_1}{d^{\alpha_1}} + \frac{c_2}{d^{\alpha_2}} \qquad (1)$$

Equation (1) is the force function used in the Group Formation Behavior model

where $c_1, c_2 \geq 0$, $\alpha_1 > \alpha_2 > 0$.

This function creates a repelling force if a neighbor is close and an attracting force if the neighbor is far away. If a neighbor is too close, the agent tends to move away and gain further separation. If the neighbor is too far away, the agent moves toward the neighbor to close the distance between them. The definitions of what is too close and too far away are arbitrary and represent flexibility in configuring the behavior, depending on the mission and desired sensor coverage.

In Equation (1), *d* represents the distance between neighboring agents. The constants, c_1, c_2, α_1 and α_2 determine the slope and equilibrium point of the force function. The equilibrium point is defined as the distance in which the combined effect of the repelling and attracting forces is zero.

This force function has the following characteristics:

1. Attraction is controlled by the $c_2 / d^{\alpha 2}$ term.

2. Repulsion is controlled by the $c_1 / d^{\alpha 1}$ term.

3. The equilibrium point where the combined effect is zero is given by $d=(c1/c2)^{(1/(\alpha 1 - \alpha 2))}$.

Note that this function represents the force applied by a single neighboring agent. In practice, all perceived neighbors apply forces. The resulting force is the vector summation of all the forces applied by all neighbors. Another model parameter that may be set is to have neighbors of different types that exert forces using different force functions. However, the simulation results presented in the next section use a homogenous set of agent types and hence a single force function.

To understand the effect of multiple force vectors on a single agent, consider agent A with perceived neighbors $N_1, N_2, ..., N_k$ with distances $d_1, d_2, ..., d_k$. The individual forces applied by the neighbors is given by:

$$f_A(d_i) = \frac{-c_{1i}}{d_i^{\alpha_{1i}}} + \frac{c_{2i}}{d_i^{\alpha_{2i}}} \qquad (2)$$

The combined force applied to agent A denoted by *F(A)* is:

$$F(A) = \sum_{i=1}^{k} f_A(d_i) \qquad (3)$$

Equation (3) represents the force magnitude. It does not represent behavior. Behavior is the reaction to the forces applied and is realized in the agent by either the desire for motion in a certain direction or the desire to remain in place.

As mentioned earlier, this paper introduces an adaptation to Reif and Wang's presentation of Social Potential Fields. The adaptation is the introduction of a *critical force*. The *critical force* is defined as the magnitude of force below which the agent feels no effect. As an example, a critical force set at 5 implies that a cumulative force, *F(A)*, would require a magnitude greater than 5 to cause a reaction by the agent. By careful selection of the force function *f(x)* and the *critical force*, a neutral zone between repelling and attracting forces is created (Dudenhoeffer and Jones 2000). Within this neutral zone, no force effect exists.

This neutral zone accomplishes two purposes. First, it minimizes movement oscillations around the equilibrium point where the sign of the force changes. Second, it provides an opportunity for additional behaviors previously subsumed by the force effect and Group Formation Behavior to have an effect on the agent.

In the agent model, during periods when the agent resides within the neutral zone, the Wandering Behavior dictates the desired motion of the agent. In our model, the Wandering Behavior directs the agent to head east. The agent wanders east until the critical force is again reached. At this point, the Group Formation Behavior is activated. This combination of both behaviors working in conjunction not only promotes a uniform spatial relationship between neighbors, but it also causes the entire formation to preferentially expand and move in an easterly direction.

This type of behavior readily supports a scenario in which the agents are batch dropped or are dispensed from a canister and are tasked with establishing a uniform sensor net across a specified area.

Collision Avoidance

An agent looks ahead at the position of his next intended move. If another agent or obstruction is detected, the agent will evaluate a position 90 degrees to the right of the intended position. Again the agent evaluates this position. If occupied, the agent will turn 90 degrees right and repeat the process. If after turning in a circle, no move is evaluated as "safe," the agent will remain in place for that simulation step. On the next simulation step, the process begins again.

REFERENCES

Arkin, Ronald C. 1998. *Behavior-Based Robotics*. Cambridge, Massachusetts: The MIT Press.

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

Dudenhoeffer, D. D. and M. P. Jones, 2000. "A Formation Behavior for Large-Scale Micro-Robot Force Deployment." In *Proceedings of the 2000 Winter Simulation Conference*, eds. J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, 972-983, Orlando, Florida: Institute of Electrical and Electronics Engineers.

Reif, John H. and Hohgyan Wang. 1999. Social potential fields: A distributed behavioral control for autonomous robots. *Robotics and Autonomous Systems* 27: 171-194.

Reynolds, Craig W. 1987. Flocks, herds, and schools: A distributed behavior model. In *Proceedings of SIGGRAPH'87*. Reprinted in *Computer Graphics* 21 (4): 25-34.

Shaw, E., 1975. Fish in schools. *Natural History* 84 (8): 4046.