AN OBJECT-BASED INTERACTION FRAMEWORK FOR THE OPERATION OF MULTIPLE FIELD ROBOTS

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF AERONAUTICS AND ASTRONAUTICS
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Henry L. Jones II

February 2003

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Stephen M. Rock
(Principal Adviser)

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Pamela J. Hinds

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Terry Winograd

Approved for the University Committee on Graduate Studies.

~

To my source of inspiration and motivation: my parents, Frank and Ann Jones

~

Abstract

Today's field robots, such as the Sojourner Mars rover or the Predator unmanned aerial vehicle, work alone to accomplish dirty, dull, or dangerous missions. Plans for the next generation of robotic systems call for multiple field robots to conduct these missions cooperatively under the direction of a single operator. This research examines the role of the operator in multiple-robot missions and creates a human-robot interaction framework that supports this role – a vital step toward the successful deployment of these future robots.

In a typical user-centered approach to the development of a human-robot interaction framework, the work practices of the robot operator would be observed, characterized, and integrated into the design. Unfortunately, there are no settings where one can study the operator of multiple robots at work because no such systems have been deployed. As an alternative, this research incorporated a surrogate setting that could be used to inform the early interaction design of multiple-robot systems. Police Special Weapons and Tactics (SWAT) teams were chosen as this setting, and an ethnographic study of SWAT commanders was conducted. Concepts from the interdisciplinary study of geographically distributed work, including common ground, shared mental models, and information sharing, were used to understand and characterize the ethnographic observations.

Using lessons learned from the surrogate setting, an implementation of a new human-robot interaction framework was demonstrated on the Micro Autonomous Rovers (MAR) platform in the Aerospace Robotics Laboratory at Stanford University. This interaction framework, which is based on the sensing and manipulation of physical objects by the robots, was derived from the finding that references to physical objects serve as an essential communication and coordination tool for SWAT commanders. A human-computer interface that utilizes direct manipulation techniques and three-dimensional computer

graphics was created to test the new interaction paradigm. Using this interface, a single operator can coordinate the actions of multiple robots. Operators with many different levels of experience with robot operation were able to conduct a variety of complex missions using the MAR robots.

Acknowledgements

The most appropriate place to start these acknowledgments is with the people who built the foundation of my education – my elementary, junior high, and high school teachers and principals in the Richton School District, and my professors at the University of Mississippi. They engendered in me a love for education by being unyieldingly supportive, caring, and challenging. They launched me to this place in the educational firmament, far above where I ever thought I might reach.

My friends in Mississippi have been an unyielding source of encouragement, even though our time together has, unfortunately, became increasingly rare. I thank Adrian Strickland, Carey Fulmer, and Vincent Jones for being my best friends in the most formative years of my life and teaching me how to be a true friend to others. Heath James, a faithful companion through many memorable experiences, is the only person I know who never runs out of an outrageously funny thing to say or do. Judson Hilburn has been a loyal friend since our first year of college and an untiring source of the best wit on Earth.

Ginger Dunnam Smith sent me to California with love, even though it meant heartache for us both. I thank her for the way she always made everyone around her -- especially me – feel better about themselves. She is the star of all my best memories from Ole Miss.

I must certainly acknowledge the outstanding friendship of my California friends, who have done a phenomenal job of giving me a second family here in the Land of Fruits and Nuts. Yee Lee, Roger Kuttan, Toby Rossmann, Jenn Stroud Rossmann, Sam Yen, Eve Farrington-Yen, Martin Snyder, Audrey Krat Frew, Eileen Huster, Louise Warren, Dave Catechi, George Lee, and Ryohei Urata have each made me feel welcome and accepted from the day we met. Carol and Don Scifres and their family have been so hospitable to me that they should certainly be considered honorary Mississippians. I could not ask for better friends.

The Aerospace Robotics Laboratory has been blessed with many academic generations of exceptional researchers, and I have been fortunate to rub shoulders with a few of these outstanding intellectuals. In particular, I want to acknowledge the influence of the previous generation of Jeff Russakow, Kurt Zimmerman, Eric Miles, David Miles, H. D. Stevens, Stef Sonck, Gordon Hunt, and Steve Fleischer; my generation of Tobe Corazzini, Andrew Robertson, and Bob Kindel; and the next generation of Chris Clark, Jason Rife, Tim Bretl, Jack Langelaan, Jorge Moraleda, Masa Matsuoka, Kristof Richmond, and Jinwhan Kim.

Our lab and our department would fall into chaos without the unending efforts of Jane Lintott and Sally Gressens, respectively. These two wonderful women have watched over me like guardian angels since our first meeting when I was a prospective graduate student, and there is no knowing where I would have ended up without them.

The ARL had two pillars of omniscience when I first began to work here, Gad Shelef and Godwin Zhang. They both overlooked the fact that I didn't know anything myself, and patiently taught me what it means to be an exceptional engineer. I am so thankful that I have had the opportunity to get to know Aldo Rossi, whose generous help with my various projects was far overshadowed by the friendship he has given and the wisdom he has passed on regarding how to have fun throughout life, no matter what.

Three fellow lab members, Bruce Woodley, Heidi Schubert, and Ed LeMaster deserve special recognition for the parts they each played in my career here. Bruce was my very first friend in the lab, and I could not have known how fortunate that was for me. He has kept me honest, educated, and stimulated the entire time that I have known him. Heidi was my first partner on a project in the lab, and that early experience teaching the free-flying robots to play catch cemented my decision to pursue a doctorate in robotics. Ed, as a fellow HUMMINGBIRD helicopter team member and contemporary through many classes, projects, and extracurricular events, has had to put up with far more than his share of bad jokes and worse stories, and weathered them all with amazing stoicism.

I feel that I could write a separate dissertation about the PhD process itself, or at least a very bad novel. If so, Andreas Huster and Eric Frew would be as likely to be my protagonists as myself, as the years that we have spent commiserating together have combined our educational experience into one entity. They made this dissertation possible through their honesty, criticism, encouragement, perspective, and weekly feedback sessions. Andreas has been a role model for me since our first meeting as two aspiring lab members, from his colored socks to his unending patience and goodwill. Eric has spent more time with me than any other person in the universe, and surprisingly has very few ill effects to show for it. He has had a profoundly positive influence on me. I cannot wait to witness the far-reaching effect on the world that Andreas and Eric will surely have.

Professor Robert Cannon and Professor Steve Rock have managed to create a research environment that makes technological progress so natural that it can be easily taken for granted. Even as I approach the completion of my degree, there are times when I cannot believe that I am actually a member of this excellent lab that I read about many years ago. Professor Cannon deserves a perpetual "Thank you" from me because he has never stopped being my biggest supporter at Stanford, particularly at those times when support was needed most. Professor Rock has been an advisor of worth beyond value; from every indication, I have had an exceptionally outstanding learning experience from start to finish at Stanford and he is the person most responsible for this. I am sure that I will draw from lessons he has taught me for the rest of my life.

My long-time roommates, John Owens and Vijit Sabnis, were perfect counterbalances to my work life in every way. Surely there must be something significant in the ability for three people to live together for seven years with so little conflict. Once our time at Stanford is over, I will undoubtedly miss their companionship, even our silent breakfasts of cereal, the newspaper, and SportsCenter.

My best friend, Clark Love, could not be thanked enough in the space available here. Somehow, he has been able to accept all my many faults and still be willing to put the energy into our friendship to keep it strong even with 2500 miles between us. Starting a company with him was one of the best decisions I have ever made, as I learned more from him through that experience than I thought possible. I hope that I can continue to return the friendship to Clark and his wife Stephanie that they have unfailingly shown to me.

I thank my dearest friend Alison Coy for introducing me to the poetry that life can be. My personality, my opinions, my experiences, and my thoughts -- the dearest things to me – would be much poorer without her. Because of her, I look at the entire world differently and I can see the best parts of it. She was the opposable thumb that I needed to pick up life and examine it from every angle. The impact she made on my life will last forever.

My extended family, particularly the McCullys and the Scotts, have been supportive of both my educational pursuits and my personal development. They are the perfect aunts, uncles, and cousins. I only wish I saw them all more often.

For their entire lives, my brothers John and Walter have fed my will to excel. They push me to work hard to be my best because they are naturally gifted in so many ways. For a very long time now, John has been the loudest and most constantly noble influence on my life. Walter has already become the person that I would like to be; I admire him and adore him. I know that I can count on Walter, John, Emily, and my nephew Samuel as perpetual sources of affection and caring.

My entire family has always shown incredible amounts of kindness, intelligence, humility, integrity, and initiative, and the supply of these traits comes directly from my grandparents. They are perfect role models and the basis of the expectations that I set for myself. I miss my paternal grandparents, Henry and Nina Jones, greatly as I complete this degree, and am thankful that my maternal grandmother, Margaret McCully, has been able to see the successful culmination of my formal education. Every conversation with her gives me hope for the future.

My parents, Frank and Ann Jones, are truly remarkable. The older I get, the more overwhelmed I become by the recognition of the many gifts they have given me, tangible and intangible. They have given much of themselves and their lives, and I am the beneficiary. Passing on their gifts to the next generation will take a lifetime of dedicated effort, just as they have done. Words cannot possibly express my appreciation to these two wonderful people.

To all of my friends, thank you. To my family, I love you completely.

Finally, I thank God for blessing me far beyond what any person could ever deserve.

Table of Contents

List of Figures

Chapter One

Introduction

Plans for future robotic systems call for multiple robots to be operated by a single person, but practical human-robot interaction frameworks for such systems have not yet been identified. In this research, an approach based on the four-step design process shown in Figure 1.1 was used to create an effective framework. This approach requires that four steps (Observation, Analysis, Innovation, and Implementation) be carried out in an appropriate setting -- ideally, one based on ethnographic observation of current operators of robots. However, as there were no operators of multiple robots to observe, a new hybrid setting approach was taken in which the first two steps of design – observation and analysis – were conducted using humans-only teams, followed by the last two steps – innovation and implementation – for a human-robot team. These four steps and the design setting will serve as the main organizational structure of this dissertation.

This dissertation work advances the fundamental understanding of the human as commander of a spatially distributed team, which in turn may be used to refine current and future human-robot interaction frameworks for field robot systems. The approach required a convergence of experimental methods from three diverse fields of study: leadership of distributed human teams, high-level control of robots, and direct manipulation computer interfaces.

The outcome of this approach is a human-robot interaction framework for multiple field robots that utilizes the sensing and manipulation of physical objects by the robots. This

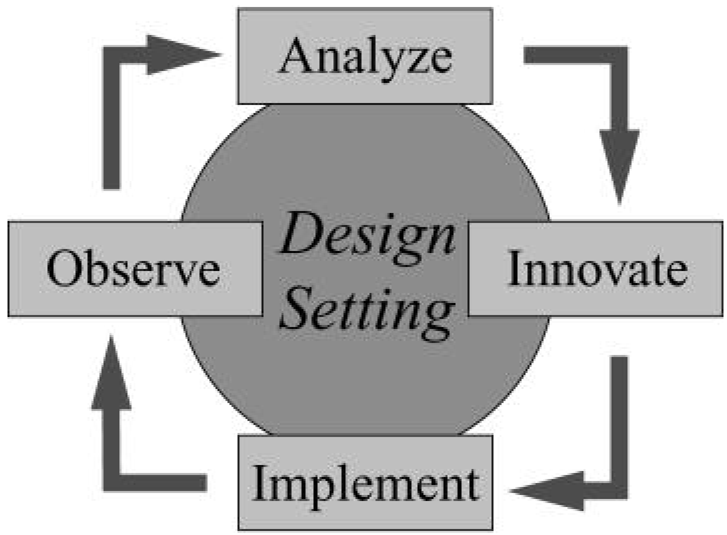

Figure 1.1: **Design Process**

This depiction of the standard design process uses four iterative steps that are influenced by the design setting.

interaction framework was based on the finding that references to physical objects serve as an essential communication and coordination tool for leaders of distributed teams. The interaction framework was demonstrated on robot hardware to show that it addresses the key characteristics of field robot deployments such as the reliance on robot sensors for all information, communications and operator bandwidth constraints, and the merging of data from multiple sources. Some assumptions about robot capability, particularly that the robots can sense and manage physical objects and that they can carry out commands robustly without constant vigilance by the operator, are necessary to support this novel interaction framework. Robots with this level of autonomy do not yet exist outside the laboratory environment, but the work of many researchers is bringing them closer to deployment in the field.

In its first three sections, this chapter will explain the motivating problem, lay out the approach to solving this problem, and state the research contributions and innovations, respectively. A brief overview of the complete dissertation is given in the final section of this chapter.

1.1 Motivation

1.1.1 Field robots and unmanned vehicles

Field robots are robots that extend a person's sensing and/or manipulating capability to a location remote from that person [1]. Field robots usually have enough autonomous capability to control their own movement and act on high-level commands from their users. These robots are important because they often are called upon to do things that humans cannot, will not, or should not do. Humans cannot yet go to Mars, they will not willingly traverse minefields, and they get bored flying over an area taking photographs for days on end. Field robots are performing these roles with humans as supervisors from afar.

Applications

Field robots typically have been deployed when the possibility of putting humans on location has been deemed too dangerous or impossible. As the robots prove their reliability for these applications, they will be used increasingly in situations that are simply dull or frustrating for humans.

One of the most visible applications of field robots has been exploration (Figure 1.2). Starting with Russian landers on the Moon's surface [2], exploration robots have extended the reach and sight of mankind to Venus and Mars [3]. These field robots, controlled directly by operators on Earth, have been either mobile rovers or stationary landers with arms. Scientists have also used various types of remotely piloted and semi-autonomous vehicles to explore the depths of the Earth's oceans [4].

A more recent high-growth area in the use of field robots has been the reconnaissance of dangerous areas by the military. These Unmanned Aerial Vehicles (UAVs), such as Predator and Hunter (Figure 1.3), keep human pilots out of harm's way [5]. Through frequent use, these craft are building confidence in the reliability and usefulness of field robots [6].

Researchers are working to develop enabling technologies that will expand further the utility of field robots. Future planetary exploration robots will utilize significant autonomous capabilities to achieve ambitious mission goals [3]. Other research is underway to create highly capable Autonomous Underwater Robots (AUVs) that will perform unprecedented science for marine institutions [7]. Military applications for robots will also grow as evidence of operational reliability bolsters the confidence of planners and engineers to move beyond reconnaissance as the primary function of UAVs [6]. Ultimately, robots and humans should be able to operate as a team with their duties mirroring their capabilities. Robots will perform remote dangerous or repetitive tasks without constant direction while humans assign those tasks creatively and according to their many sources of information about the mission. In the resulting system, the human may be considered part of a spatially distributed collaboration with a computer acting as the conduit between the operator, multiple robots, and other people or objects in the environment.

Components

Three components make up a field robot system (Figure 1.4) – the operator, the robot, and the computer interface between them.

Operator

Even with very autonomous robots, ultimately some operator or team of operators who deployed the robots is responsible for the success of their mission. Operators of the currently deployed systems play an active role that can range from direct teleoperation to sensor monitoring. Field robot operators differ from users of most other robots because of the fundamental constraints on information about the robot (Section 1.1.3) created by the distance between the operator and robot.

Robot

The robot is the physical machine that accomplishes the mission at the work site. Robot capabilities vary dramatically and are primarily dependent on the requirements of the mission. Field robots usually have more advanced autonomy, including decision-making and fault resolution capabilities, than service and entertainment robots. Information about the status and conduct of a field robot typically must come from the robot itself, as there is rarely a third-person perspective on the robot worksite. This characteristic creates special constraints on operation methods.

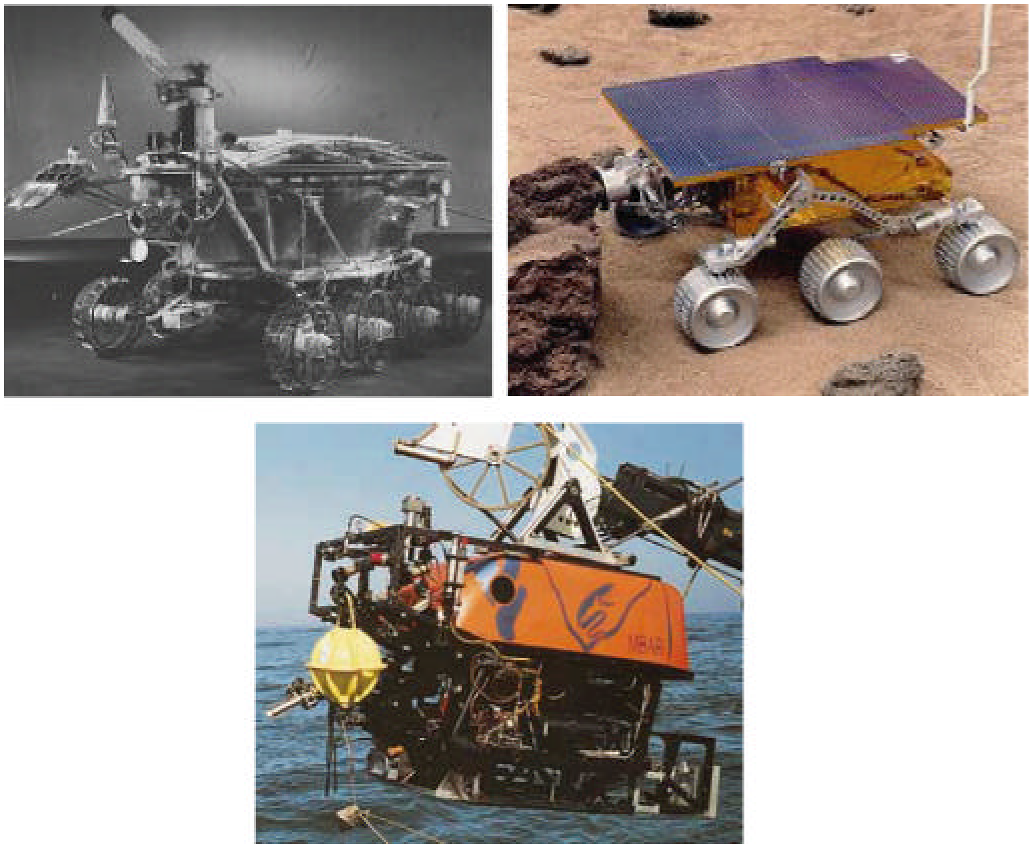

Figure 1.2: **Field Robots Used for Exploration**

Clockwise from top left: The Lunakhod rover was sent to the moon by the Soviet Union in 1970; the Sojourner rover was part of the Pathfinder mission to Mars in 1997; the Ventana unmanned submersible is used by the Monterey Bay Aquarium Research Institute for many types of oceangoing research.

Computer interface

A computer of some type forms the vital link between the operator and the robot. The interface presented by the computer to the operator acts as a lens through which the operator sees the robot and its worksite. Due to the nature of field robot applications, data regarding the worksite or robot operation may be incorrect or incomplete. The computer interface must act as the clearest possible lens so that the operator can maintain the best available situational awareness of the mission.

Figure 1.3: **Unmanned Aerial Vehicles for Military Reconnaissance**
Left: the General Atomics RQ-1 Predator UAV, flown by the United States Air Force;.
Right: the TRW Hunter UAV, flown by the United States Army

1.1.2 Multiple robots, single operator

Although currently deployed field robots operate alone, designers have proposed systems of multiple robots for many future operations. For example:

- Search and rescue operations following a disaster: multiple robots work alongside and support human members of emergency response teams [8],
- The Defense Advanced Research Projects Agency (DARPA) Tactical Mobile Robotics (TMR) Program: teams of small, low-cost, semiautonomous mobile robots perform urban reconnaissance [9],
- US Marine Corps' Broad-Area Unmanned Responsive Re-supply Operations (BURRO): autonomous unmanned helicopters provide logistical support to forward battle elements [5],
- Coordinated ocean search and rescue: long-duration aircraft enable non-stop operations [10],
- Space telescope construction: robots minimize or eliminate the requirement for astronaut participation [11], and
- Mars exploration and construction: rovers and other vehicles prepare for eventual human exploration [12].

These systems are expected to provide significant improvements over current single robot deployments in many ways, as the additional robots should be able to enhance robustness, increase sensor coverage, provide redundancy, and widen the scope of the missions. However, there is one drawback to increasing the number of robots without a subsequent change in their current design: an unacceptable increase in the number of operators required for each deployed system. Most current field robot systems require multiple operators for each robot, yet designers would prefer that the number of operators in future systems be significantly reduced, ideally to just one.

The deployment of multiple-robot, single-operator systems will require significant achievement in three research areas:

- Self-monitoring and robust fault handling,
- Independent collaboration and cooperation, and
- Efficient command and control methods.

By robustly self-monitoring themselves and acting appropriately when failures occur, robots take on greater responsibility for how they interact with their environment. Robust robots allow the operator to spend less time monitoring each robot and to consider the status of the team as a whole. Potential dangers to the robot may originate internally (e.g. when fuel is low or when actuators malfunction) or externally (e.g. when the robot must avoid obstacles or reject unintentionally harmful commands from the operator). A few different strategies for addressing these issues exist, from contingency management systems to reactive behaviors [13]. The best choice is usually application-dependent, determined by the complexity of the robot, the level of structure in the environment, and the resources available during development. Field robots must prove themselves to be much more reliable before operators are likely to turn their attention away for long periods. In the near term, operators who serve as dedicated trouble-shooting "mechanics" could be used to free up the robot team commander for higher-level concerns.

To operate effectively in groups, robots need to organize themselves independently to collaborate on tasks. They should be able to interact safely and effectively with one another. Reliance on a central location for such coordination would reduce the flexibility of the robot deployments and potentially introduce a single point of failure for the entire system. Humans are currently the source of coordination when multiple robots work together, but this requires substantial efficiency sacrifices during system design. If the field robots are able to conduct parts of their mission independently, then the workload required of an operator is reduced and his or her attention may be used elsewhere.

The third area where technology advancements are required, determination of efficient command and control methods, is the field of research addressed by this dissertation. Human-robot interactions for single robots have been developed and tested over many years, but useful interfaces for the unique environment of multiple robots and a single operator have not yet been identified. Work in this area requires that the advancements in robot robustness and cooperation succeed so that the operator is able to move beyond the role of robot driver or data monitor. A lack of attention to this area will result in an overall reduction in the capability of the robot system by limiting the effectiveness of the operator while increasing his or her responsibilities.

Unmanned aerial vehicle (UAV) development programs have recently become the most active area of unmanned vehicle and field robot development in all three of the aforementioned research areas. An oft-stated future goal is for one operator to control multiple UAVs simultaneously. For example, DARPA plans state that by 2010, a single operator should be able to direct *at least* four active-duty unmanned combat air vehicles (UCAVs) simultaneously. If the number of operators is reduced to one, cooperative UAV missions are expected to improve because all members of the UAV ground control system would have the same knowledge of UAV system and mission status. Reducing the operator-to-robot ratio is also expected to result in significant operational and support cost savings [14]. For the DARPA UCAV program to be considered a success, it must demonstrate that a small team of operators can direct numerous unmanned vehicles, possibly up to 20 UCAVS, in a coordinated fashion [6].

Military experts, in unofficial personal conversations, have said that there is a momentous gap between the multiple-robot expectations of these military planners and the realities of the military systems' current development. Access to the latest military interfaces for inspection was very limited, but they appear to be extensions of previous interaction methods that rely on the specification of robot position as the primary command mechanism. The most widely adopted human-robot interface in the US military has allocated 4 displays for one robot – two for the air vehicle operator and two for the mission payload operator [14]. A recent classified system had been specified to have a vehicle:operator ratio of 3:5; however, the actual system had a 1:4 ratio. The designers of the second generation of another system, with a 1:3 ratio in the first generation, have been told to invert the ratio. The developers of

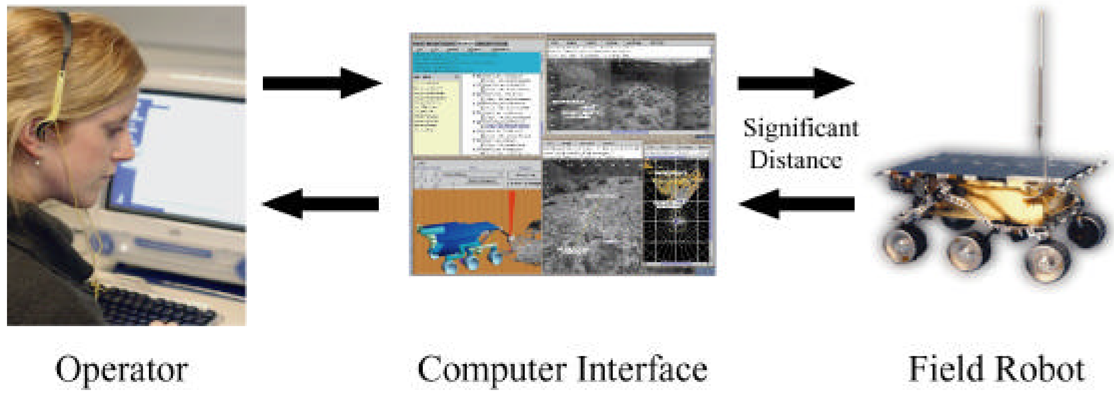

Figure 1.4: **Field Robot System Components**

these systems don't know if they can even expect a 1:1 ratio in the near future, yet they expect that the demand for and expectation of high ratios to grow.

Certainly, a need for greater autonomous capabilities by the vehicles contributes to this discrepancy between requirements and capabilities. Many advances in the areas of operational robustness and team cooperation are necessary to achieve the future system goals. Yet, part of the explanation for this gap between expectations and available technology may be a lack of previous research addressing the operator command and control issues. According to the Air Force Research Laboratory, in 1998 [15],

- No methods existed for operating several UAVs from one station,
- No programs were addressing the applicable enabling technologies, and
- Designers planned to "do human factors" after systems design, even though the human component is one of the most important aspects of a multiple-UAV system.

1.1.3 User interaction issues

The research described in this dissertation is in a new area of multiple-robot study, in which researchers have begun to examine the issues that controlling multiple-robot systems will create for their users and for humans working alongside robots in the field. Most prior multiple-robot research focused on other difficult problems, such as navigation, self-monitoring, and autonomous control. While progress on these other issues has made robots much more capable, they have not become much more widely deployed. When looking to the future in 1996, Thomas Sheridan, one of the foremost experts in robotics, said, "It is clear that progress on the following human-robot interface problems are and will be crucial to making (use of robots) widespread (emphasis added to highlight areas addressed by this research) [16]:

- **Techniques for allocating functions between human and machine**
- Computer graphics
- Decision aids
- **Command and control languages and techniques"**

A National Research Council study of Unmanned Air Vehicles [17] agreed that the human-vehicle interaction was one of the largest hurdles for unmanned systems deployment, finding that "Experience with current operational and developmental systems (or concepts) has shown that the integration of the human and machine components of the system is a much greater challenge than many anticipated."

A number of fundamental conditions generate this challenge, and they are discussed in the remainder of this section.

Remoteness of human controller

A variety of problems is created when the operator is removed from the immediate location of the robot. The scope of sensory cues that are taken for granted for basic operation when the operator is collocated with the robot includes indirect audio feedback (whirring fans, servomotor whining, gear grinding), visual information (high-frequency jitter, worn or broken components), and even smell and touch (burning or hot electronics components). In addition, the operator may have few options for handling complete robot failures. There is no "Simulator Reset" button available to rescue the remote robot from a disastrous situation.

Lack of independent 3rd-person sensors

Because the operator is remote, she[1] cannot personally double-check the data from the robot sensors. In most field robot applications, there are no independent 3rd-person sensors and thus no extrinsic way to double-check robot information. Although operators can be presented with reconstructed 3rd-person perspectives of robot operational environments,

---

1 For convenience in upcoming discussions, feminine pronouns will be used throughout this dissertation to refer to operators of robots and masculine pronouns will be used to refer to commanders of human teams.

this display method can be deceiving because the entire environment is built from data provided by the robot sensors from a 1st-person perspective. In addition, important information may not be observable by the robot at all. As a result, the successful execution of an operator's intent depends on only the partially-observable physical state of the robot and its environment.

Time-variant capabilities

The capabilities of the robot can change from moment-to-moment within the time span of a single mission. Robots are complicated mechanisms controlled by complex computer software, so there are many potential points of failure -- both permanent (such as when a wheel motor breaks) and temporary (when a grasp point is not visible on an object). In a collocated human-robot system, the underlying reasons for time-variant capabilities can be determined by the operator by inspecting the robot and its environment. For field robots, this information may be completely hidden from both the robot and the operator.

World model location for multiple robots

A field robot maintains a model of its environment for use in its own planning, and it also sends data about this world model to the computer interface. The operator sees this model, reconstructed by the interface, and combines it with external knowledge about the system. However, the proper way to handle the multiple world models created when multiple robots work together is not clear. If the world models are combined automatically, the correct choices of which data to use and the location of the data fusion process are not clear. On the other hand, if the world models remain separate, potential sources of confusion for the operator are created.

Interdependence of robot and user interface programs

All robots require complex software programs to operate, and field robot systems further require separate human-computer interaction applications based on equally complex software. In current systems, these two code bases are interdependent – usually, the interface is explicitly programmed to provide the correct robot functionality to the operator. Changes to the field robot thus require changes to the interface as well. An interdependent relationship at the software code level merely adds complication for developmental or research systems, but the actual performance of multiple-robot operational systems could suffer if the robots and the interface application are so tightly coupled.

Bandwidth limitations

Operation of multiple robots by a single operator puts stress on specific parts of the field robot system, particularly the workload of the operator and the capacity of the communications system. These two areas can be thought of as limitations in bandwidth of the operator and the communication system, respectively. These bandwidth limits form fundamental constraints on the operation of multiple field robots.

Human bandwidth

Increases in workload requirements for field robot operation typically have been addressed by adding additional operators. As a result, most current field robots require more than one person to operate. Single operator control of multiple robots, on the other hand, requires that each robot instead only require a fraction of a single person's mental bandwidth. Consequently, the attention and processing power of the operator should be treated as a very valuable resource. Reducing the operation requirements of multiple robots into the workload capacity of a single operator requires focused study of the most efficient means of using the available operator bandwidth.

Communications bandwidth

Another engineering hurdle for multiple robot systems is the bandwidth requirement for communication between the robots and the operator. For instance, the U.S. Department of Defense is only able to fly three reconnaissance unmanned aerial vehicles at one time in a given theatre due to communications bandwidth limitations. Because future increases in available bandwidth will be met with increased demands from all bandwidth users, communications bandwidth will continue to be a fundamental constraint for field robot operation.

Efforts to increase the efficiency of communications bandwidth use through advanced communication or data compression techniques will provide short-term improvements but will not contain long-term answers as bandwidth demands increase. In addition, these approaches do not address the operator bandwidth constraint. Likewise, decision aids at the operator interface help the user but do not alleviate the communication bandwidth problem. Increased autonomy on the robots can reduce the data exchange required for operation and thus help with both constraints; autonomy in this respect is a worthwhile goal. However, the operator must maintain a certain level of situational awareness and influence the actions of the robots to some extent. Therefore, a limit exists on the amount of autonomy that is acceptable. The most useful place to study, then, is the operator – to understand the minimum information and control that she needs to provide effective leadership, and then subsequently determine the level of autonomy required of the robot.

1.2 Approach

Given the unique environment of a multiple field robot system, this research took a novel approach to creating an interaction framework suitable for a single operator.

A useful way to picture the basic design process for many systems, particularly those with significant computer software components such as field robots, is to construct a four-step loop as shown in Figure 1.1. This iterative or, in some cases, spiral approach [18] is an alternative to concentrating all design efforts at the start of a project. The first step, observation, seeks to create raw data about the system or its use. Analysis, the next step, takes this raw data and applies rules, algorithms, and organization to draw meaningful conclusions. The innovation step then takes the analysis results and creatively seeks additions, improvements, or other changes that might move the system's performance towards its desired state. Implementation, the final step in one loop through this iterative process, carries out the innovations so that observations of the modified system can be conducted to start the loop again. Most design processes ultimately stop after an implementation meets the design requirements; others may go on continuously or might end unsuccessfully in the midst of the loop before the product is complete.

A very important aspect of the design process is the choice of setting. Whether it be a field robot system, a street intersection, or a CD-ROM drive, the setting effects the nature of the observation and all the subsequent steps. Most field robot research has concentrated on the robot as the setting for design.

1.2.1 Pursue user-centered design

As a rule, systems built around autonomous robots have often provided mechanisms for human involvement only as an afterthought. In addition, the human-system interaction frameworks for robots are usually designed by the engineers responsible for the robots' development. The resulting interface thus is often based on the engineer's model of the underlying system rather than the task that the robot was built to accomplish [19]. However, robotics could benefit from the lessons learned in other fields [16] that have embraced the concepts that constitute user-centered design.

User-centered design, a concept popularized by Don Norman and Stephen Draper [20], is a philosophical approach to interface design that seeks to put the user first and the technology second. Typical user-centered design efforts incorporate end user input from the beginning to encourage the maximum possible ease of use and utility of the final product. "Products" created through a user-centered design process are more likely to be used correctly, to require less instruction, to cause fewer errors, to be used more often, and to be more enjoyable to use [21].

There is one major issue with applying user-centered design concepts to the design of an interaction framework for operating multiple field robots – there are not yet any users to study. In fact, requiring a single user to operate multiple robots is so different from existing field robot operation environments that extending lessons from these current interaction frameworks would not be likely to provide enough insight for effective design.

In addition, the development of the three field robot components (operator, robot, computer interface) is tightly coupled and typically requires much iteration to converge on an effective final design. Such iteration could have been used to address the need for users to study by waiting until the first iteration was complete. Unfortunately, due to the complexity of most field robot systems, iterations are costly in resources, time, and money. The method used to address this dilemma is described in the next section.

1.2.2 Find an analogue for field robot operation

If a close analogue to field robot operation – with an existing user base – can be found, then a user-centered design process can be pursued. Yet, the command of robots on other planets or far under the sea is clearly an extraordinary situation. Humans tend to find connections between past ordinary experiences and unusual events as a way to cope more easily with the associated uncertainties. An open question before this research began was whether design could take the same step and apply known work practices to an unusual but similar work setting.

This research used the setting of geographically distributed work teams as a proxy for the work environment of a field robot operator. A great deal of recent study has focused on such teams, as the Internet has allowed virtual teams and telecommuting to become a normal part of the business world. Some of the most important concepts developed through this research field that are applicable to field robots are introduced in Chapter Four.

This dissertation work is the first to examine the existing body of knowledge in humans-only teams and apply it to teams of humans and field robots. This research augmented the prior humans-only team research through ethnographic studies of task-focused geographically distributed teams – police Special Weapons and Tactics (SWAT) teams. The effect on the design process of this crossover of humans-only research into robotics, creating a hybrid design setting, is shown in Figure 1.5.

Police SWAT teams provide an analogous setting to field robot operation for three main reasons: they have a single commander who is ultimately responsible for the team's success but must command from a remote location, they require cooperation among agents who have relative sensors with limited range, and all information received by the commander originates from the field agents. Although SWAT teams are not perfect analogues – they are far different from robots in autonomy, sensing and communication capabilities – they are consistent in key areas and are very suitable for ethnographic study. A more comprehensive discussion of SWAT teams as the ethnographic setting for field robot research is found in Chapter Three.

1.2.3 Complete one design cycle

Completion of one design cycle (Figure 1.5) would serve two purposes: the development of a human-robot interaction framework for the operation of multiple robots by a single operator, and the verification of the hybrid setting design process. This section provides a brief overview of the cycle, which forms the outline of the remaining chapters of this dissertation (Figure 1.6).

Observation

The first step of the design process is observation within the design setting for the purpose of developing data about the system. For this research, the observation took place in a non-robotics setting, police Special Weapons and Tactics (SWAT) teams. Five events (four field exercises and one actual incident) were observed, and they produced high-quality information for the remainder of the design cycle.

The main technique for observation used in robotics in the past were extensions of human factors studies to robot operation [22], but these methods are not as useful in the study of people working in unstructured environments. To address this shortcoming, this research turned to ethnographic field study techniques as a means to provide the necessary data. The observation step is described in detail in Chapter Three.

Analysis

The analysis step takes the raw data from the observation step and applies rules, algorithms, and organization to draw meaningful conclusions. This step also required techniques from outside the robotics body of knowledge, and the proper tools were again found in the geographically distributed work literature. The most important of the tools were the existing theory and structure that had been developed for spatially distributed teams of humans. These concepts were found to be directly applicable to both the SWAT teams and to the subsequent field robot interaction framework design. The analysis revealed that physical objects provided the basis for the SWAT command communications, and identified two primary roles of the SWAT commander – cultivating common ground and coordinating action. These concepts, described in Chapter Four, allow generalizable results to be obtained for use not only in robotics but also for the broader distributed work audience.

Innovation

The key roles, tools, and techniques revealed in the SWAT commander setting next needed to be converted into meaningful design innovations. The innovation step includes the identification of useful concepts that already exist, the extension of these concepts in new directions, and the creation of new ideas.

Through consideration of the fundamental results of the analysis step and existing knowledge of robotics, an applicable robot control architecture was identified – Object-Based Task-Level Control. Originally created for controlling field robot systems in space applications [23], Object-Based Task-Level Control (OBTLC) technologies had been applied to autonomous underwater vehicles (AUVs) [24], flexible manipulators [25], and factory workcells [26]. In OBTLC, the robot handles its own low-level control locally while the operator provides high-level task commands to the robot. OBTLC uses physical objects in

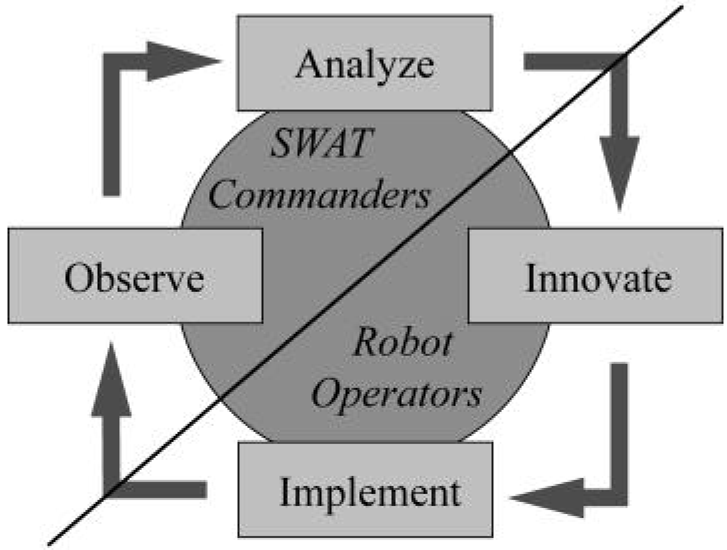

Figure 1.5: **Hybrid Setting Design Process**

The design setting has been split so that the observation and analysis steps take place with SWAT commanders, but the innovation and implementation are carried out with the robot operators.

the environment as the basis for the interaction with the human. However, because OBTLC was only used in situations where a single robot performed a single task, many of the field robot issues listed above in Section 1.1.3 were not addressed. In particular, past OBTLC-based interaction applications relied on perfect global sensors, were not capable of handling time-invariant skill lists, were not extendible to more complex robots that could perform multiple tasks on a given object, and had considerable interdependencies between robot and interface. A full discussion of OBTLC is given in a Chapter Five.

Once the roles of the operator had been characterized through observation and analysis and the robot architecture was determined, the third component of the field robot system – the computer interface – had to be addressed. Study of previous OBTLC computer implementations and examination of similar existing computer interfaces led to the development of a direct manipulation interface that used a three-dimensional graphical representation of the robots and their workspace. This interface is described in detail in Chapter Six.

Implementation

The final step for this research was to implement the results of the innovation step and test the complete human-robot interaction application. This implementation, on the Micro Autonomous Robots system in the Aerospace Robotics Laboratory at Stanford University, shows that the hybrid setting provided useful results for the initial design of a field robot system. The success of the system also shows that physical objects can serve as the basis for a human-robot interaction framework for a multiple-robot system. Single operators, after minimal training, were able to achieve complex goals using multiple robots simultaneously. A full description of the implementation is given in Chapter Seven.

1.3 Contributions

The primary contribution of this research is an object-based human-robot interaction framework that enables the command of multiple robots by a single operator. The secondary contribution of this research is the utilization and validation of a hybrid setting approach to user-centered design.

This section will briefly discuss these main contributions as well as associated ancillary contributions. When appropriate, references to the relevant dissertation section will be given. A more comprehensive discussion of these contributions is presented in Chapter Eight.

1.3.1 Created a new object-based human-robot interaction framework that allows a single operator to operate multiple complex field robots

The new interaction framework utilizes the objects sensed by the robots and the tasks afforded by those objects to coordinate the action of multiple robots simultaneously. Previous human-robot interaction applications for multiple-robot systems (Section 2.2.4 provides a survey of such systems) were designed to give the operator control over the positions of the robots only. The use of objects and object-based tasks encapsulates more intelligent robot behaviors so that field robots are capable of working without constant human direction.

Three contributions were made in the course of creating the new interaction framework.

Developed and demonstrated the first user-centered human-robot interaction framework for the operation of multiple heterogeneous field robots by one person

The user-centered design approach taken in this research was experimentally validated on a number of robots in a variety of situations (Section 7.2), demonstrating that the framework is a useful and flexible method for the operation of multiple heterogeneous field robots.

Extended an existing effective human-robot interaction paradigm (OBTLC) to a multiple-robot, multiple-task, multiple-object environment

This research made significant extensions to the Object-Based Task Level Control (OBTLC) paradigm to allow operation of multiple complex, heterogeneous robots. The new functionality required innovations with both the robot and computer interface components of the system described in Chapters Five and Six. This extension introduced three new concepts for the OBTLC paradigm.

An object-based dialogue ("Do what where?") that transcends an existing object-based manipulation paradigm ("Put that there")

This research developed a dialogue interaction metaphor for field robots that is flexible, dynamic, and task-oriented, based on a new "Do what where?" dialogue paradigm (Section 6.4) that is an extension of the well-known "Put that there" interaction paradigm developed by Cannon at Stanford University [27].

Context-sensitive affordances of multiple task choices

Context-sensitive affordances, a familiar concept for human-computer interactions, were used in a human-robot interaction for the first time (Section 6.4).

Integration of software agent paradigm to provide user assistance

Software agents were added to the OBTLC framework (Section 7.1.6) to assist the user with the more complex interaction requirements presented by a multiple-robot system.

Identified the implications of using physical objects as the basis for an interaction framework for the operation of multiple field robots

While the use of physical objects for references is natural for the operator and should produce a more effective human-robot interaction framework, this focus creates distinct challenges for implementation on real robots. The effects of these issues on robot system design are presented in Sections 5.4 and 6.3.

1.3.2 Utilized and validated a novel hybrid-setting approach to user-centered design

By utilizing a hybrid setting, the first interaction implementation could be informed about the roles and tools required by the operator even though no operators were available to study. Previous early-stage interaction development in robotics either was not concerned with the user or waited until the entire system could be built and tested. This research validated the hybrid setting approach by applying the analysis from a non-robotics setting to the development of a human-robot interaction framework and testing its feasibility with an actual multiple-robot system. As this approach was being pursued, three contributions to teamwork studies and to robotics were made.

Conducted field studies of leadership in extreme distributed work teams

Research in leadership in all types of distributed teams is still in its early stages, and this was the first research of its kind to study the leadership of 'extreme' distributed work teams. The focus of this study were police Special Weapons and Tactics teams (Chapter Three), who are spatially distributed once they are deployed and are led by a Tactical Commander who is located remotely from the team members. The field studies yielded two contributions that further the fundamental understanding of the extreme distributed work teams.

Recognized the importance of physical objects as tools in communication and coordination

The most important communication and coordination tool for the team were references to the physical objects in the environment. As will be discussed in Chapter Four (Section 4.3), objects provide unique common points that facilitate the reconciliation of various perspectives and serve as focal points for cooperative action.

Identified two vital roles of a team leader and observed how those roles are conducted

The analysis of the SWAT team observation (Section 4.3) revealed two important roles of the leader:

- Cultivation of common ground: This research found that the distributed team leader was in a unique position, due to his remote location and ability to focus solely on team leadership, to cultivate common ground throughout the team.
- Coordination of action: This research discovered that not only was the responsibility for coordination within the team heightened for a distributed team leader, but he also served as the sole integrator of external information and constraints into team activity.

Advanced the state of understanding of the role of the operator in field robotics

The roles of the robot operator and the tools she should be given to operate are not well understood nor have they yet become the subject of much research. The design requirements due to these roles and tools will have a significant effect on the entire robotic system, and should guide future research efforts in field robotics.

Identified the atomic component of field robot interaction framework: the determination of valid tasks to perform on an object and the subsequent command of a valid task by the operator

Based on observations and analysis of an analogous setting, this research proposes (Section 4.4.2) that the atomic component of the human-robot interaction framework should be the discovery of valid tasks to perform on an object and the subsequent command of a valid task by the operator. "Atomic" in this sense means that it is the smallest unit of interaction, from which interactions that are more complex may be created but smaller interactions should not be attempted.

Highlighted areas for focus in robot autonomy research to enable better integration with an operator

Three of the most useful such areas for autonomy research identified by this research are:

- Identification and manipulation of physical objects,
- Self-monitoring and intelligent notification when problems occur, and
- Independently organized cooperation and collaboration on high-level tasks.

Integrated research in the two fields of geographically distributed work and robotics for the first time

Research in geographically distributed work had not previously been extended to include field robotics, and the use of concepts from this interdisciplinary field had not been a formal part of any previous robot interaction design. This is the first application of any body of humans-only teamwork research to teams with robots. This research identified valuable teamwork research and applied its concepts to the development of a human-robot interaction framework.

1.4 Thesis Overview / Reader's Guide

This section provides the reader with a brief overview of the remainder of the dissertation to follow. The organization is also shown in graphic form in Figure 1.6.

Chapter Two: Terminology and Related Work

Chapter Two provides background information. Since three different fields of research were employed for this work, the first section gives definitions of terminology from each field. The remainder of the chapter discusses the past work to develop methods for operating robots, in particular the work that has utilized multiple robots or that has emphasized the operator through studies or design practices.

Chapter Three: Police Special Weapons and Tactics Teams

Chapter Three begins the description of the research approach with a discussion of the observation step – the ethnographic observation of the SWAT environment. The first three sections describe SWAT teams and basic terminology, introduce the topography and chronology of a SWAT incident, and discuss why SWAT teams are useful analogues for field robot teams. The fourth section describes the uses and methods of ethnographic observation. The observations of the exercises, including specific details and dialogue examples, are described in the final section.

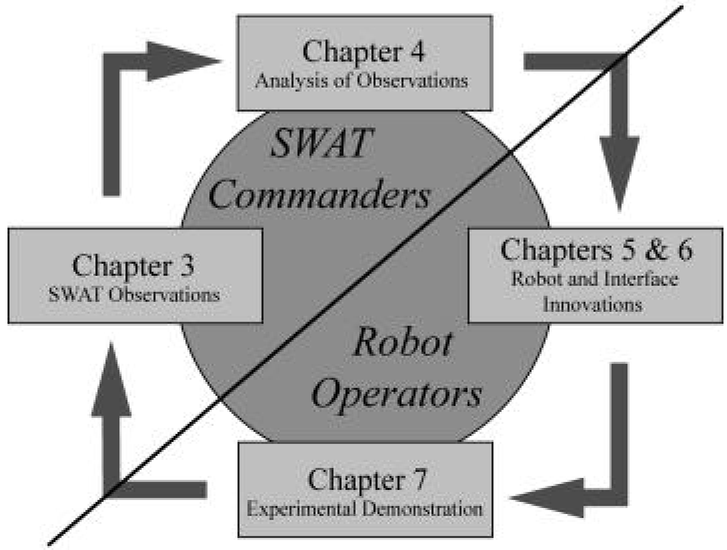

Figure 1.6: **Design Process as Thesis Outline**

Chapter Four: Analysis of SWAT Command

Chapter Four presents the analysis of the observations from Chapter Three. The first two sections discuss the process of analysis and the related work. Next, an analysis of the command of extreme distributed work teams is followed with conclusions about the important roles that the commander plays and the mechanisms he uses to perform these roles. The concluding section looks at what these conclusions mean for technology design, and field robot design in particular.

Chapter Five: Object-Based Task-Level Control of Multiple Robots

Chapter Five describes innovations for design of a new human-robot interaction framework given the analysis presented in Chapter Four. The first section expands upon this analysis as it applies to multiple field robot operation. Then, after a review of the different ways that field robots could be operated, more detail is provided for a robot control architecture that had been used for one or two robots performing a single task – OBTLC. This architecture contains critical qualities required by the analysis, such as the use of physical objects as an operation mechanism, but also introduces some problems when attempting an extension to multiple complex robots. The final two sections describe these problems and the methods devised to address them.

Chapter Six: Direct Manipulation Interface for Field Robots

In Chapter Six, innovation of the computer interface component is addressed in light of the constraints imposed by the operator and field robot components. The chapter describes different interface modalities, particularly for controlling automated systems such as robots. Many different methods that could have been used, but one has been refined for systems of multiple entities – direct manipulation interfaces (DMI). Within direct manipulation, a particular method of interaction has been developed for dynamic systems of virtual robots – a genre of video games known as Real-Time Strategy (RTS). However, there are issues with extending DMI/RTS interfaces for use in the operation of actual robots. The remainder of the chapter describes ways to address these issues.

Chapter Seven: Experimental Demonstration

Chapter Seven describes the experimental system built to show how all of the preceding chapters resulted in a human-robot interaction framework that allowed a single operator to command multiple robots, which also validates the hybrid setting design process. Detail is provided about the various subsystems, their integration, and some of their shortcomings. Demonstrations conducted with the system, which show the utility of an object-based interaction framework for the operation of multiple field robots, are also described.

Chapter Eight: Applications and Future Work

Chapter Eight concludes the dissertation. Applications for this research are suggested, as are areas for future work. For example, future search and rescue efforts are likely to be partly automated and will use many autonomous vehicles to provide comprehensive around-the-clock search capability. These complex systems will require human-robot interaction applications like the one presented here as well as advances in research areas such as autonomous decision-making and cooperative strategies. The research contributions are also reiterated in light of the previous seven chapters.

Chapter Two

Terminology and Related Work

This chapter provides background material to establish a common base of understanding for all readers, as concepts from three diverse fields will be used extensively throughout this dissertation. In the first section of this chapter, the important terms for the three fields – robotics, remote and distributed work, and human-computer interaction -- will be defined as they will be used in this dissertation. The next section will review the previous work that has developed methods for operating field robots, in particular the efforts that have emphasized the role of the operator through design practices or focused studies.

2.1 Terminology

Field Robot

Field robots are uniquely distinguished from other robots by the trait that they operate at a distance from their operator, and typically at a distance from any other humans. The Sojourner rover on the Mars Pathfinder mission that explored the surface of Mars, for example, is a well-known field robot. In this dissertation, the term *field robot* should invoke connotations for the reader of robots in applications such as planetary or underwater exploration, or automated search and rescue operations.

The distance between the field robot and its operator create many issues for system design. Any information about the state of the robot and its environment must come from the first-person perspective of the robot's sensors. Although in some cases a third-person perspective is available via a remote camera or other robots, such a condition is an exception rather than the rule. In addition, on-site adjustments or troubleshooting are impossible, requiring field robots to be significantly more self-sufficient than robots working in a factory or a home.

Operator

The *operator* is the person using the computer interface to send commands to the robot and observe the information sent by the robot. Throughout this dissertation, the pronoun forms *she* and *her* are used generically to refer to the human operator. This dissertation will use the word *user* interchangeably with *operator* for variety, although *operator* seems to provide connotations that are more appropriate. *User* will be solely used in the common term *user-centered*, although *operator-centered* would have the same meaning.

Commanders and Command

In this dissertation, the most common term for the person responsible for the conduct of a humans-only team will be *commander.* The terms *leader*, *manager*, or *director* could also have been used, but *commander* is most appropriate because the individuals in the examples contained in this dissertation are *in command* of the teams. This dissertation will use the pronouns he and him to refer generically to commanders of human teams to be consistent with the gender of the commanders who were studied. *Command*, as defined by the United States Marine Corps, is the exercise of authority and guidance [28]. Because *command* also implies consent by the commanded, it is more appropriate for teams of humans than the term *operation*.

Leaders and Leadership

Much of the existing work for humans-only teams that examines team structure calls the highest person in the hierarchy the team *leader.* Although it will be used only rarely in this dissertation, the term *leader* when used is meant in this hierarchical sense. The act of being a leader, *leadership*, can sometimes assume an esoteric connotation that would be inappropriate in association with inanimate objects such as robots. Consequently, *leadership* will be used sparingly in this dissertation and when used will only refer to the acts of the leader in humans-only teams.

Control

The determination of basic robot motion primitives, from servo commands to joint torque settings, is usually referred to as *low-level control* since it composes the basic building blocks of robot functionality. Once these primitives are abstracted into groups and sequences, using them is called *high-level control*. The result in practice is a spectrum from very low-level ("Send a voltage of +5.0 volts to the left front wheel motor") to very high-level ("Build a Moon base.")

Control vs. Operation

With field robots, the extent to which the operator can and should dictate low-level control functionality is very limited. The high-level control options thus available are more aptly described as *operation* rather than *control*. For a field robot, the robot itself must handle the low-level control. Since the operator does not have direct access to the low-level robot primitives, she does not *control* it. Consequently, the operator should be considered to *operate* the robot while the field robot *controls* itself.

Interface

Although the term *interface* is used in multiple ways in the robotics and computer science communities, to avoid confusion in this dissertation the word will only mean the computer program specifically utilized to exchange information between the robotic system and the operator. This typically is a graphics-based representation on a computer monitor (as it will be in this dissertation), although it could also be a text-based screen on a cellular phone or an audio-only system using voice commands.

Interaction

This dissertation will often use the term *interaction*, and understanding its intended meaning is vital for ultimately grasping the advances of this research. The human-robot *interaction* is not tangible – it encompasses not only the human-computer *interface* described above, but also the framework of methods and actions invoked by the human, computer, and robot and the human operator's mental models of the other components in the system. This broader definition of interaction mirrors the definition and themes developed as an *interaction framework* in the field of Human-Computer Interaction (HCI). Because some robotics concepts are also called *frameworks*, the term *interaction framework* will generally be shortened to *interaction* for the remainder of this dissertation.

Human-Computer/-Robot/-Machine/-System Interaction

The research described in this dissertation does not fall into a well-developed field of human interaction with technology. *Human-Robot Interaction* (HRI) research typically studies the interactions between collocated humans and robots. *Human-Robot-Computer Interaction* (HRCI) similarly focuses on collocated humans, computers, and robots in a group setting. *Human-Machine Interaction* (HMI) research is normally regarding the use of a machine by a collocated human, although supervisory control and the assistance of computers in decision-making are areas of this field that are closely related to this dissertation. *Human-Computer Interaction* (HCI) is a growing and relatively large field that concentrates on the computer as the end object to be operated, not as a mediator in the operation of a physical dynamic system. HCI is nonetheless the human interaction field whose principles are most readily applied to this research. The phrase *Human-System Interaction* would appear to be the most appropriate term for this research, but it has become a catchall term that includes all of the fields of interaction study described above. Based on the terms used in related research [29], this dissertation will use the term *Human-Robot Interaction* or the acronym HRI when necessary to refer comprehensively to the interaction between the human user, the local computer, and, through the computer, the remote robots.

Usability

An interaction's *usability* refers to "the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of a system or component [30]." The treatment of usability as a design concept emerged as a result of the intensive research into and use of more advanced technology during the Second World War with the realization that the adaptation of machines to the human operator increased human-machine reaction, speed, and performance. This dissertation will often use the adjective *usable* to describe an interface or its underlying interaction, with the implication that usability-enhancing concepts heavily influenced its design.

Remote and Distributed Work

From shepherds to astronauts, many types of workers have faced the challenges of performing work at a remote distance from a significant support infrastructure. Distributed workers, such as a multinational product design team, are a more recent phenomenon. However, these two types of workers share many of the same challenges and constraints. Research in *remote and distributed work* has sought to characterize these work environments and propose methods to increase their efficiency. Research in remote and distributed work is informed by a variety of disciplines, combining relevant components of psychology, communication, and organizational behavior to advance the understanding of how people work when they are spatially and/or temporally separated from their co-workers or work resources. With the advent of the Internet, Computer Supported Cooperative Work (CSCW) has emerged as a significant research area that studies how workers can use computers to facilitate cooperation.

Distributed Teams

Distributed teams, as the term is used in the field of organizational behavior, consist of people working together to achieve some commonly defined goal who are not all able to work in the same location at the same time. Thanks to modern telecommunications, some teams are even "virtual" – never meeting face-to-face throughout their existence. Although steps can be taken to emulate collocation, humans actually rely on many cues for cooperation and not all are easily communicated electronically. The communication restrictions and reduction in cues are quite similar to field robot applications, yet research on distributed teams has not included human-robot teams until this dissertation work.

2.2 Related work

This section presents an overview of the areas of active research in multiple-robot systems and then discusses related prior research for single- and multiple-robot systems. The first subsection presents a discussion of the ongoing efforts to increase robot autonomous capability. The remaining subsections present a review of the robotics research that has focused on the role of the operator. Although this dissertation work falls within the subsection describing the operation of multiple robots (Section 2.2.4), the other sections are important for understanding the complete context of the challenges for deploying future multiple-robot, single-operator systems.

2.2.1 Research areas to improve robot autonomy

Many basic robotic challenges exist that must be addressed before deployment of multiple-robot vehicles can become widespread. This subsection provides background on some important topics in which progress must be made in conjunction with the operator issues highlighted by this dissertation work.

Control

The oldest area of research in robotics has been the determination of low-level control algorithms to handle the basic movements of robots. Although simple locomotion and sensing may sound easy to humans, programming these traits into robots has been and continues to be a challenging task. Current research in control seeks to increase the inherent safety and robustness of the control algorithms.

Autonomous path planning

To increase the effectiveness of the human operator's commands, robotics research was conducted to allow the robot to perform some actions on its own. Some of the first successes for such automation were methods for the robot to determine a path from its current state to a goal state. Depending on the type of robot, this path could either be an arm moving through three-dimensional space or a rover traversing an area. Ongoing work in this field seeks to reduce the risks involved in robot motion and the computational time necessary while increasing the complexity of the environment.

Autonomous mission planning

Once robots could reliably handle their own movements locally, roboticists began to focus on allowing the operator to make commands at a higher level than position goal states. Additional autonomous capability enabled mission goals to be pursued by single, and then multiple, robots. This continues to be an active research field as system designers seek the ability to carry out complex missions with rapidly changing constraints and goals.

Fault detection and handling

Because field robots are typically deployed to environments from which recovery is difficult if not impossible, the robots themselves must be able to detect and handle problems when they arise. This fault detection and handling has become an important area of research as overall mission autonomy has enabled longer and more ambitious missions.

Multiple-robot control

Systems of multiple robots have gained popularity as a research topic in recent years. Research of the *control* of multiple robot systems, which typically exhibit emergent or reactive behavior, has only recently moved from simulation work to robotic implementations. Most of the research in the control of multiple robots focuses on the significant challenges of system architecture, communication, and other basic technical hurdles rather than operator involvement. Examples of such systems are:

- AuRA, a hybrid deliberative/reactive robotic architecture for multiple robots. Hybridization arises from the presence of two distinct components: a deliberative or hierarchical planner, based on traditional artificial intelligence techniques; and a reactive controller, based upon schema theory [31].
- ALLIANCE, which focuses on fault tolerance and reliability for a system with multiple cooperating robots. Small- to medium-sized teams in dynamic environments cooperate to perform missions composed of loosely coupled subtasks [32].
- ROBODIS, a web-based dispatching system for autonomous mobile service robots [33].
- The Real and Virtual Environment (RAVE) for running and managing heterogeneous mobile-robot systems also provides capabilities to facilitate development of these systems. RAVE has been used to develop a heterogeneous team of small mobile indoor robots for surveillance tasks, and to command and control a group of two homogeneous outdoor All-Terrain Vehicles. RAVE is primarily a development environment rather than an end-user oriented operational system [34].
- An architecture, proven only on virtual robots, for controlling the task sharing of distributed autonomous robots by having the operator grant simple rewards [35].

2.2.2 Overview of research on methods of robot operation

Current research on the operation of robots falls primarily into two categories: those methods designed for a single robot and those designed for a system of multiple robots. One active research segment known as Human-Robot-Computer Interaction involves both, as it studies the social interactions between co-located humans and robots. A noted researcher in this field is Yuichiro Anzai at Keio University in Japan, who was one of the first to focus on the "problems discovered when we extend the target of research on human-computer interaction from computers, which are essentially passive, to more active machines such as autonomous mobile robots." His goal is to enable humans to behave naturally and cooperate with each other while using multiple robots and computers [36]. Anzai developed the RT-Michele architecture for real-time inter-agent communication between robots, humans, and computers.

Human interaction through supervisory control

When robots were first deployed at a distance from their operators, effectively making them the first field robots, the initial research investigated ways to create a system that could mimic the direct control of a collocated operator and robot – "teleoperation." In the presence of time delays, this method of robot operation can become extremely tedious. Nonetheless, most field robot systems today are controlled using teleoperation.

An extension of teleoperation made possible through advances in robot autonomy, supervisory control is gaining importance in field robotics as the robots move beyond the laboratory and into common use. Supervisory control, a standard concept in machine automation, is defined as a situation where one or more human operators are continually exchanging information with a computer that interconnects through artificial effectors and sensors to the controlled process [1]. There are three common interaction metaphors for supervisory control of field robots: control panels, virtual dashboards, and virtual reality environments.

Control panel

A typical control panel display such as the one for the NASA Fido rover shown in Figure 2.1 gives the user the impression that she is controlling the robot as she might a machine. Dials, gauges, and data readouts make up the bulk of the information shown to the operator. This interaction is usually the simplest to create, but the most difficult for new users to learn.

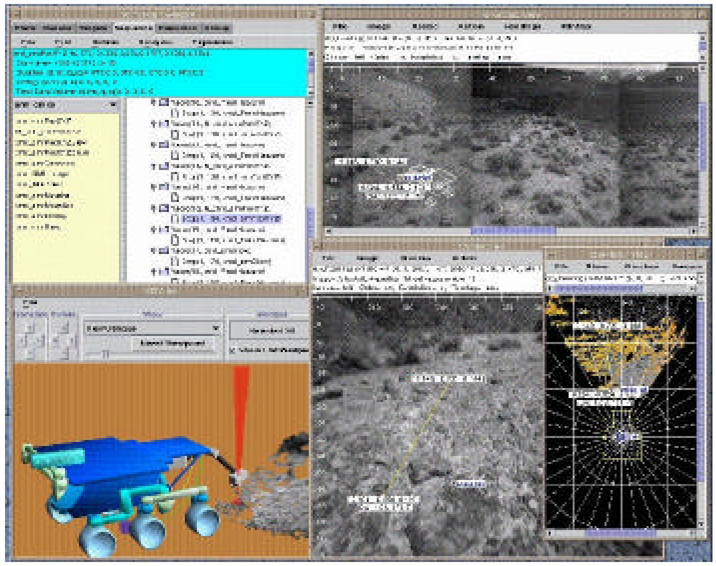

Figure 2.1: **Example Control Panel Interface**

The operator interface for the NASA Fido robot, the Web Interface for TeleScience (WITS), shows the various robot sensor outputs through dedicated windows, including one for a graphical representation of the robot pose and the returns from a laser rangefinder.

Users send commands to the robots through dial changes, button selections, and command text entries.

Dashboard

The dashboard creates the impression that the user is driving the robot by providing a first-person view from the robot and a set of dials and gauges much like the dashboard in a car. Users operate the robot through extensions of the driving metaphor such as control wheels, buttons, and sliding levers such as in the interface for the MLB Bat unmanned aerial vehicle pictured in Figure 2.2. This interaction is not difficult to create and is recognizable by users, but there is no clear way to scale dashboards to systems with more than one robot.

Virtual reality

With modern graphics hardware and software, a representation of the robot's world is created on a computer and displayed for the user in a lifelike format. The best systems give the user the impression that the user is collocated with the robot. This interface is difficult to create but is readily understood by users. However, virtual reality is better suited for a display of information than an ongoing interaction with an actual dynamic system since current input

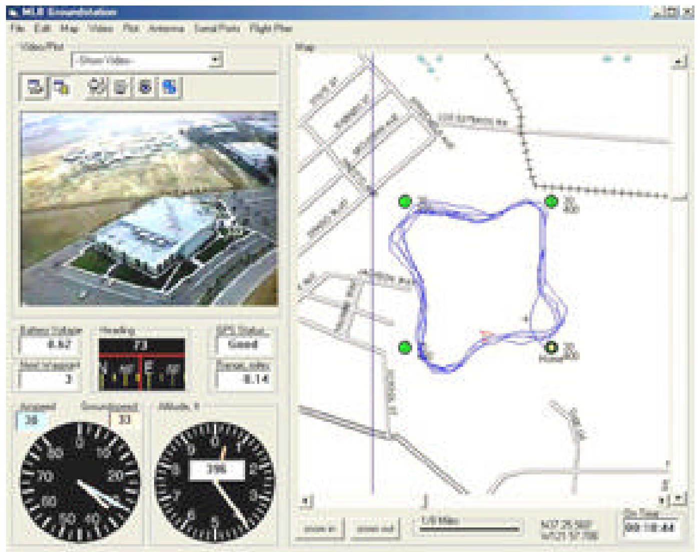

Figure 2.2: **Example Dashboard Interface**

The MLB Bat unmanned aerial vehicle operator is shown the typical airplane gauges (heading, airspeed, altitude) and a forward-looking video as if she were in the airplane.

mechanisms, such as virtual object manipulation, are constricting and difficult to master. The Virtual Environment Visual Interface, a virtual reality environment used for operating rovers and underwater vehicles [37], is shown in Figure 2.3.

Each of these three common interface methods has significant advantages for certain applications, yet none of them appears to extend readily to robust operation of multiple robot systems. Control panels are most useful as engineering interfaces to single robots, dashboards present only one particular perspective at a time, and virtual reality has difficulty when the “realities” sent by the robots’ sensors are contradictory.

2.2.3 Operating single robots

Although though a large amount of previous work exists in Human-Robot Interaction, the subset of HRI research that is similar to the research described in this dissertation is small when constrained to research focused on:

- Field robots or unmanned vehicles,
- Multiple robots in operation simultaneously, and
- User-centered development of the human-system interaction.

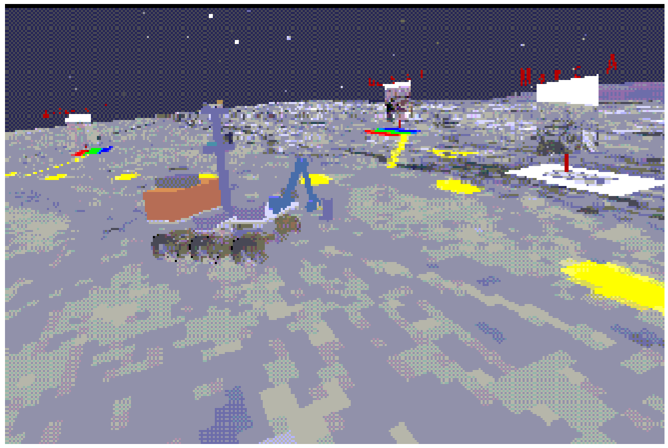

Figure 2.3: **Example Virtual Reality Interface**

Data collected by a Mars rover – in particular, sample locations and imagery – is presented to the remote operator in this virtual reality interface.

This section and the next (for single robots and multiple robots, respectively) describe the related research that addresses some or all of these issues. In each section, a review of relevant work with robots other than field robots will be followed by discussions of the virtual and experimental field robot research. At the end of Section 2.2.4, the work most related to this dissertation is presented.

A wide variety of human-system interfaces for single robots exist. Autonomous helicopters have been controlled using point-and-click [38] and virtual dashboard [39] techniques; autonomous underwater vehicles and space vehicles have been directed using virtual environments [37] and high-level tasking [40, 41]; intelligent arms have been instructed using gestures [27] and graphical icons [42]; and many robots have simply been fully teleoperated [1].

Robots other than field robots

Since the largest segment of HRI research is not conducted with field robots, much of it is not directly applicable to this thesis. However, the following single robot systems discussed in this subsection are worthy of mention.

Manipulator arms have been operated using high-level object-based point and direction telerobotics [27] and a graphical robot programming language designed to aid handicapped individuals in work settings called RoboGlyph [42]. Both of these studies took the needs and perspective of the end user very seriously during system development, and although usability studies were not carried out, these robots should be considered as contemporary models of robot arm usability. However, the environments for most manipulator arms differ from those of field robots.

The MOVAID robot, designed for residential care of disabled and elderly, was developed with user constraints in mind. As with the interaction developed through this dissertation work, the MOVAID GUI guarantees that only allowable actions can be requested of the robot, but for MOVAID this filtering comes from logic in the GUI itself, not the robot [43].

A few robotic systems built for collocated human-robot interaction utilize a natural language dialogue. For instance, the Royal Institute of Technology in Sweden has developed a fetch-and-carry robot to assist physically impaired people in an office environment. The robot responds to natural language dialogues, and the dialogues are augmented by a GUI that affords commands in an imperative, sentence-like structure with buttons and drop-down selection elements so that the design features of the speech interface and the GUI are consistent with respect to each other. They have also conducted some user studies that have necessitated interesting changes to the system, including the addition of a small doll for visual feedback [44]. This and other dialogue-based interactions for robots are typically for household robots and primarily concerned with explicit robot motions.

Field robots

Field robots, as a particular type of robot, have not been the subject of much HRI research, although the distance between robot and human should not diminish the importance of the human-robot interaction. In fact, the distance could be considered to increase the research challenge because it eliminates many operational cues standard in most HRI scenarios. Instead, as stated by a field robotics researcher, "The operational human-system interfaces for these vehicles have typically evolved from interfaces that were originally developed to provide insight into vehicle operations for the engineers who were designing, developing, and testing the vehicles and their onboard autonomy [39]." During the literature search for this dissertation, no ethnographic fieldwork or other significant study of the user perspective for operational field robots could be found.

Figure 2.4: **Example Unmanned Vehicle Control Station**

This operator station for the CIRPAS unmanned aerial vehicle is typical of current systems: one operator on the left pilots the aircraft, while the operator on the right controls the payload.

Currently deployed field robots provide evidence that supports this lack of usability focus during development of the operator interface. An instructor of unmanned aerial vehicle command and control for the United States Army said "the interfaces that we have do not appear to have been designed with a discussion with the operators. There are many aspects of them that don't make sense from an operator's perspective [45]." The interfaces developed for these current systems, such as the one for the CIRPAS unmanned aerial vehicle shown in Figure 2.4, require at least two operators – one as a pilot of the vehicle and another as the payload operator, with all coordination between these two tasks taking place via human teamwork rather than onboard the vehicle or as part of the human-robot interface.

The DARPA TMR program at Georgia Tech did expand their MissionLab development environment to accommodate formal usability testing. By recording user actions during pre-mission planning, they could use them to refine the design interface itself. The only published experiment in this program thus far used one robot to explore a room down a hall from the operator [46] and did not include a user study. Apparently, the results of the usability tests have not yet been fully analyzed or incorporated into subsequent systems.

In research similar to some aspects of this dissertation work, an operator at Wright State controlled a mobile robot via a dialogue that utilized a language lexicon that served as a minimal spanning semantic set for single-robot two-dimensional navigational tasks. The task command lexicon consisted of a verb, destination, direction and a speed [47]. In contrast with this dissertation research, no interactions with objects were specified, no discussions of capability took place, and only a single robot was considered.

The concept of Object-Based Task-Level Control (OBTLC) developed at the Aerospace Robotics Laboratory at Stanford University provides fundamental technologies that enable a unique method of field robot operation. OBTLC has been used to operate underwater vehicles, space robots, flexible manipulators, and manufacturing work cells. OBTLC promotes the idea of a human-robot 'team' that is created by designing very capable robots that can manipulate objects in their environment with only high-level commands from the operator. However, no OBTLC implementation has moved beyond a single-task, single-object scenario with two robots, so its scalability to complex deployments has been unproven.

2.2.4 Operating multiple robots

Comparable research to this dissertation work – concentrating on the *operation* of multiple robots by a human at a high level – has been very rare. Indeed, such research is scarce even for single robots, yet there are a few reasons to move the research focus beyond single robots to a problem of higher complexity:

- User interface shortcomings in single robots can often be masked by increased user workload or the addition of more operators; single robot operation can be generally accomplished by various methods of teleoperation or supervisory control,
- Operating many robots simultaneously should put additional unique stresses on the operator that are not likely with a single robot,
- The research would be more related to an active research field (multiple robots), and
- Human operator issues of any sort are rarely addressed by the current research with multiple robots, yet the expected demand for such research is high.

A very thorough and oft-cited review of cooperative mobile robotics makes no mention of user experience, utility or usability of these systems when giving a summary of the major, yet tractable, challenges for the near future [48]. This may be interpreted as a sort of myopia in robotics that focuses on the raw technical challenges – a standard condition in young technical fields – rather than a sign that the operation of multiple robots is not a significant unsolved problem.

As discussed in Section 2.2.2, if one operator were expected to command multiple robots, no HRI systems previously developed for single robots appears to address the problem of this additional complexity. Direct teleoperation, for instance, would either overstress the operator or underutilize the robots [49]. Extensions of the robot interfaces of more automated systems, such as control panel or dashboard metaphors, for single robots do not appear to naturally accommodate multiple robots.

Robots other than field robots

The research on multiple robot systems that has incorporated a human operator has largely concentrated on methods of cooperative motion and task planning for surveillance and exploration, with the user utilized either for initial planning or assistance during operation. For example, a two-component interface from Japan allows for either monitoring or operation of each robot in a multiple-robot scenario; the selection must always be one or the other for each robot, selected via a virtual control panel with switches [50].

To address a nuclear power plant servicing application, the ACTRESS system was created to allow operators and robots to cooperate on a task. The nuclear plant has a human operator, and the robots that service the plant have a different operator – and the two operators talk about what needs to be done. The ACTRESS researchers have been examining the performance of the cooperative work with human operators and multiple robots. The human-robot interface is a web page that is mostly set up for teleoperation of individual robots, although higher-level motion commands are possible. This project, when it reaches the experimental stage, will benefit from being able to test out ideas on a model plant [51].

Virtual field robots

Of the field robotics research that has addressed the operator experience during the operation of multiple robots, most of the work has only utilized virtual robots, likely due to the cost of field robot hardware development and the late state at which most operator-related design is conducted. One of the leading military programs is the “playbook” system developed by Honeywell to control multiple unmanned aircraft, which has only been tested in simulations thus far. This GUI allows the operator to stipulate plans to various levels of abstraction, allowing the remainder to be fleshed out by an automated planner [52].

Gage [53] looked at the command and control of a system of more than 100 robots for military-type missions. He investigated ways to control the movement and positioning of the robot group as a whole, rather than the movement of individual agents. Each individual agent's motion is based on the motions of the other agents, and is most strongly influenced by its nearest neighbors. A coverage behavior maintained a spatial relationship which adapted to local conditions to optimize some function, such as the detection rate/range of targets or the probability of undetected enemy penetration. Gage looks at two ways to control group movement. The first is to bias the motion of each agent in the desired direction. The second is to directly control the movement of a small number of agents and let the others follow due to their coverage behavior.

In recognition of current shortcomings of Unmanned Tactical Aircraft (UTA) Command and Control concepts for multiple vehicles, a private company was tasked by DARPA to design, prototype, and evaluate an intelligent, decision-aiding software system for the command and control of multiple UTAs by a single operator. This project was based on an expert systems research, not robotics. The result uses a plan-goal graph and was never implemented on a robot [54].

Heterogeneous robots

Systems with multiple heterogeneous robots are slightly more difficult to operate than homogeneous robots, since the operator has to differentiate between robot types and their corresponding capabilities. In the course of this research, no work examining the operation of non-virtual heterogeneous was found.

MokSAF provides a multi-agent environment that enables the investigation of the roles that different agents play with a heterogeneous team of humans and software agents. In particular, MokSAF is useful for creating military movements of these heterogeneous units. The main problems addressed were the development of routes for all units that satisfy given goals and constraints. The MokSAF capabilities were tested by twenty-five three-person teams to determine which type of route-planning agent was most successful, not to study the interaction between humans and robots [55].

DARPA SIMNET allows hundreds of soldiers to train together in a virtual world. In addition, the virtual environment also contains a large number of autonomous vehicles coordinated by an operator at a single workstation. The operator has not been the focus of this component of SIMNET, but rather the routines developed for maneuvering these many vehicles among obstacles and moving vehicles [56].

Only one research project addresses the issues surrounding the use of spoken dialogues with many heterogeneous robots. However, CommandTalk, a spoken-language interface to synthetic forces in entity-based battlefield simulations [57], is only intended for use with virtual robotic vehicles.

Field robots

Research implemented on an actual field robot so that real experiments can be conducted requires a significant time and resources investment. As a result, such projects are few in number.

The MAGIC2 system, developed for operational control of unmanned air vehicles, combines control panels for the control of unmanned aerial vehicles but appears to be limited to a maximum of four vehicles per operator [58].

The Basic Unexploded ordinance Gathering System (BUGS) system is composed of five small robots that are given basic instructions by the operator and coordinated paths by a central planning unit. This system performed better than a similar system that relied on a random search pattern by the robots [59].

The MACTA hybrid agent/reactive architecture, using a computer-based Reflective Agent, utilizes predictive planning to provide a supervisory control user interface to two cooperating robots implementing the Behavioral Synthesis Architecture. These robots work together to move objects, but the experimental object appears to be singular and static. MACTA focuses on "behavior scripts" and their ability to satisfy human-designated goals [60].

Raytheon built a multi-modal human-robot interface consisting of a ruggedized computer, see-through video display glasses, an instrumented glove, a vibrotactile display on the forearm, a thumb-mouse, a bone-conduction microphone, and a wireless network transceiver. This system was used to control three disparate robots. The user interface was static, and the system tasks were all movement-oriented [61].

The CyberScout uses a group of two mobile and four stationary sentries capable of cooperating for reconnaissance and surveillance. These robots use computer vision and a distributed agent-based software framework called CyberARIES to operate without the aid of an operator [62].

Multiple field robots interacting with/manipulating objects

One of the most difficult tasks in robotics is the manipulation of objects. There has been extremely limited research specifically on the operation of multiple field robots interacting with objects, mostly because such robots are difficult to develop and deploy. The human operator issues are not necessarily exposed in the related research on two-robot cooperative manipulation of objects, since there has been little difference in task specification from that used with single robots.

Neither does there appear to be much multiple robots research to develop an interaction paradigm that affords action with objects in the environment. Instead, the main body of research, as described above, has concentrated on reconnaissance, exploration, or other coordinated movement. The only differing work found was an extension of the Object-Based Task-Level Control architecture developed at the Stanford Aerospace Robotics Laboratory presented by Dickson [63]. His AUTOMAN (AUtonomous robot-Team Object MANipulation) architecture facilitated the autonomous manipulation of objects using a team of two free-flying robots; this research incorporated a single task, a single object, and two robots that were both tasked automatically on every run. While other OBTLC research has established that operation centered on the objects in the environment is an extremely effective method of robot control [64], and Dickson extended the capabilities to two-robot systems, no one outside of the ARL has adopted this control architecture.

User-centered design of experimental field robots

Very few projects utilized user-centered design concepts for field robot design and were able to implement the effort on real robots.

In the early stages of this research, the experience of the user was highlighted as an important facet of field robot system design through the design of a task-based interface for the operation of the ARL autonomous helicopter [38]. The most relevant other research effort, conducted by Terry Fong at the Carnegie Mellon Robotics Institute, allows humans and multiple robots to collaborate on robot motion through spoken and electronic dialogues.

The dialogues are adapted according to the user, using a user model with three user stereotypes -- novice, scientist, and expert. The experiment, although with multiple robots, focuses on manual teleoperation. The operator is presented with the most urgent task that needs assistance [65].

The most quantitative-oriented study of the user experience during the operation of multiple robots is the doctoral dissertation of Khaled Ali at Georgia Tech. His appears to be the first such experiment to study usability and other human factors in field robot operation. Ali ran over 100 people through tests that measure the safety, effectiveness, and ease-of-use of operational paradigms that vary the amount of automation and the group size. However, Ali defines all types of multiple-robot tasks as some type of movement, which is fundamentally different from the approach taken by this dissertation's research. In fact, although Ali suggests that an object-oriented interaction such as the one discussed in the following chapters "probably makes the robot group easier to control," he says that it is less "conventional." The robot group sizes and methods of automation chosen are also substantially different from this research [66].

2.2.5 Human-Automation Interaction

A large body of work has been assembled on the proper design of the interaction between humans and complex machine automation. An excellent review is available in the book Human Factors (1997) in the article by Parasuraman, "Humans and Automation: Use, Misuse, Disuse, and Abuse [67]." Other research has determined a high positive correlation between operators' trust in and use of automation, and an inverse relationship between trust and monitoring diligence [68]. A study within the unique environment of field robotics, where operators do not often have much choice in the level of automation possible and have only limited cues to establish trust, would provide an interesting supplement to the Muir and Moray findings.

Chapter Three

Police Special Weapons and Tactics Teams

This chapter is the first of five chapters that describe the research approach shown in Figure 3.1. As discussed in Chapter One, the initial step in the design of an interaction for multiple robot operation is to gather data through observation. Since no operators yet exist for these systems, this research sought an analogue setting to provide appropriate data – police Special Weapons and Tactics (SWAT) teams. The objective of this chapter is to introduce the terminology, structure, and operating environment of SWAT teams and give an overview of the observation data gathered through the course of this research.

The first of the five sections provides a basic definition of a SWAT team and introduces SWAT terminology. The second section describes the layout of an incident site and a typical incident timeline. This background information leads to a discussion in the next section of why SWAT teams are appropriate analogs for teams of field robots. This argument is particularly important since the observations and analyses of the SWAT team environment, given in this chapter and the next, provide the foundation for the human-robot interaction design in the subsequent chapters. Four field training exercises and one actual SWAT incident were observed to understand better the SWAT environment and learn how the SWAT team works effectively. Such observations are a type of ethnographic field study, a formal research technique whose background and methods are described in the fourth

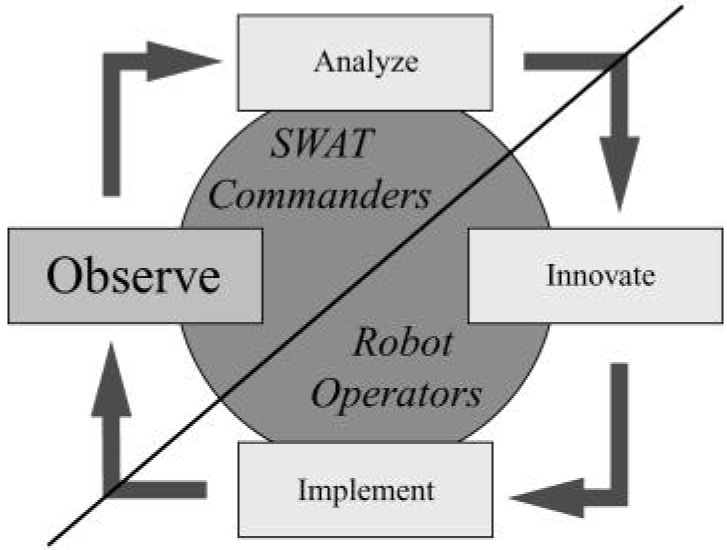

Figure 3.1: **Step One – Observe**

section. The final section gives an account of the SWAT observations, including comprehensive descriptions of the events that took place.

3.1 SWAT background information

Police Special Weapons and Tactics (SWAT) teams are responsible for handling high-risk tactical situations involving barricaded suspects, hostage situations, and the serving of warrants. The primary goal of SWAT, as stated in a SWAT training manual, is the "successful conclusion of high-risk situations through the use of specially equipped and highly trained personnel without injury or loss of life to citizens, suspects, or police officers [69]." The chief components of SWAT teams are assault/entry teams, a Hostage Negotiation Team (HNT), sniper teams, and the command structure.

There are two types of SWAT deployments: planned and unplanned. A planned deployment, such as a warrant service that might prove dangerous, is preceded by an extensive amount of intelligence gathering and planning. Unplanned deployments, which occur when hostages are taken or a suspect otherwise creates a situation that calls for a SWAT presence, do not usually have the luxury of time to gather intelligence information or plan. In such cases, the SWAT leadership tries to take steps that will delay suspects so that more planning can take place with a greater amount of information.

3.1.1 Terminology

Suspect

The person or persons responsible for the presence of the SWAT team are categorically called the suspect or suspects. This is true even when this person is not believed to have yet performed a crime, such as when someone is armed but secluded and unresponsive. The suspect's safety is a very high priority unless the SWAT team members feel that the suspect may physically harm someone else.

Teams

Hostage Negotiator Team (HNT)

The Hostage Negotiator Team is made up of police officers and psychologists who have been trained to end SWAT incidents through negotiation with the suspects. The HNT usually has a "throw phone" that can be taken to the suspect that provides a secure and dependable line of communication between the HNT and the suspect. The SWAT command structure tries to give the Hostage Negotiator Team as much time as it needs to end the incident peacefully. The HNT also helps the assault teams determine the location of the suspect and hostages prior to an assault.

Sniper team

Certain members of the SWAT team are trained and outfitted as snipers. As such, they deploy individually to useful vantage points around the incident site. The snipers are given a general idea of where to be and then find an appropriate location based on previous experience. The snipers are often out of sight of one another and consequently communicate via a dedicated radio frequency called the SniperNet. They have high-powered rifles with telescope sighting that they also use to monitor closely and report all movements they observe. One of the tactical dispatchers is responsible for communications between the snipers and the command structure, and this dispatcher documents all SniperNet activity.

Assault teams

The remainder of the SWAT team officers comprises the assault team or teams. The nature and number of assault teams depends on the tactical commander's assessment of the situation. In most cases, an Emergency Response Team (ERT) of three or four members is immediately sent at the start of the incident to the most likely point of exit for hostages or suspects. Other members may make up a scout team of four to ten officers to augment the early information gathering of the snipers. One to three assault teams were created in each of the observed SWAT exercises and incident. Each team had a leader that was the primary point of contact for the command structure, although the other members sometimes provided or required information directly.

Command structure

At an incident site, the command structure consists of an incident commander, the Hostage Negotiator Team leader, the logistics supervisor, and the Tactical Commander (TC). The incident commander is ultimately responsible for everything related to the incident. He is typically a police lieutenant with experience as a SWAT team member. The incident commander makes the final decisions regarding assault launches, significant negotiation concessions, interaction with the press, and coordination with other police and public entities. The Hostage Negotiator Team leader and the logistics supervisors, as their titles suggest, lead the HNT and maintain sufficient logistics for the team to operate, respectively. The Tactical Commander was the focus of my research observations. The TC's role is to coordinate the activities of the tactical teams – assault, emergency response, sniper, and perimeter control – throughout the exercise. Further information about the roles of the Tactical Commander will be provided in the sections to follow.

Tactical dispatchers are a non-leadership component of the command structure who manage the communication for the SWAT team and record important data. The tactical dispatchers create maps according to intelligence gathered by snipers, scout teams, and interviews from local residents or building managers. They also compile all known information and log all radio traffic. These maps and logs are kept on large sheets of paper that are taped in prominent locations at the command post. There is one main dispatcher who handles much of the regular communication between the SWAT command structure and any deployed team members. Another dispatcher communicates with the snipers and logs everything they report. A third dispatcher keeps a running log of all radio traffic and significant events. Any other tactical dispatchers who are present might be tasked to keep maps updated or seek additional information from other police or public sources (such as DMV records or criminal histories) and add that to the posted information.

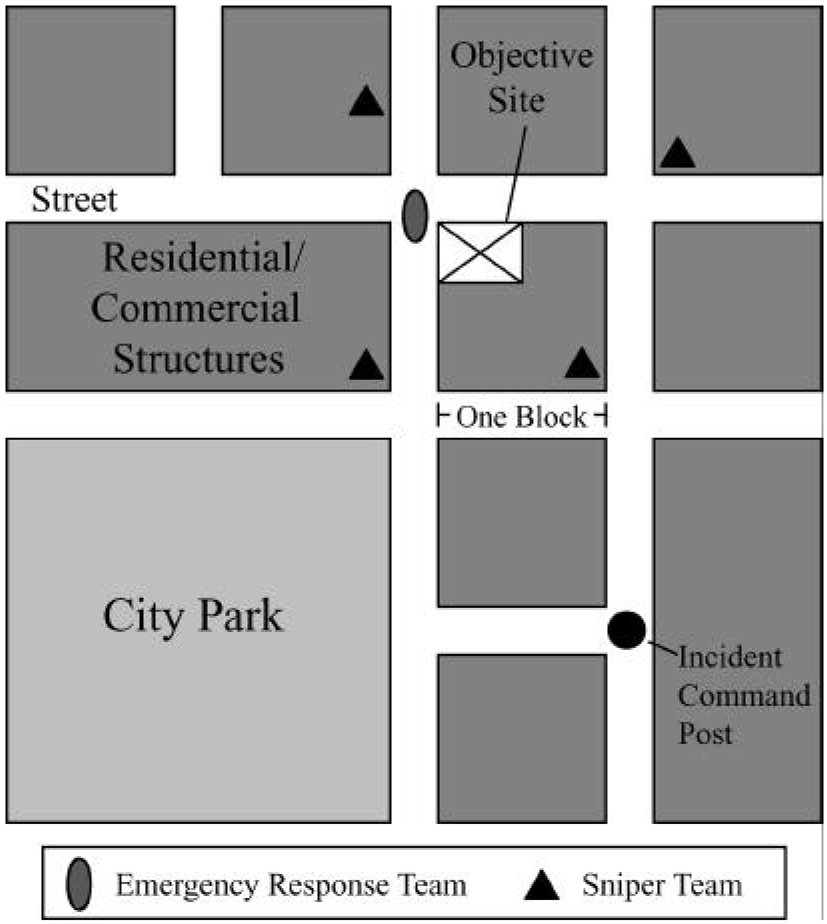

Figure 3.2: **Example Incident Topography**

The objective site is usually guarded by an Emergency Response Team and surrounded by sniper teams. The Incident Command Post is located at a safe distance out of the line of possible weapons fire.

3.2 SWAT incidents

3.2.1 Topography

A generic diagram for a SWAT incident is shown in Figure 3.2. The location of the suspect or suspects is called the Objective Site. The incident commander establishes an Incident Command Post (ICP) at some distance from the Objective Site, located to provide a safe working environment for the command structure that can be reached quickly by foot from the Objective Site. Typically, there is a physical distance of a few blocks between the

ICP (and, consequently, the Tactical Commander) and the incident site (and tactical teams). The TC thus does not normally have a direct line of sight to the team or even the site itself. The Tactical Commanders relied on radio (voice only) technology to gather information from and convey commands to the tactical teams.

Hardcopy city planning maps are kept in each patrol car, so a map of the neighborhood is immediately available to the SWAT team. These maps have the most exact info on the scene available – street locations and widths, buildings, open lots, and routes of access. Other information from the HNT and other intelligence sources, such as auxiliary maps and building layout descriptions, may be photocopied and distributed to the appropriate team members. Additionally, the tactical team usually makes its own maps on paper and in chalk on the roadway to ensure that all members are familiar with the environment and to incorporate their specialized assault information.

In each of the observations, the focal point was one particular building. The SWAT team names the sides of the buildings so that they can be more easily referenced later to determine position inside and outside of the building. The customary system is to name the front side (where the main entrance is) the '1 side', and then proceed clockwise around the building (2 Left, 3 Back, 4 Right). This simple system can be a source of confusion, however, as the buildings in each exercise either seemed to have more than one "front" or had some other nonrectangular features. The direction of the front side must then be discussed in terms of cardinal directions, which can be a source of confusion itself given the peculiar nature of San Francisco Bay Area street naming convention. Consequently, in each exercise the TC used a map to make the side numbers explicit, saying in Exercise 4, for instance, "This is going to be the number 4 side here," and so forth. Nevertheless, the importance of a simple and accessible system for building description such as this outweighed the cost of any initial confusion.

3.2.2 Chronology

As shown in Figure 3.3, the SWAT incident proceeds from the initial scouting process to the suspect's arrest. This figure only includes the action of the SWAT team – it does not include the alleged crime, initial response by uniformed officers, or other activities in which the SWAT team does not actively participate.

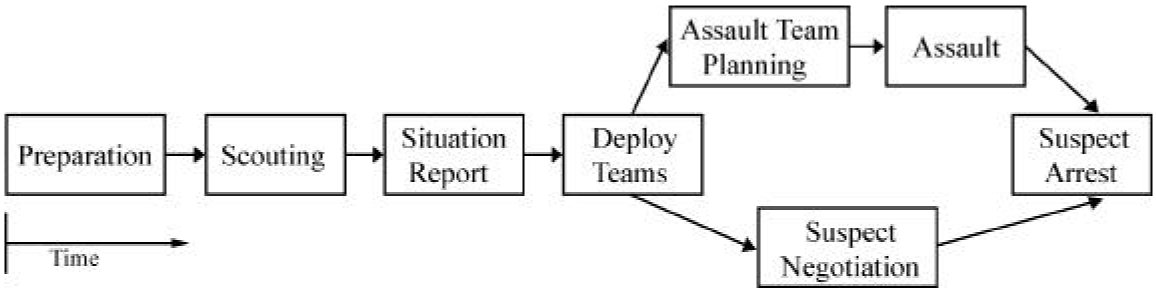

Figure 3.3: **Typical Incident Chronology**

After scouting the objective site and briefing the team via the Situation Report, the SWAT team's efforts split between planning for an assault and trying to resolve the incident peacefully.

As the exercises begin, the SWAT tactical team assembles for an informal radio communication check. There is no clear leader during this process, although the Tactical Commander participates. All officers check their radios and make sure that they have the correct channels for the assault team and for the sniper network. Weapons status is then triple-checked: unloaded or dummy weapons for exercises and loaded and safetied weapons for an incident.

The Tactical Commander and team leaders then meet face-to-face at the edge of the ICP. The TC asks the team leaders for recommendations for two to four scouts, who are immediately sent out to get an idea of the location of doors, windows, and other access areas.

After these scouts return and other intelligence has been gathered from the HNT and the tactical dispatchers, the Tactical Commander briefs the SWAT members not already deployed (such as the Emergency Response Team) through a Situation Report. A picture of a Situation Report in progress, with the TC in the center facing the camera, is shown in Figure 3.4. The first part of the briefing is a rundown of the situation specifics – the number and description of suspects, reports of weapons, descriptions of the area and buildings, and any constraints on their operation. The TC, with input from the squad members who were scouts, decides on a course of action. The commander then specifies the roles that each officer will perform during the action. In particular, the TC names who will be on the sniper teams, who will continue as scouts, and who will be sent as the Emergency Response Team (ERT). He then dispatches the teams out to the objective site according to the practiced standard procedures for the initial phase of an incident. Any deviations from standard

Figure 3.4: **Situation Report**

The Tactical Commander, without headgear and facing the camera in the center of the picture, briefs the SWAT assault and scout teams on the latest incident information.

procedures, such as team size or off-limits areas, are explained to each element as it is sent out. Immediately thereafter is a face-to-face meeting of only team leaders and team members, as the TC returns to the ICP to assume his remote command role.

As the scouts, snipers, and ERT deploy to the objective site, the assault team members begin to plan an assault. These officers move nearby to draw up plans of the assault in chalk on the street and practice the assault coordination and timing. Meanwhile, the Hostage Negotiator Team tries to convince the suspect to end the situation peacefully. If the team is successful, the suspect will leave the building and usually will be taken by the Emergency Response Team. If the HNT is not successful, the incident commander will tell the Tactical Commander to initiate the assault, and the plan produced by the assault team will go into effect. The assault is devised to ultimately end in the suspect's arrest with all participants safe and unharmed.

3.3 Why SWAT?

Studies of SWAT teams provide an excellent place to start the development of interactions for field robot operation. There are a number of reasons why SWAT tactical commanders are sufficient analogues to use as surrogate robot operators. A few significant differences between field robotics and SWAT also limit the analogy.

3.3.1 Reasons why SWAT serves as useful field robot analogy

There are five reasons why police SWAT teams serve as a useful analogy for research on field robot operation.

Nature of mission

The basic natures of the SWAT and field robots missions are similar. Some task must be completed in a given timeframe by a group of entities doing the work, while one remotely situated person is ultimately responsible for a successful outcome.

Commander sensory cues

The most important aspect of the SWAT environment with respect to this research is that the Tactical Commander must affect change at a remote location without any direct sensory information. The TC must construct a physically-consistent mental model of the objective site from reports by a distributed network of scouts, snipers, and assault teams, and use that model to form relevant commands. He must also see to it that the best possible end result is obtained while integrating external information and managing the conduct of many separate entities to avoid dangerous situations.

Distributed agents

With many obstructions commonly separating the team members from one another as well as the commander, the SWAT team must function as a distributed team of the same nature as a system with multiple field robots. There will be times when the team members will be collocated, when they will be near enough to sense the same objects but not one another, and too far away for any sort of direct contact. The SWAT team members must coordinate their motions well enough not to endanger one another, but do not work so close as to require coordination of all their activities.

Physical objects

The SWAT team, in contrast to many distributed teams, does work by moving around within a shared physical workspace. There are objects in the space that are vital to the successful completion of the mission: doors, windows, hallways, furniture, suspects, and hostages.

Hierarchical and peer-to-peer command

Although the commander provides many directives to the SWAT assault teams, they also work out a great deal of local coordination themselves. They make many of the low-level decisions that the commander would be unable to do given his limited sensory perception of the objective site.

3.3.2 Reasons why SWAT serves as a good setting for ethnographic observation

SWAT team exercises were an excellent setting for ethnographic observation for field robot operation for two reasons.

Access to complete environment

As a research setting, SWAT teams were very useful. The researchers were able to walk throughout the entire exercise area, from the Incident Command Post to the assault team and sniper positions to the suspect and hostage location, allowing them to determine the true state of affairs and then compare that to the perceptions of the commanders. The ability to make this comparison and watch which factors contributed to deviations was the source of many important insights that would not have likely been discovered otherwise.

Potential application arena

The original motivation for contacting the SWAT leaders was due to the possibility that the SWAT environment might be a good potential application area for robotics. Through the course of the observations and analysis for this research, the first of their kind in the SWAT field, the data mostly contradicted this perception.

Short-term

Many robotics researchers cite the SWAT environment as a 'dangerous' place where robots would be welcome. This research determined that they would be welcome, but will have to undergo many significant advances before anything other than teleoperated machines will be allowed to participate in the SWAT assaults. However, autonomous robots may be useful in the short term as inexpensive sources of surveillance and communication relays, especially small unmanned aerial vehicles.

Long-term

SWAT environments truly are dangerous, and if the human officers could be replaced by robots while maintaining the same level of safety and professionalism, they would likely be welcomed. This research generated useful data as a first step in understanding some of the qualities that such robots must have [70].

3.3.3 Limitations of SWAT as useful field robot analogy

Some aspects of the SWAT environment that limit its applicability to field robotics.

People are not as capable of robots and robots are not as capable as people

Robots and people may be difficult to contrast in some respects; robots would be able to assume some functions of the SWAT team members given current technology. However, far more traits distinguish the two. Given the current state of robotics, two of the most noticeable differences are the SWAT teams' use of voice to communicate and their lack of global positioning data.

Field robots would not use voice to communicate

There are two significant reasons why the use of commands based on voice are not a good model for commanding robots. First, and perhaps most obvious, the current and near-term state of voice recognition software has not shown itself to be a particularly attractive interface method for computer control. Since many companies are working on this technology, positive changes should be forthcoming in a few years. The second reason, however, is unlikely to change. Voice, as a method of communication, is a particularly low bandwidth mechanism. It is difficult for people to talk and to listen at the same time, and the bit rate for voice communication is very slow. Field robots, with the telecommunications infrastructure they are likely to have on board, should be able to take advantage of their ability to communicate at a much faster rate. One possible compromise for this constraint is for a computer to act as an intermediary, communicating with high bandwidth to the robot and communicating using voice with the operator.

SWAT does not utilize global position sensing

One of the greatest technical advances for robotics in recent years has been the availability of global position information using the Global Positioning System (GPS) while outdoors. While the SWAT team members could carry a GPS sensor with wireless communication to a central base, the team does not utilize any sort of positioning system other than frequent updates to the Tactical Commander of their position relative to the objective site. With global position data, the Incident Command Post might be able to display the locations of all units in real-time. In fact, one TC did say that he wished they had such a system, but they cannot justify its cost for the marginal expected utility. A field robot system, on the other hand, can be expected to have positioning information, either using GPS or a similar indoor-based method, that will be regularly communicated to its operator. The effect of this information on the conduct of the Tactical Commander role is unknown. As a final note, the fact that the SWAT team operates effectively based only on relative position information should be seriously considered as a useful data point for designers of multiple field robot systems.

Cannot comprehensively capture all SWAT data

Although the researchers were able to move freely throughout the SWAT exercises, they were unable to place additional sensors to monitor important events. They could take a few snapshots with a still camera, but the team leadership was uncomfortable with video recording. The researchers also made audiotapes of segments of two exercises, but the quality of these tapes is not high due to ambient noise. The team leadership was somewhat reluctant to allow these tapes and would not allow the principals to wear microphones. Also, the researchers had to observe quietly and discreetly, so advanced coordination between researchers elsewhere in the exercise would not have been welcome. Finally, the researchers were only able to observe whatever happened during the exercise, rather than propose situations and observe the effect on the team. An observation environment with more complete control might have produced results that were more comprehensive.

Optimality of SWAT teams for command and control is unproven

As a final issue on the use of SWAT as a model for the operation of multiple field robots, evidence of the optimality of SWAT team leadership mechanisms, or even that the SWAT team conducts itself in an intelligent way, does not exist. These are very valid reasons for skepticism. However, determining the optimal method of command seems impossible and at least has not been done so far despite extensive analysis. In addition, the members of the SWAT team put their lives in the hands of the SWAT team leaders and one another. The command methods that have developed after years of trial and error should be considered, at the least, a very useful starting point for field robot operation.

3.3.4 Alternative ethnography settings

Other ethnography settings were also considered but were not as attractive as SWAT teams. Future work could consider these other options, which may provide significantly different lessons than the ones learned through this research. Football and other sports' coaches are readily accessible commanders of multiple-entity teams, but they do not command remotely nor is the environment of set plays, huddles, and competition consistent with field robotics. Orchestra conductors have developed interesting mechanisms for coordination of multiple heterogeneous entities over time, but do not command remotely or wield significant re-planning powers while concerts are underway. Managers of distributed project teams, the typical setting for geographically distributed work research, were attractive because of the existing body of knowledge, yet their teams and the teamwork setting are usually too virtual to be applicable to the physical objects manipulated by field robots. Finally, military commanders were seriously considered due to the ample documentation of their procedures and the large extent that field robots are likely to play in future military operations. However, gaining access to military leaders is difficult, and the scale, scope, and geography of their typical missions would have been a difficult setting for ethnography.

3.4 Fundamentals of ethnographic observation

In the 1980's, there was a refocusing of interest on the part of technology system designers away from the view that technology supported individual tasks and toward the view that human activities were carried out, for the most part, in cooperation with other humans. Consequently, new technologies were designed to support the cooperative nature of most human activities. A new field evolved called CSCW (computer-supported cooperative work), which was concerned with the design of computer tools for the support of group work [71]. As a consequence of this shift in focus, there was a realization that the methods most often used to analyze users' need and activities, and to evaluate designs, were not sufficient. Looking at individual cognitive processes and evaluating the fit between isolated tasks and technologies would not provide the perspective needed to design and evaluate technologies for group work. Ethnographic methods were adopted for understanding group work practices and for linking this understanding to design. Blomberg et al's *Ethnographic Field Methods* [72] is an excellent resource for additional information about the usefulness of ethnography in technology design.

3.4.1 Background

Motivation for using an ethnographic approach

The ethnographic approach, with its emphasis on "natives' point of view," holism, and natural settings, provides a unique perspective to understanding work practices. As practiced by most ethnographers, developing this understanding requires a period of fieldwork where the ethnographer becomes immersed in the activities of the people studied. Typical field work involves some combination of observation, informal interviews, and participation in the community when possible [73]. Designers of technology, for the most part, have been interested in understanding human behavior to the extent that it enables them to design products better suited to the needs of the users. Designers, therefore, tend to spend more time testing and evaluating their designs and less on understanding the underlying behavior. By adopting ethnographic methods, designers get an alternative methodology that gives them access to people's common practices.

Guiding principles of ethnography

There are four main principles of ethnography: natural settings, holism, descriptiveness, and focus on members' points of view [72]. 'Natural settings' refers to a commitment to study people in their true settings, not reproduced in a laboratory or other experimental setting. Holism emphasizes the concept that behaviors take place in a broad context whose removal could change those behaviors in important, nontrivial ways. Descriptiveness, contrasted with prescriptiveness, highlights that ethnographers seek to describe how people *actually* behave, not how they *ought* to behave. Finally, with the realization that one can never see the world exactly as another does, research methods should be aimed at getting as close to an insider's view as possible. This often manifests itself with the use of terms and concepts that have meaning to the study subjects rather than the researcher.

3.4.2 Methods and techniques

There are three primary methods for conducting ethnographic fieldwork: observation, interviewing, and video analysis. Since observation was the primary method used in this research, it will be the only method given a more thorough explanation. Formal interviews were not used because they would have likely yielded excessively subjective information, and videos of the SWAT team in action were not allowed.

Observation

Because ethnographers are interested in understanding human activity in the settings in which it occurs, most ethnographic investigations involve some period of observation. The ability to observe and record ongoing activities becomes critical to the success of the research.

There is a well-known axiom in anthropology that what people say and what they do are not the same. This is one of the principal motivations for including observations of ongoing activity in any study of human behavior. The distinction between what people say and what they do is related to the distinction between ideal and manifest behavior. Ideal behavior is what every "good" member of a society should do, whereas manifest behavior is what people actually will do. Asking people about their behavior will often yield responses closer to the ideal than to the manifest. Yet, the manifest behavior is what must be addressed in design. The distinction between what people do and what they say is also related to the fact that people often don't have access to the unarticulated knowledge associated with some behaviors. In these cases, the proper vocabulary with which they could provide a description is not available.

Roles

The spectrum of observation roles is best described by examining the two extremes, the passive observer and the participant observer. The passive observer seeks to be a "fly on the wall," being as unobtrusive as possible. Maintaining this role is often difficult and requires some culturally acceptable role that allows the observer to "hang around." The participant observer, meanwhile, observes while being a full participant in the activities of the group. An advantage to this technique is that the researcher gains firsthand experience of the events that are under study. In addition, such a role is often the only way to gain access to certain societies. However, the disadvantages to being a participant observer range from the hassles of logistics to the constraint of only seeing the society from one particular vantage point.

Given these factors, a continuum of observer status naturally develops. Often an observer will move back and forth throughout the spectrum as the situation requires. In any case, it is imperative that the ethnographer maintains good social relations with the people observed, and an adaptable approach to observation usually proves useful.

Focus of observation

Once the decision to observe is made, there are still many questions to answer, such as what to observe, when to observe, where to observe, and at what point the observations are to be concluded [74]. The answers depend on the research direction, and can range from a particular person for one month continuously to a large group observed periodically over many years. Although there is no fixed rule determining the end of observations, the most accepted rule of thumb is that when the researcher is no longer surprised by what is being observed, she has probably seen enough.

Note taking

Best practices for note taking depend largely on the individual, but it is nonetheless one of the most important links between the field experiences and how those experiences are interpreted later [75]. Different situations will require different techniques, but some attributes are universal: clearly noting the difference between paraphrases and direct quotes, distinguishing between facts and conjecture, and as much information about date, time, place, and other setting as can be considered appropriate.

Interviews

Observations are seldom able to stand alone and are often coupled with informal interviews or discussions. The interviews provide a mechanism to clarify points that were not clear during observation. When observations and interviews are intermixed in time, the observations can lead to well-refined questions and may confirm or contradict information gleaned from interviews.

Comparison of ethnography to other user-centered design methods

The principles of ethnographic methods can also be described through a contrast with the traditional design approaches to understanding user needs: customer surveys, operability assessments, focus groups, and field trips. The primary area of difference is in context – while ethnography seeks the user in her natural setting, customer surveys, operability assessments, and focus groups put the user into testing scenarios that may significantly change user behavior. The traditional approaches also require the designer to present the user with a set of questions or measurements that may create results based on prescriptive, rather than descriptive, traits that describe the technology instead of the work. Finally, none of these methods encourage the user to be a collaborator in the design, whereas ethnography has the potential of providing a context for mutual understanding between user and designer.

3.5 SWAT observations

SWAT exercises are complex dynamic environments. One could learn quite a bit from any of the various participant subgroups. However, over the course of the first observation, it was concluded that the most appropriate and useful relationship to study for application in field robotics was that between the tactical field commander at the tactical command post and the distributed tactical (scout, emergency response, assault, and sniper) squads.

The primary observation method was to monitor the conversations between the tactical commander and the teams, which mostly were conducted via the radio but were made face-to-face in some cases. In addition, the training environment allowed the researchers to observe the true state of the scenario by moving throughout it. This was very useful to compare with the commander's apparent perception of the situation. Discrepancies between actual world state and perceived world state are a significant issue in field robot deployments – affecting the accuracy of the operator's situational awareness – and the SWAT methods for handling this issue would be very instructive. The commander was not usually interviewed during the exercise, but a short time (from one day to one week) later with little apparent loss in accuracy.

One difficult aspect of the observation process was the small knowledge base of the experimenters with respect to SWAT operations, techniques, and experience. It is possible that some significant amount of communication took place at a level imperceptible to the researchers. Also, results would have likely differed had the researcher been a participant in the exercise rather than an observer. A field researcher with a stronger SWAT background might have focused on other events and derived different results.

3.5.1 Exercises as useful and accurate study settings

Field exercises were chosen for observation rather than actual SWAT missions for two reasons. First, exercise administrators go through extensive efforts to replicate the events and activities likely to be encountered in an actual mission. Participants in the exercises are active-duty officers who perform the roles they would in an actual mission and whose lives depend on their readiness for these missions. Therefore, one could expect high fidelity between SWAT exercises and actual incidents. Second, due to the high level of physical danger in most SWAT missions, it is unlikely that researchers would have been allowed access to more than a small portion of the SWAT incident scene, much less the completely free movement that the exercises provided.

On one day when a SWAT exercise was planned, an actual incident occurred not far from the exercise site. The researchers were able to observe the actions of the SWAT team from inside the police perimeter. These observations completely confirmed the reasoning given above – the incident, from all observable measures, appeared to take place exactly as the exercises did, and access to the various aspects was so limited for safety reasons that observations of actual incidents would not have provided nearly as much insight into SWAT operations as the exercises provided.

3.5.2 Description of exercises

The leaders of the Palo Alto-Mountain View Regional SWAT Team gave permission for a small team of researchers to observe the field training exercises. The team members were given orange vests to signal that they were observers of the exercise and thus were able to move freely through the incident area. This provided an opportunity to observe the evolving state of the exercise as well as view the events from various participants' perspectives. The team was consistently introduced as university researchers there to observe how SWAT teams functioned and not to analyze the performance of the team. This introduction created goodwill from the SWAT members towards the researchers.

The SWAT team allowed this observation as part of a continuing effort to participate and assist in the local community. The Palo Alto police department has a longstanding positive relationship with Stanford University, although cooperation towards published academic papers such as this is rare. This effort was seen as a way to extend the relationship in a new direction.

For SWAT field exercises, the primary objective is to execute realistic but controlled scenarios. The nature and location of the incident to be enacted is unknown beforehand to most of the team. To avoid causing unnecessary alarm, the exercises are typically held in relatively remote commercial locations on weekend days, although residential areas also are used if sufficient notice can be given. The team observed had an average of forty members present at the exercises, with around fifteen in the Incident Command Post and the rest deployed to the objective site. The pace of operations is typically measured and slow, although this pace can accelerate significantly.

The primary method for data collection was the observation of four field exercises. Researchers paid particular attention to the behaviors and communications of the Tactical Commander at the command post, although some observations were conducted at the Objective Site. This enabled comparisons of the actual situation with the TC's perception of the situation. Throughout the exercise, researchers used sporadic and informal interviewing as necessary to clarify salient points or fill in background details. Extensive field notes describe the TC's behaviors throughout the exercise, as well as some ancillary notes on the behaviors and conversations between members of the tactical teams, and significant events in the exercise. The tactical teams in the four field exercises were led by three different Tactical Commanders. Where possible, researchers captured artifacts either by drawing them or photographing them at the scene. Audio recordings of parts of two exercises from the Incident Command Post were made.

To become familiar with police and SWAT procedures before attending the exercises, one researcher participated in two regular patrol 'ride-alongs' with SWAT team members. Through informal discussion, the functions of the SWAT team, terminology, and other background information was learned. The ride-alongs also provided time to convey the purpose of the study and to build the trust necessary to gain access to the training exercises.

Nonetheless, there was still a great deal about the SWAT environment that was unknown to the researchers when the first exercise began. Consequently, the observer role chosen was that of the known, acceptably incompetent observer who passively observed and learned and did not participate. This role seemed to remain appropriate throughout the study. The predominant method of study was observation, with only sporadic and informal interviewing as necessary to clarify salient points or fill in background details.

Notes were taken by hand to avoid the distracting sounds of a laptop keyboard. Direct quotes were marked in the notes and checked against the audiotape later when possible; if no audio record existed, the quotes were considered to be paraphrased and marked as such. Immediately after the exercise, the handwritten notes were transferred to a computer and additional information recalled from memory was added. Analysis of the notes and follow-up interviews for fact checking and clarification took place over the following week.

There were no observed conflicts with participants at any point in the study. A measure of restraint was used when moving about the incident area to avoid distracting or disrupting the exercise. There was some distrust of having an outsider visibly recording the conduct of the officers, but the stature of the officers who had made the introduction generally quelled any uncertainty. Some SWAT members took special pains to volunteer further explanation throughout the exercise. The researchers were subject to the same security checks required of all exercise participants.

3.5.3 Observation data

For each exercise, an overview will be provided that gives the basic details, particularly the type of incident, its setting, and any significant events. More detailed information about the exercise, selected from the field notes on the basis of uniqueness and clarity, will then be given. The complete field notes are not included in this dissertation.

Exercise One

In the first exercise, the scenario started as a hostage situation at a private residence and then moved to a hospital clinic. The SWAT team was called to a residential neighborhood where a hostage situation was underway. Just before the assault was to begin, the administration revealed that the hostage and suspect had left the house before the SWAT team had arrived – according to the participants, an annoying but authentic scenario indicative of the realism of the exercises. The hostage had been taken to a hospital for treatment, and the suspect had taken additional hostages at a hospital clinic. The SWAT team moved immediately to the hospital site and redeployed the units. The team searched through the hospital building and had to overcome potentially dangerous (but unintentional) misinformation about the floor plan of the building. The assault team ultimately conducted the assault into a clinic office.

Details

Upon arrival, an Incident Command Post was set up at a street location around a curve from the incident site. The TC conducted the Situation Report nearby. Neighbors assisted assault team leaders in drawing a floor plan of the residence in chalk on the street. The assault team made plans to enter the residence. Just before assault the began, the team leadership got a report of disturbance at a hospital nearby and determined that it was the suspect and hostage who had somehow gotten out before SWAT arrived. Everyone moved to the hospital site.

The hospital administrators had maps of the different buildings and the complex, which the assault team leaders, tactical commander, and incident commander used to plot general strategy before sending the teams out. The building with the hostage had a basement, so the teams started their search there and worked their way up; meanwhile, the Hostage Negotiator Team reached the suspect by telephone.

Once the team knew that the suspect and hostage were on the third floor, they went to the pile of maps that the hospital administrator had brought and pulled out the one that said "Third Floor". One researcher had been sitting near the pile and saw that there were actually many buildings in that pile, and they had pulled out a map for the wrong building. The building name was in small print on the floor plan, but it is unclear if they would have been able to detect their mistake anyway since the names weren't familiar to the team. The researcher did not say anything to the participants but told the exercise administrators at the first opportunity. In the map the team was using, there was a hall around the outside of the building with access doors to the two stairwells in the middle of opposite sides. The assault team leader came back, looked at the map, and formed a plan with the tactical commander based on this map. The plan was to have two teams coordinate and time their entry into each door at the same time, splitting up once they were in the hallway and moving quickly towards one another down the two halls doing a room-to-room search. There was some indication from the suspect that there was a hall down the middle, but that contradiction with their information didn't register with anyone. In reality, the building had one hall down the middle connecting the two doors the assault teams would use, and there was no hallway around the outside. When the assault began, the two teams jumped inside and were each surprised to see the other team facing them down at the other end of a long hall. There was significant confusion among the teams and at the ICC when the reports came back in. The teams adapted quickly and improvised from their initial plans to conduct room searches on each side of the hallway until they eventually converged on the suspect. The commanders pointed out in the debrief after the conclusion of the exercise that there was a significant opportunity for lethal crossfire and other dangers, particularly if the urgency surrounding that assault had been greater.

Exercise Two

A workplace hostage incident was the scenario for the second exercise, held at a school supply distribution warehouse. The large, cluttered environment of the warehouse limited the usefulness of hand signals and consequently increased the assault team dependence on the Tactical Commander as the central node for coordination. This exercise provided some of the clearest dialogues between the Tactical Commander and the various team members. Quotes from these dialogues are provided in the next chapter along with analysis of their content.

Exercise Three

A workplace hostage situation was again the scenario for the third exercise, held at an empty software company office building. Three disgruntled workers who had recently been laid off had returned and taken coworkers and managers hostage. The building had a symmetrical 'L' shape that was a source of confusion among the team and the commander. At one point, an intended diversion almost backfired when this symmetry led two groups to the same stairwell.

Details

The two-story building was shaped like an L, with the front door entrance in the crook of the L accessed from a courtyard. Three staircases inside – one main one near front door, others in each wing near the end access door – would become important features of the SWAT exercise to follow.

Midway through the exercise, the tactical commander waited for word from the HNT that the suspects were ready to make an exchange of hostages for water. He was informed that the deadline for delivery of water to the suspects was approaching – with one minute left. When queried by the TC, the assault team leader said he had two minutes before being ready to secure the hostages and deliver the water. The TC immediately responded, "Make it

30 seconds." The assault team leader led the team to exchange the hostages less than a minute after this command was given.

Towards the end of the exercise, the Tactical Commander was looking at the map to figure out how the team could get up to the 2nd floor where the hostages appeared to be. He met face-to-face with the Incident Commander (IC) (also an experienced SWAT commander) to plan possible diversions and discussed these plans with the assault team leader deployed in the field.

> TC: Where is the team deployed?
>
> ATL: Mainly in #2 stairwell, team of 4 separated.
>
> TC: Can you leave two at the throw phone and take the rest of the team to the #3 stairwell?
>
> ATL: Do you want us to move to the #3 stairs?
>
> TC: Yeah. When you can get there (#3 stairs) and you're ready to move on the (#3) stairwell, let me know and move into the (#3) stairwell.

The tactical commander then briefed the Hostage Negotiator Team:

> TC: The rest of the team is going to move to this (#3) stairwell. OK, and we're going to go up and take the landing and stage right there on top of it. That's the stairwell on the 3 side. We're gonna go right up there onto the 2nd floor and just clear out from there. Even while they're negotiating. We don't know where the suspect is.
>
> HNT leader: How many (officers) are going through there?
>
> TC: Whatever I have left. I mean the entire team. There's, I believe eight. (Eight was in fact correct.)

When queried by the researchers, the TC said that everyone but the phone team would be moving to go up the #3 stairs (a different stairwell than where the hostages were) and start clearing the second floor.

Just as the assault team was about to go up the stairs, a hostage appeared at the bottom of their stairwell. The hostages and suspects were actually in the opposite stairwell than where the TC and everyone else thought they were. The phone had been taken to the stairwell in the other wing, with no suspects there to pick it up, and the entire assault team was poised at the bottom of the stairwell where the suspect expected only a minimal force to be delivering the phone. If events had taken place with just a few seconds difference, this would have probably resulted in considerable confusion on the part of the assault team if not outright danger for the hostages and the team. Ultimately, though, the situation was handled by the assault team without the suspects realizing what was going on – they moved quickly and quietly to the other stairwell and sent two members back with the phone.

Exercise Four

The fourth exercise observed was at the same location at as the third, started an hour after the conclusion of that exercise. In this instance, the entire SWAT team – commanders and assault team members – had complete knowledge of the building beforehand. Although they took the same precautions in this exercise as with any other, the exercise ended much more quickly than any of the others.

Details

The report from officers on the scene was that a husband took his wife and several others hostage one shot reportedly fired. The TC sent the snipers to take positions, which they did with very little instruction this time since they knew the building well.

One escaped hostage said that the remaining hostages and suspects were in the "Hercules Room" on the second floor. All hostages and suspects were in this room. Everyone in the exercise already knew the location, access routes, and size of the Hercules Room based on their experience from the previous exercise.

The TC conducted a standard Situation Report, where the team made simple plans to clear the floor and then proceed with an assault assuming that the location of the suspect would not change. Few instructions were given.

Incident

An actual SWAT incident was observed when a planned exercise was preempted by an emergency SWAT deployment. While the SWAT team was preparing for the exercise, an incident had coincidentally begun just a few blocks away in a medium-density residential neighborhood. A deranged woman had barricaded herself into her bedroom and no one was sure if she was armed or not or if her children were with her in the room. She had a criminal history involving gun possession, so the SWAT team was called in. One assault team had to break a window to get the HNT throw phone into the room, but she refused to use it. However, the HNT was able to confirm that she was alone. Just as another assault team was about to use chemicals to immobilize her, she came out of the room peacefully of her own accord.

Details

The SWAT team formed up on the edge of a park and school playground. An elementary school was very near the incident site. The Tactical Command Post was set up under a tree on the street at the edge of the park. Also at the site were two large police vans from Mountain View and the SWAT van, as well as a few police cruisers and a fire engine. This was slightly more equipment and personnel than was observed at the SWAT exercises. In all other observable measures, the incident was conducted just as the exercises had been.

The researchers were moved away from the command post for safety and security reasons. They were not able to stand in close proximity to the command post to hear discussions among the commanders and dispatchers for the remainder of the incident. They were given a radio tuned to the correct frequency to allow them to hear the radio traffic between the command post and the deployed teams. There was also some intra-team communication. Relying on the radios definitely reduced the understanding of some of the communication since the transmissions were not always clear.

Chapter Four

Analysis of SWAT Command

This chapter provides an analysis of the observations of the commanders of police Special Weapons and Tactics (SWAT) teams, and the ramifications of this analysis for field robot interaction design. This analysis step is the second in the design process outlined in the Introduction chapter and shown in Figure 4.1. As discussed in the previous chapter, the leader of the SWAT tactical teams, known as the Tactical Commander, provides an excellent model for the operator of multiple field robots. This individual is ultimately responsible for the success or failure of a mission that must be accomplished through the coordinated efforts of agents who operate outside the commander's direct perception.

The main goal of this chapter is to introduce the roles and command methods of the Tactical Commander, which were the primary results of an ethnographic analysis. This analysis will be used in Chapters Five and Six to generate innovations for a human-robot interaction for operating multiple robots. The first of the four sections of this chapter provides background about the process of analyzing such field observations, an area with less formal foundation than the observation process described in the previous chapter. In the second section, an overview of related work provides insights into the role of commanders in high-pressure, dangerous settings, the role of commanders in distributed work teams, and ways in which work teams compensate for distance when intense collaboration is required. In the next section, an analysis is performed to extend this body of research to the command of extreme distributed work teams. The analysis reveals two important roles that such a

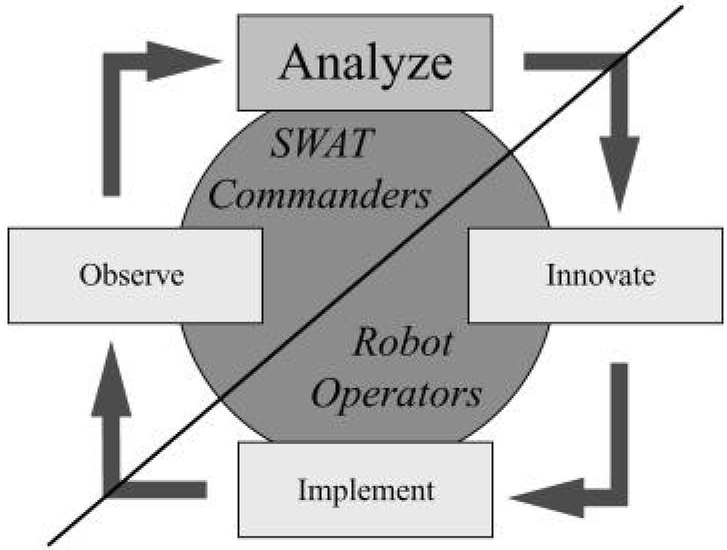

Figure 4.1: **Step Two – Analyze**

commander plays – cultivating common ground and coordinating action – and a mechanism he uses to perform these roles – referencing of physical objects in the environment. In the concluding section, the ramifications of this analysis for field robot system design are examined. In particular, the human-robot interaction needs to support the operator's command roles and the robots must be capable of sensing and working with objects.

4.1 Analysis of ethnographic field observations

Three design steps remain after the observations have been made (Figure 4.1): the observations need to be analyzed, and then this analysis must be transformed into design innovations and an implementation. Unfortunately, the amount of previous work documenting the transition from observation to analysis to innovation is not very extensive. These steps are easy to conceptualize, but there is little in the way of concrete examples of their use, much less a solid theoretical foundation. One attempt to address this issue, a formalized observation process devised by Beyer and Holtzblatt that ultimately results in design practices [76], has not been widely adopted and appears to be excessively constrictive in the methods used for the study.

One approach to the analysis step, utilized in this research, relies on related previous work to provide a framework with which to make sense of the observations. In this approach, preexisting research efforts are sought that have developed theory that may be applicable to the setting and observation results. This theory is then used to extract useful concepts from the data that can then be passed on to the innovation step. One of the effects of using this analysis process is that the ultimate design goals can be considered when reviewing the related work and when formulating the new analysis concepts, which can result in a faster overall analysis time and increased utility of limited data. If the objective of the approach had been to create new theory, there would have been a requirement for more structure in the data observation process. This alternative analysis approach requires more observation data and can yield a broader theoretical contribution, but is not focused on informing design.

4.2 Related work

To provide the framework for this analysis, the most applicable theory comes from outside the field of robotics. Previous research in remote and distributed work, and Computer-Supported Cooperative Work (CSCW) in particular, developed concepts applicable for the use of technology when the resources available are spatially distributed. Robotics has looked mostly at the effects of this distribution on robot engineering choices (e.g., coping with significant time delay), not at how it might be understood and manipulated to better accommodate the operator. This section discusses the prior geographically distributed work and CSCW research related to this application to robotics.

Distributed teams

Over the last several years, an increasing amount of research in Computer-Supported Cooperative Work has focused on geographically distributed work. More is now known about how people coordinate activities with distant colleagues and how technology supports distant work [77-79].

Distance typically decreases the likelihood of communication and collaboration [80, 81] in part because distance and reliance on communication technologies makes it more difficult to develop and maintain common ground [82-84]. People cannot easily point to objects of interest, receive feedback on how their message was received, or repair misunderstandings [83]. Maintaining a shared awareness of the work environment and of ongoing activities is also particularly difficult for distributed team members [85, 86]. For example, Cramton [86] reported that team members had difficulty accurately interpreting events at distant sites because they did not have critical contextual information.

One way that distributed teams can successfully overcome the lack of proximity and impoverished communications is to strive for more loosely-coupled systems [84, 87]. Grinter et al [88] observe some benefits of increasing the modularity of the work so that constant interaction and information sharing between distant team members is not required. In complex systems, particularly those susceptible to disaster, constant communication is crucial for coordination and sensemaking [89, 90]. However, because communication is more difficult on distributed teams, the process of sensemaking can be more challenging [86]. Although little research has been conducted on the role of the leader in distributed teams, this dissertation suggests that the leader can play a crucial role in ensuring that team members have similar information from which to make sense of their shared reality. This is consistent with Weisband's [91] findings that leaders of more successful distributed teams actively shared "other awareness" information – information about what others in the team are doing.

Extreme distributed work teams

Extreme teams, which work under hazardous conditions, are most often associated with the military. Leadership of such teams has been studied extensively, both in the sense of leader as source of inspiration [92] and leader as organizer [93], as have the effects of leaders on the teams themselves [94].

Little research has examined work teams that come together for a single event, are highly interdependent, and whose performance can save or cost lives. Jones and Hinds call these intense teams operating under hazardous conditions "extreme work teams [95]." Although extreme work teams can be collocated or distributed, the interest for this research is in extreme work teams that have the added challenge of coordinating work without a shared physical space – extreme distributed work teams.

Spatial distribution in extreme work teams is particularly important because of the challenges introduced for collaboration. Lack of contextual information, reduced communication, and less common ground have the potential to severely limit the ability of extreme work teams to coordinate interdependent activities. As mentioned above, teams may seek to overcome this limitation through loose coupling of their efforts.

However, some geographically distributed work does not lend itself to loose coupling [88]. SWAT teams are an example in which there is strong, unavoidable interdependence. Weick's [90] study of firefighters in the Mann Gulch disaster illustrates the importance of coordination on these teams and the high cost of failure (13 lives lost) when coordination breaks down. Because loose coupling is not a viable option for these teams and the cost of errors is high, they must develop ways to coordinate effectively at a distance.

This research contends that the commander's role in sharing awareness information will be especially important in extreme distributed work teams because there is not time for all members to communicate and receive feedback from one another and because errors and misinterpretations can be fatal.

4.3 Command of extreme distributed work teams: Commander roles

This section provides the results of the analysis of the observation data described in the previous chapter. Across the four exercises, two key roles of the tactical commander – cultivating common ground and coordinating the actions of the distributed team – were observed.

4.3.1 Cultivating common ground

The commanders of the SWAT teams cultivated common ground that they subsequently relied on to coordinate the distributed work of their teams. Kraut et al [83], Olson & Olson [84], and others have emphasized the importance of common ground for many aspects of distributed work. In every exercise observed, the SWAT teams cultivated common ground through training, meeting face-to-face before dispersing to the objective site, and re-calibrating during the incident by establishing relative locations of common objects in the environment. The face-to-face meeting at the start of the crisis and the re-calibration process were driven by the Tactical Commander.

Establishing common ground through training

One way to establish common ground, particularly on distributed teams, is to provide common training [79]. Although there were no observations of SWAT training other than the exercises, it was clear that training in standard vocabulary, communication procedure, and a set of standard procedures for discreet actions (e.g. how to respond when a hostage is released) were common across members of the teams. In the exercises observed, the Tactical Commander frequently mobilized action by invoking standard procedures in which members of the teams had been trained.

Calibrating common ground

SWAT training recognizes that the spectrum of possible scenarios is so broad that the team members can only be given basic building blocks for action, and then the team members must be adaptable to a wide variety of situations. When a call for the SWAT team goes out, the members have no idea what aspects of their training will be needed or what actions will be necessary. In all four exercises, the Tactical Commanders dealt with this uncertainty by meeting with the teams face-to-face before dispersing to the objective site. They referred to this meeting as the Situation Report, discussed in more detail with an accompanying picture in Chapter Three. The Tactical Commanders consistently used these meetings to provide their tactical teams with the established information and reinforced specific elements of their training.

As a means of understanding the situation, the SWAT command structure obtained street maps and building floor plans from city documents and landlord records. The TC's used these documents as exhibits during the Situation Report, consistently referencing them with phrases such as "the door here" and "clear out through here."

In Exercise 3, for example, the TC had maps of the building that he lifted high so that the entire team could see. Using his free hand to reference the map, he described the situation and provided the immediate plan:

> ERT (Emergency Response Team) is moving into the breach point – the front door. The suspect and hostages are reported to be in the Hercules Room, just off the stairwell on the 2nd floor. Make a rapid clear through the bottom floor but on the top floor, move carefully. We're gonna have to leave people (officers) here (staircase) just in case people (hostages or suspects) come down it.

While saying that the "ERT (is) moving into breach point – the front door" and that the "Hercules room (is) just off stairwell on 2nd floor," the TC referred to maps of both floors that he held up so that everyone could see.

The commander also used this face-to-face meeting to gather opinions and reach consensus for some aspects of the mission. He asked the team "Should assault team move up to lock down the front door?" and was answered negatively. He then asked if the team was "Not opposed to moving in through another door?" to which they also answered negatively.

Re-calibrating common ground

The most challenging component of cultivating common ground took place after the Situation Report ended and the team moved into their spatially distributed positions. Team members began to coordinate with each other by referring to common objects. However, because they had different perspectives on these objects after dispersing, there was frequently some confusion. The commander played an important role in reconciling team members' different perspectives.

Utilization of objects

The SWAT participants consistently used physical objects and the location of those objects as the foundation for their dialogues, as in the following examples from Exercise 2, when the TC needed to ascertain the status of doors leading into the warehouse (object use in **boldface**):

> TC: Red, what is your access to **the right garage door**?
>
> OR: I cannot see **a garage door** from here.
>
> TC: Move further down **the fence** until you can see the garage doors.
>
> OR: OK. I can see **the garage door**.
>
> TC: Can you verify that **the right garage door** is open?
>
> OR: I only see **one garage door**. Are there **two garage doors**?
>
> TR: Black sees **two garage doors**, and thinks **the right one** is open but can't be sure because of **the car**. Can you confirm?
>
> OR: I see that… oh, now I see there are **two doors** but they look like one from here. Yes, **the right door** is open just a little.

and when the TC provided an updated plan to the assault team in Exercise 3:

> The rest of the team is going to move to **this stairwell**. OK and we're going to go up and take **the landing** and stage right there on top of it. That's **the stairwell on the 3 side**. We're gonna go right up there onto the 2nd floor and just clear out from there.

The use of objects often went further than just providing the relevant information for the status or plans of team members, becoming a mechanism for determining the spatial relationship between team members, as in this example in Exercise 2:

```
TC: Green, tell me where you are right now.

OG: Coming through the ditch outside on the 4 side.

TC: Tell me when you can see the office door.

OG: I can see the office door anywhere along this ditch.

TC: Where are you in the ditch right now?  Do you see
Black near the fence?  How far are you from him?

OG: I don't see Black but I see White near the fence.  I
am 30 yards from White, in the ditch.

TC: (Knows that Black and White are together) OK.
```

This technique was observed in the incident as well, to determine the relative locations of two different teams:

```
ATL1: We’re in the interior.  Several attempts at voice
and knocking and no response.

TC: Confirming team at bedroom door?

ATL1: Affirmative.  Bedroom door is closed.  Rest of the
house is clear.

TC: To team 2: Can you advise which corner suspect is in?

ATL2: Confirming the 2-3 corner. (This refers to the back
left corner as seen from the front door.)
```

Merging models and translation of perspectives

To determine their spatial relationship relative to other team members, members of the tactical teams frequently contacted the TC, who typically had the most complete and comprehensive understanding of the incident. However, the researchers noticed a significant burden that the TC was thus required to shoulder throughout the exercise – he needed to keep track of the perspective and knowledge base of each agent under his command. For example, during Exercise 3 a sniper reported:

```
White male with handgun, 2nd floor, opening 10.
```

which the TC translated without pause into a pertinent description in the frame of reference of the assault team leader:

```
(Sniper) 01 has got visual on one of the suspects on Side
2 in window above your breach point with at least one gun.
```

Task models

The TC was not only interested in the physical location of the units but also in their capabilities for action. For instance, in this dialogue from Exercise 3, the TC worked with the leader of the assault team to get a landline phone to the suspect:

```
TC: We’re losing phone contact.  Phones are going dead.
Can you spare me two to come back and deliver throw phone?

TL: Affirmative.
```

```
TC: Alright, send two to come back and pick up the phone.
I'll find out where we're gonna bring it to.  Do you have
a preference where you wanna go in?
TL: We're gonna deploy it from the west side.
TC: What's the ETA to get back?
TL: Two minutes.  Be right there.
```

A few minutes later, after the officers had returned to their team with the phone, the TC asked the leader:

```
TC: Are you ready to deliver the phone?
Team Leader: We're ready to deliver the phone.
```

There was then a delay of several minutes before the TC gave the order to deliver the phone as he waited for agreement from the Hostage Negotiator Team.

In another example, the TC needed to confirm the location of the ERT and consequently the capabilities of the team:

```
Confirm you're still in front lobby.  (We) Have someone to
bring out the front door.
```

Through many such dialogues with the distributed teams, the TC was able to improve his mental model of the incident and, in particular, understand the effect that he might be able to have on the situation by commanding action from his teams. The moment-to-moment capabilities of the teams were constrained by the myriad of variables in the environment that were not observable by the TC. Thin walls, open doors, and excited suspects are all examples of small details that effected the options available to a team. By asking the teams directly for reports of their abilities, the TC could be assured that his commands would be valid and relevant for the situation of each team.

Verification

Throughout each exercise, the TC moved around the ICP area, monitoring the radio chatter, talking to the Hostage Negotiator Team, getting reports from the various ICP dispatchers as they became available, and looking at the maps and logs. Periodically, the TC was asked explicitly about the state of the incident and he always had at least a rough but correct idea of his units' locations.

4.3.2 Coordinating action

Members of the SWAT teams cultivated common ground throughout the exercises. They were observed to use this common ground to initialize coordinated action. In addition, the coordination of the team's actions with other external entities (e.g. negotiators or paramedics) was an extremely important function of the TC.

The next sections highlight two ways in which the Tactical Commanders relied on common ground to coordinate action.

Referencing standard procedure

Because members of the SWAT team had a base of common ground established through their training, the TC usually assumed that a set of basic SWAT capabilities were available on request, such as 'clear-outs', dynamic or stealth movements, or holding hostages or suspects. For example, in each exercise the snipers asked for rules of engagement as suspects appeared in their sights. The reply from the TC was a simple statement of "standard rules of engagement" which conveyed an extensive set of behaviors and commands that were immediately invoked.

Using the addressee's frame of reference

The other method the TC used for coordinating action was to give commands in the addressee's frame of reference. For the most part, the dialogues were conducted relative to unit position. The TC smoothly alternated between the points of view of the team members under his command. In a typical example, in Exercise 4, the TC first told some team members who had a hostage in custody, "Have her (hostage) go out those doors that are right by you." He then immediately told the rest of the team, in a different location, "You are clear to go through the first part (of the floor clear-out procedure) to top of the (next) stairs."

The TC also conveyed the location of the teams relative to one another as well as to other objects. For example, the TC coordinated between the Emergency Response Team (EMT) leader and the assault team by conveying each of their locations. When the EMT leader asked if they could move inside the front door (where presumably they could have a more active role in watching for activity in the building), the TC replied:

> I don't have a problem with that. Just remember that the assault team will be moving from your left. Stand by before you move. I'll have the assault team stand by in the lobby by the double doors.

Use of objects

As with the cultivation of common ground, the TC used objects, referred to in the units' local frames of reference, to coordinate action as shown in this example from Exercise 2 (object references in **boldface**):

> TC: White, do you have a visual on **the suspect**?
>
> OW: No, there is **a large stack of boxes** between me and where I hear what I believe is **the suspect**.
>
> TC: Black, do you have a visual on **the suspect**?
>
> OB: Yes, I can see what appears to be **a foot with a blue tennis shoe** but not more than that.
>
> TC: Black, do you see **a stack of boxes** to your left in the direction of White?
>
> OB: Affirmative.
>
> TC: Black, do you see **a location** for White to egress to that remains in cover?
>
> OB: Yes, there is **a desk** with a computer immediately to his left when he comes around **the stack** that he should be able to get to.
>
> TC: Did you get that, White?
>
> OW: Affirmative, moving to **the desk**.

Wider context within local referencing

The following example illustrates how the TC could provide wider context for the units when it was needed, but still maintain the use of local referencing.

> TC: Green, do you see a door in the east wall below the suspect?
>
> OG: Affirmative.
>
> TC: We need to get someone in that room to make sure the suspect does not move further into the building.
>
> OG: White is in position to move into that door. I can cover the door while he moves in. Black can also move to that door if you want him to - he'll just take a second to get into position.
>
> TC: I want White to stay where he is to be able to assist with the assault up the stairs in a couple of minutes.
>
> TC: Black, move into position to move through the door under the suspect.
>
> OB: Moving into position.

Use of reverse local referencing

The use of relative positions was even used by the unit leaders when talking to the TC, who should not necessarily be expected to be familiar with their frame of reference. In this example, the Assault Team Leader (ATL) requests information from the TC to determine whether some noises being heard were friendly or not:

```
ATL: We're at the top of the stairs.  We hear something
over on the right side, are we clear to move through
there?

TC: (Who knew that the noises were being made by
administrators of the exercise) Yes, you're cleared to
move through.
```

Confusion due to incorrect central information

These translations by the TC between the various perspectives among the teams and team members were frequent, and relying on the TC as the locus of coordination generally was extraordinarily reliable. However, the reliance on the TC that developed also proved to be a weakness. For example, at one point in Exercise 1, the assault team asked for a floor plan for the area in which the suspect and hostages were located. The TC had unintentionally been given the drawings for a different floor than the one desired. An assault plan was devised based on this incorrect floor plan to enter on opposite sides of the building and proceed in two teams around two hallways that passed along the walls of the building. Upon commencing the assault, however, the assault teams reported that there were no outside hallways, but they could see one another down one central hallway going through the center of the building. This error caused an unexpected, potentially lethal crossfire situation that every effort is usually taken to avoid.

Coordinating with external entities

The SWAT team does not operate in a vacuum. In fact, quite a few completely external factors influence the conduct of the team, such as the weather, coordination with firemen and paramedics, and public pressure. However, the individual team members, as a rule, cannot be expected to be aware of these factors while concentrating on their own job. The task of integrating this external information and passing it on to the team – either explicitly through communication or implicitly by changing team goals or strategies – falls to the Tactical Commander.

Throughout the exercise, the TC moved throughout the ICP area, looked at maps and logs, talked to the HNT either directly or (more often) through their tactical dispatcher representative, solicited reports from the various other dispatchers, and listened to the radio chatter.

As an example of this coordination, in Exercise 3 the suspects, or at least one suspect, was demanding that the throw phone be brought to the top of some nearby stairs. The HNT then sent a message through a dispatcher that they needed the phone on top of stairs, and needed from the TC an estimated time of arrival of a team to take it there. The TC responded that the first floor had to be cleared first and that it would take a few minutes.

The suspects had earlier demanded water for themselves and the hostages. They had agreed to give up some hostages for the water. They also wanted the phone, but had not received it. The TC asked around in the ICP and was told the water will take a few more minutes to get, but two assault team members were back to take the throw phone. He tells the collected ICP inhabitants, including the Hostage Negotiator Team dispatcher: "OK. I'll tell them the phone first for the hostages, then the water." In this way, the TC changed team goals (from a water-for-hostage to phone-for-hostage exchange) to accommodate changes in external information.

4.3.3 Discussion

In analysis of the observations of SWAT field exercises, the cultivation of common ground was found to be an ongoing and critical activity for these extreme teams. One of the most important and challenging activities for the commander of these teams was ensuring that team members re-calibrated when the situation or environment changed and team members no longer shared common ground. This was particularly dangerous when common ground had slipped away unnoticed, leaving team members assuming that they had a shared understanding of the situation and environment.

The analysis also revealed that the TC coordinated action using common ground as the global frame of reference. While the TC developed awareness of the entire situation by integrating information for all sources, he could decompose that picture to issue commands that used the addressee's frame of reference, and share awareness information about objects and people from the addressees' perspective.

Application to distributed teams

Analysis of the conduct of commanders of SWAT teams also provides general knowledge about leading distributed work teams. The analysis suggests that these roles are present to some extent in all distributed teams, but play a vital role in extreme distributed teams because of their unique characteristics. With distributed teams in less of an "extreme" setting, the members may be able to carry out more direct communication, reducing the demands on the leader.

This research found that strong leadership in a face-to-face meeting at the outset of the activity was crucial for establishing common ground. In the Situation Reports, commanders focused on making sure that team members had a shared understanding of the situation and understood others' roles. This is consistent with Armstrong and Cole's [85] finding that distributed teams were more effective if they met face-to-face to discuss how they planned to approach the task and the roles team members would play. Along with Weisband [91], this research found that leaders are particularly important in establishing common ground during these early phases of a project. Thus, leaders of distributed teams may be more effective if they assemble new teams early on using the richest possible medium and build a foundation of common ground that can be leveraged later in the project.

Another analysis result is that a vital role of leaders on extreme teams was to share awareness information about the activities and positions of other team members. It was inefficient for team members to share this information because they did not know who would need it and they did not know how to convey the information from the perspective of the recipient. Leaders who are in contact with all team members and have a global picture of the team's status are in the unique position to provide this translation function and ensure that team members understand their colleagues' situations. Weisband [91] observed that distributed teams were more effective when their leaders shared awareness information about others on the team. The results given here support and extend this work, suggesting that leaders of distributed teams will be more effective if they convey information from the perspective of the recipient. This analysis suggests that sharing information based on the recipient's perspective aided significantly in teams being able to make sense of the situation and their role in it.

Confusion due to incorrect model integration

The difficulty of maintaining common models of the environment was made evident through one incident during the third exercise when the command team and the assault team were confused about which staircase they were supposed to climb. The building was symmetrical, with stairs on each end. The plan was to use a hostage exchange to attract all the suspects to one staircase and send the SWAT assault team up the other staircase. However, the diversion almost backfired when the assault team unknowingly went to the wrong staircase. They were surprised to see a hostage in the stairwell just as they began to move. They informed the TC that they had a hostage in custody in the staircase, and in a few seconds the TC realized the source of the confusion and had the team immediately move away from the staircase to avoid the potential conflict.

Maintaining a mental model of the world

The SWAT team dispatchers, who are responsible for gathering and disseminating information, made maps of building interiors and exteriors and labeled the unit locations – for the first thirty minutes. These maps were not updated throughout the remainder of the incident. Contrary to TC statements in interviews, the Tactical Commanders used the maps infrequently and preferred to develop a comprehensive mental picture of the incident. This might be due to the difficulty in representing all the knowledge effectively on paper or reluctance to rely on a dispatcher's interpretation of events. In the field, the assault teams had paper copies of floor plans and street maps, but otherwise they too maintained their model mentally.

4.4 Relating ethnography to design

Ethnography is relevant to design for several reasons. First, since designers' creations often are used in settings that the designers know little about, some understanding of those user settings is needed so that the technology is better suited for its intended use. Without some information about the settings, designers must rely on their own experience and imagination and run the risk of designing technologies better suited to their own needs than their users. Second, those creations can have a real intentional or unintentional effect on shaping work practices, which can have a significant effect on the utilization of the technology and efficiency of its use. Third, users are often unable to give meaningful responses when asked about how they might use technology. Ethnography allows the designer to envision technology use and have meaningful discussions with its future users.

The purpose of this analysis step was to inform the design of a human-robot interaction for the command of teams of field robots. There were three key results significant enough to be passed on to the next design step, innovation.

4.4.1 Operator roles

The field robot operator should carry out the roles identified for the SWAT commander – cultivation of common ground and coordination of action. These roles for the robot operator differ slightly from the existing ideas about what part the operator needs to play in existing robot systems. In current multiple robot systems, the robots typically have a decentralized approach to receiving and sharing data and the operator accesses and adds information from the periphery of the system. As cultivator of common ground, the operator is the central location of information that can then be disseminated as appropriate. As coordinator of action, the operator needs to have access to the current robot capabilities to integrate the system state and external information into valid and useful plans. The computer interface needs to adapt to the roles as well, providing mechanisms to cultivate common ground between robots and coordinate their action.

4.4.2 Use of objects

The roles of the operator, and indeed the complete communication and conduct of the team, relied heavily on the use of physical objects in the environment as points of reference. This use of objects should be replicated in some manner in the human-robot interaction. Most existing robot architectures to not use objects as a mechanism for control.

Given the ubiquity of the use of objects in the SWAT setting for collaboration and coordination, the following principle was formulated with which to base any new interaction:

Base the interaction on the objects the robot senses and the tasks that can be performed on those objects.

This principle provides the operator with the information she cares most about: the objects in the world whose state she is managing and the means of modifying those states.

4.4.3 Use of local frames of reference

Rather than require all SWAT members to work within one global frame of reference which would have required an extraordinary amount of information sharing, the Tactical Commander integrated all the individuals' perspectives into a consistent mental model but communicated using each individual's frame of reference. Multiple robot systems have generally sought to form one global model that is then used by all robots and operators. The new human-robot interaction created through this research should seek to preserve the local frames of reference while providing tools to assist the operator with her global mental model.

Chapter Five

Object-Based Task-Level Control of Multiple Robots

As discussed in Chapter One, after analyzing the observation data, the next step in the design process (Figure 5.1) is to create innovations that can be carried out in the final step, implementation. The innovation step includes the identification of useful concepts that already exist, the extension of these concepts in new directions, and the creation of new ideas. The innovations for a field robot system may be for both its robot and computer interface components.

The innovations described in this chapter are the selection of an appropriate robot control paradigm given the analysis of the SWAT observations, and the extensions necessary to use this paradigm for the operation of multiple robots by a single operator. The next chapter, Chapter Six, will discuss the innovations surrounding the computer interface component. The first of the five sections in this chapter reviews the three requisites for the control paradigm were derived in Chapter Four: that it support the operator roles identified in the SWAT analysis, utilize physical objects as common points of reference, and accommodate the use of local rather than global models of the environment. The next two sections describe the various ways that field robots could be operated, and then focus on one particular method – Object-Based Task-Level Control (OBTLC). The fourth section, through a discussion of past achievements by OBTLC robots, points out that OBTLC has

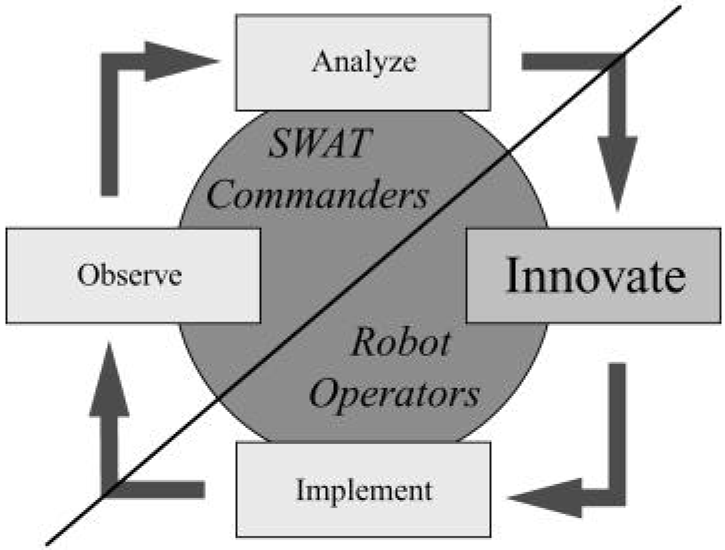

Figure 5.1: **Step Three – Innovate**

the potential to fulfill the requirements developed through the SWAT analysis, but it has not been extended to systems with more than two robots or more than a single object or task. Some issues arise when extending OBTLC to multiple complex robots, such as the lack of an authoritative source of global information and the challenge of determining the correspondence between objects detected by distributed robots. The final section describes methods of addressing these and other problems through careful information integration and the introduction of software agents.

5.1 Requirements for development of the robot component

Given the analysis results from Chapter Four that have determined the requirements of multiple robot operation, the next step is to establish what possibilities and constraints exist for robot design. The most important criteria, based on the analysis, are that the robots support the key operator roles, use objects as points of reference, and maintain a local model of the environment.

5.1.1 Robot support of operator roles

The operator needs to take on two important roles to be an effective commander of an extreme distributed team such as a multiple field robot system: cultivation of common ground and coordination of action.

Cultivating common ground for multiple robots

To gain a sense of how to cultivate common ground in robot applications, it is helpful to look at the different steps of this cultivation as found in the ethnography analysis from Chapter Four.

Establish common ground through training

For robot systems, the training aspect comes at the time of design and the initial setup phase of the system when basic capabilities are implemented. This is true for meta-issues such as centralized planning techniques and emergent behavior strategies as well as the robots themselves. Additional utility might come from robots that could be easily 're-trained' by the operators in the field between missions, rather than requiring reprogramming by the original developer.

Calibrate common ground at the start of a mission

Calibration began for the SWAT team with the Situation Report and the distribution of maps. This step for the robots would consist of uploading the important relevant databases to the extent to which they are known, and building as much infrastructure to assist the operator (e.g., a three-dimensional representation of the fixed items in the workspace) as can be provided.

Re-calibrate common ground throughout the mission

The re-calibration step takes place continuously as the mission is underway. Re-calibration has the same components for both the SWAT teams and robots: merging data models and building the task model.

An important responsibility of the commander is to merge the local models derived by each distributed member into one logically consistent global model. This can take place completely mentally, as it does with the Tactical Commander, or it can be assisted by technological means. However, integrating these models *automatically* using technology may not only prove difficult, but could add cognitive workload to the operator if the electronic version is different from her own internal mental one.

Just as the local data models must be obtained and merged, a model that contains the tasks possible by each distributed agent needs to be built and maintained so that the operator (and possible surrogate operators such as automated mission planners) knows what capabilities are at her disposal. Like visually surveying the tools in a toolbox, the operator can both determine the possibilities for direct action and use the capabilities as building blocks for more complex creative problem solving.

Coordinating the actions of multiple robots

The other key role for the operator is to coordinate the action of the remote agents. In robotics, this task is often undertaken by automated mechanisms. The complexity of real world tasks has thus far constrained automated planners from widespread implementation and use, even when those tasks are simple by human standards. Consequently, humans are currently the best planners for real world missions. The robot system design needs to incorporate features that employ the strengths of the human as coordinator, such as coordination within the context of external demands.

Tactical Commanders were able to achieve complex tasks very simply by referring to the use of standard procedure. These procedures encapsulate a great deal of information, decision-making standards, and constraints. For a robot, this could be translated into high-level tasks that are built from lower-level capabilities. By providing the operator with the choice of high-level tasks predefined by the developer or operator, complex behavior can be initiated through simple direction.

One of the largest drawbacks of automated planners is the difficulty with which they can accept new, unknown external constraints on the system within the context of the plan. Planners must include cost functions or other mechanisms for relationships between resources and constraints to influence the conduct of the system. In real world applications, situations often arise that were simply unimaginable at the time the system was created. For such situations, the human is an extremely flexible and creative resource for finding solutions.

5.1.2 Use objects sensed by the robots

All elements of the SWAT team used physical objects as significant points of reference throughout their mission. The commander used them both to cultivate common ground and to coordinate action. Objects are powerful tools because they are able to encapsulate so much other information beyond their name and shape. In particular, objects are able to afford a range of action options to the person interacting with it. For instance, a simple wooden chair will have a certain size, weight and color that may be readily described. Yet, a person also knows that it can be sat upon, stood upon, used to break a window, used as a dance prop, or even splintered into firewood, among many things. The chair may be used in conjunction with a table or an orchestra pit and is usually in the company of other chairs. This extra information allows communication, particularly regarding action and the state of a physical environment, to be much more efficient.

Objects also have a physical presence that is typically mutually exclusive of the existence of other objects. Consequently, objects are an excellent way to consolidate perspectives. If one person sees the Golden Gate Bridge on his left and Alcatraz on his right, and his friend sees both directly in front of her, then since there is only one Golden Gate Bridge and one Alcatraz they can know conclusively that they are in separate locations and can even create a rough idea of their relative positions.

Perception and recognition of objects is difficult for robots to do robustly. The determination of an object's proper and current affordances must be explicitly taught to the robot. Keeping track of unique objects over time is also an important capability that is easy for humans but still difficult to provide for robots. These topics are being actively addressed by ongoing research [96, 97].

5.1.3 Use robots' local frames of reference

The deployed SWAT team members had no ability to see what the other members were seeing, nor even where the other members were unless they were directly in their line of sight. Robots, on the other hand, may be generally assumed to be able to communicate readily with one another. Yet, in areas where their sensing systems overlap the robots will likely disagree about the state and possibly even the nature of the environment. Efforts can be undertaken to consolidate these views, but ultimately a robot senses what it senses and this perception is what it must use for its own feedback control loops. Consequently, the operator must send commands that are pertinent, consistent, and sensible in the individually sensed local frame of reference of the field robot. This ensures that the robot is clear about what must be done and that the closed-loop feedback on the robot does not have any unnecessary error introduced. This requires the operator to maintain a separate local model for each robot that, given the potential amount of data produced by multiple robots could be confusing and mentally taxing. Software can be introduced to ease this burden, but it must take care that the independence of the local frames of reference be maintained.

5.2 Methods of operating robots

Over the course of the developments and deployments of the myriad field robot systems, many different methods of operation have been used. The design of the control system architecture is a major issue during the development of field robots due to high-dimensional sensory data, computationally intensive processing, and real-time execution constraints, and the usefulness of the various alternatives varies greatly depending on the application [98]. As shown in Figure 5.2, one useful way to consider the variety of control architectures is by seeing them as a spectrum with the level of human or computer control as the independent variable.

Teleoperation

The oldest method of field robot operation is teleoperation. Teleoperation is simple to implement and simple to understand. In teleoperation, the operator controls the movements of the robot directly. The servomotors move in direct response to the operator commands, and the operator typically receives the basic sensor measurements. There are many examples of robot teleoperation, from the Luna moon rovers in the 1960's to flying target drones of the 1970's to today's underwater remotely piloted vehicles (ROVs). While teleoperation is usually simple to implement, there are many well-documented problems with teleoperation, particularly in scenarios with significant time delay or when the complexity of the system is large.

Supervisory Control

When sufficient autonomy can be created to allow the robot to perform some tasks autonomously, the operator is said to use supervisory control. Supervisory control has a

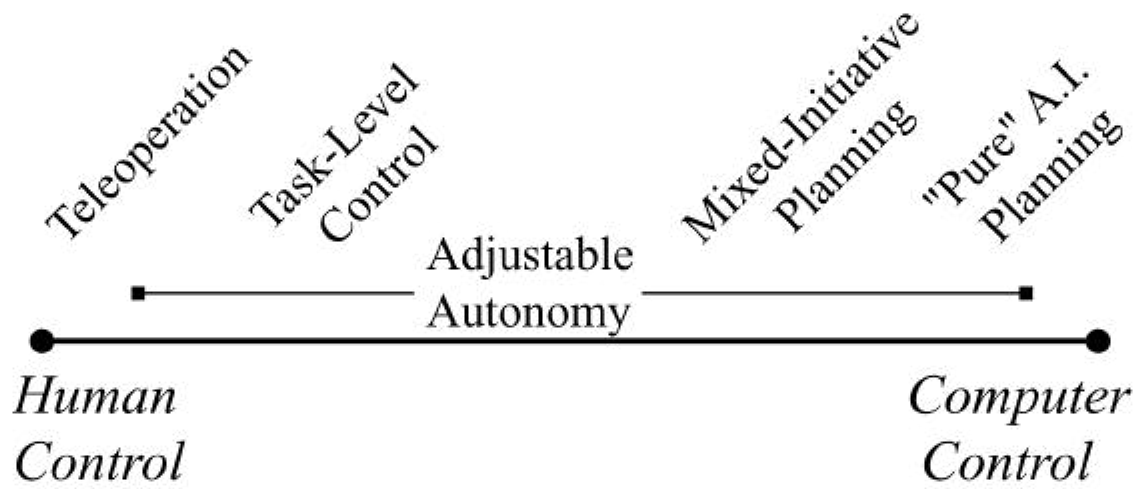

Figure 5.2: **Spectrum of Operation Methods**

focus on abstracting the control of the robot to a higher level; the operator selects the task to be completed and monitors the robot's execution of the task. If the execution is performed correctly, the operator selects a new one and the process continues. If there are problems with the execution, the operator will either take over and teleoperate the robot or end the task altogether. While the goal of such automation is often to reduce operator workload, research has shown that operators typically perceive systems with such automation to be more demanding [67].

Autonomy

Although not seemingly a method of operation, automation is one extreme of the spectrum of human involvement in robot operation and as such it should certainly be considered an operation method. Automation is often programmed by the developer, moving the role of 'operator' from the human on the scene to the human at the drawing board. This change in the location of decision-making has proven difficult to sustain for complex real systems, since unanticipated problems often arise that the autonomy was not built to handle. For simpler systems, however, complete autonomy can be extremely efficient and effective.

Others

There are many other control architectures that exist within the spectrum between teleoperation and autonomy. Mixed initiative planning systems trade off responsibility for planning between an operator and an automated planner [99]. Adjustable autonomy seeks to enable a range of autonomy during operation [100], Swarming behaviors are a recent innovation for multiple robot autonomy [101], and operator-devised scripts have been used to give human input into autonomous behavior [58]. Some of the most recent research has sought to make humans and robots collaborators on the mission [65].

5.3 OBTLC

Of the robot operation methods discussed in the previous section, none provides support for the findings of the analysis of Chapter Four. None of them uses objects explicitly, and the roles played by the operator either do not encompass the cultivator and coordinator roles or require so much else from the operator that these roles become minor. A robot architecture was sought which could make the robots conform as closely as possible to the requirements suggested by the Chapter Four analysis.

The Stanford Aerospace Robotics Laboratory has developed and implemented Object-Based Task-Level Control (OBTLC), a robotic control architecture to support the creation of human-robot teams. Originally developed for use with remote robotic systems for space applications [23], OBTLC has been employed in space robot simulators [102], flexible manipulators [64], factory workcells [26], and underwater vehicles [40]. The philosophy of OBTLC has three main tenets [41]:

- modern technology cannot produce a substitute for human intelligence,
- automatic feedback control is unaffected by any limits in human bandwidth, endurance, or capacity to manage multiple tasks simultaneously, and
- when humans and computers are insightfully integrated, the resulting robotic system is more effective than if either were used alone.

The phrase "object-based task-level control" describes the interaction between the human and the robot. "Object-based" designates the focus of the human supervisor's attention when using the robot. Rather than concentrating on *how* to control the robot by driving the low-level actuation, the operator perceives the situation at hand and formulates *what* needs to happen to accomplish the mission goals. The term “task-level” signifies the level of human interaction. The information passed from the human to the machine is a command that may range from “move yourself to the left” to “follow the red object.” The robot then obeys this instruction autonomously. As a result of this division of labor, OBTLC has been shown to be an attractive alternative to the use of teleoperation or strict supervisory control [38, 64].

The key attribute of an OBTLC implementation is the creation of a human/robot team. Ideally, humans and machines would coexist in a unified system with the duties of each member optimized with respect to their inherent abilities. Human intelligence is more capable of inferring information from scarce telemetry, setting goals and planning missions, and responding appropriately to unexpected situations. At the same time, computers are capable of the many chores necessary to control robots at a low level, such as controlling multi-input/multi-output systems or processing large amounts of sensed data. Although this ideal relationship is very difficult to realize [21], OBTLC does provide a framework for iteration of research towards this ideal. As such teams are deployed, some very interesting research possibilities will be created, since previous studies of automation use have been with automation as a team tool [103], not as a team member.

Previous interfaces for OBTLC systems were created to display the newly developed capabilities of a particular robotic system. The interface software only needed to afford one operation to the user, and it was usually written for one particular robot/task/object scenario. For such a singular purpose, an OBTLC interface is not difficult to create. In fact, these past OBTLC human-robot interactions made robotic control extremely straightforward and highlighted the power of OBTLC.

Previous work: General capabilities

OBTLC was used to develop a wide variety of capabilities that automate what a skilled teleoperator could do with robots meant for near-earth orbit space applications, and it was then extended to a range of tasks that would not have been possible whatsoever through manual operation. Capture of a moving, spinning object, slewing it to a new position, and inserting the object into a fixture was done using the OBTLC framework by arms on a fixed base (the arms had both rigid [23] and flexible drive train [25]) and a free-floating base [102].

The robots that consisted of arms on a free-floating base were able to make the capture if target was evading [104] or even if the environment was full of moving obstacles [105].

OBTLC was also used for other robot applications, such as the operation of intervention-capable autonomous underwater vehicles for station keeping and object retrieval [24] and object search and tracking by an autonomous helicopter [38]. The participation of the human operator in augmenting the object detection and modeling of the robot was examined by E. Miles [106], the most significant previous research to focus on the role of the human in the OBTLC architecture.

Previous work: Use with multiple robots

Several projects implemented OBTLC on robotic systems with two robots, and two projects used three robots. Two projects used arms on a fixed base: two arms with flexible drive trains cooperated to do a delicate insertion task [107] and two arms captured moving objects and transported them cooperatively [26]. Two other projects used robots that floated freely about the workspace and used a pair of two-link arms for manipulation: two robots captured a moving object and then cooperatively docked it to a fixed base [63] and the same two robots assembled two objects into one [108]. The formation flying work of Robertson [109] and Corazzini [110] utilized three robots, although the robots did not apply much of the OBTLC control mechanism since there was no manipulation of objects.

Previous work: Use with multiple objects

Three OBTLC projects have used multiple objects. The Russakow research assembled two like objects into one new object [108]. The work by Pardo was also an assembly, and although his objects were of many shapes and sizes they were all of effectively the same building block type. In the human perception research by Miles, the identification and manipulation of unknown objects was of four different objects [106].

Previous work: Use with multiple tasks

In a sense, the OBTLC research that consisted of grab, move, or dock compound tasks utilized a multiple-task process. However, these tasks were executed according to a predefined finite state machine and only one capability was offered to the operator: the determination of the final position and orientation of the only object the robot considered to be available for manipulation.

In all, a number of tasks have been developed but no previous work had provided more than one at any time to the human operator. The list of simple tasks developed thus far includes grabbing, inserting, following, nudging, and throwing objects. Possibilities of complex or compound tasks would be relative station keeping, pointing a camera, inspecting an area, connecting two objects, or assembling a structure

5.4 Challenges with extending OBTLC to complex robot systems

There are a few reasons why OBTLC has not been extended to complex robot systems – those with simultaneous utilization of multiple robots, objects, and tasks. Two of the reasons may supercede the others but are not important from a fundamental research perspective: running experiments with multiple robots is more difficult so students avoided applications that required demonstrations with more than one or two robots, and the experimental and computational resources were not readily available to create multiple robot systems. Other reasons exist, though, that are fundamentally challenging for the use of OBTLC with multiple complex robots. Because of these problems, OBTLC had proven to be a powerful but non-scalable solution to field robot operation.

No source of global information and limited local sensing

OBTLC systems require that objects in the environment be sensed and identified before they can be utilized. This makes sense – a robot cannot intentionally use something it does not perceive in some way. Although OBTLC has been built with field robots in mind, previous demonstrators operated under the assumption of complete state knowledge of all objects in the environment. As mentioned previously in this dissertation, this is unrealistic in a typical field robot deployment. Some robots might be able to operate in a fully structured environment where all objects could be labeled and readily monitored, such as in on-orbit space construction. Otherwise, sensing the state of all objects in the environment is very difficult, and usually even more difficult to identify the objects. Only the work by Miles has addressed this issue, augmenting the robot's perception with the human operator [106]. Because the operator relies on knowledge of the objects' states to generate all commands, this is an important attribute when designing the computer interface for an OBTLC-based interaction. Yet, perfect knowledge is fundamentally impossible in an actual operational system, and even approximations are unrealistic when the system contains the varied perspectives of multiple robots.

Object correspondence

Related to the aforementioned problem of requiring the robots to sense the objects themselves without a global sensor, there is the issue of object correspondence when two or more robots work in proximity to one another. For robots to cooperate on object-based tasks, they must agree on the state of the objects in the environment. If there is disagreement, then fundamental problems with the task execution are likely to occur. Some method or methods need to be used to determine object correspondence between robots.

Multiple tasks: Beyond drag-and-drop

Providing commands to the robots was straightforward with previous OBTLC implementations because only one task capability was being developed and consequently only one task would be required by the operator (who was usually the developer). Since all OBTLC interfaces to this point have used a mouse, a graphic representation of the robot workspace, and a robot performing a motion task, a simple drag-and-drop mechanism was sufficient. Yet, if more than one task is possible, such a mechanism is ambiguous. Some method must be used to differentiate between the various possible tasks available for the operator and then to communicate the appropriate choice back to the robot. In addition, the actions possible by a robot may change from minute to minute and these changes must be reflected in the interface. Finally, past interface applications would have to be updated and recompiled after each change in robot functionality, even if the functionality of the interface itself stayed the same.

5.5 Methods to address the challenges of OBTLC for complex systems

Based on the results of the analysis in Chapter Four, the Object-Based Task-Level Control architecture was chosen for implementation. This choice yielded a set of challenges given in the previous section. This section describes the innovations that were made to address these challenges that ultimately allowed the implementation to be discussed in Chapter Seven.

System assumptions

Some assumptions about the capabilities of the system and its operating environment were necessary to provide a foundation for the innovations. These assumptions are consistent with the current or near-term capabilities of field robots given the expected success of ongoing related research efforts. Implementation of a system that incorporates these assumptions is described in Chapter Seven.

Robot requirements

The robots must be able to

- sense the state of objects including their position, orientation, and type
- manage knowledge about these objects over time
- communicate their own state and the state of the objects they sense
- determine which object-based tasks are possible at any time given a query about an object
- accept object-based task-level commands
- conduct high-level tasks autonomously as exemplified by previous OBTLC research projects
- robustly detect and handle faults, either internally or with the help of specialized human operators who act as mechanics
- maintain their physical safety thorugh successful obstacle avoidance and rejection of unintentionally harmful commands

Environment qualities

The environment must be structured enough to be consistent with the perceptive abilities of the robots. For example, exploration of a Mars rock field without the ability to perceive and distinguish rocks would not be an appropriate application. In addition, the rate of change of the environment relative to the time delay for communication should be slow enough so that the level of interaction by the operator is effective.

Addressing the lack of global sensing: Maintain local models

As observed for the distributed SWAT officers, a useful practice for a distributed team is for the local models developed by each member to be maintained by that member. For field robots, this means that although the operator may derive a comprehensive world model built from the sensor information from all the robots (either mentally or with assistance from some technology), the actual task instruction must be given in the perspective of the robot. The robot will not likely know the state of the entire system [111], and will need to use its own internal local sensing for closed loop control to carry out the task. The ramifications of this approach for the computer interface will be discussed in the next chapter.

Object correspondence

To determine correspondence of objects between local environmental models, some standard mechanism is necessary. This correspondence information is useful for the operator, to other robots, and to planners or other system-wide entities. Since the data from robots can be extensive, automating this process to the extent possible would be very beneficial. A few methods to do this exist, but none is generic and the most useful are application-dependent. The human brain performs correspondence tasks continuously, at a high rate, and with an allowance for quite a bit of fuzziness or uncertainty in the information. For that reason, the human operator should have some input on the correspondence between robot sensors. How to represent the robot data so that the human can best make such a distinction effectively was an uncertain issue that is addressed in the next chapter.

Moving beyond drag-and-drop: Multiple-choice commands

As more than one command may be possible for a given robot/object pair, there must be a greater task-related infrastructure in place than in past OBTLC implementations. Rather than have the task automatically called by the interface when a drag-and-drop occurs, the system should find out what choices are available to the user. As mentioned in Section 1.1.3, changes in the robot capabilities may even occur in the course of a mission. A mechanism needs to be in place to ask the robot what is possible, and the robot needs to be able to answer that question. A dialogue about capabilities, just as between the SWAT commander and his field units, must take place between the operator and each field robot. To implement this model, each robot will need to maintain a list of objects that it senses, and lists of the actions that it can carry out on each object. Managing this information will create a burden that must be shared by the operator, the computer interface, and the robots.

Chapter Six

Direct Manipulation Interface for Multiple Robots

As Chapter Five described the innovations within the robot component of the field robot system, this chapter will discuss the innovation step (Figure 6.1) as it applies to the human-computer interface. As outlined in Chapter One, the next step in the design process will be to implement these innovations from this chapter and Chapter Five.

The first of the four sections of this chapter discusses the factors influencing the design of the human-computer interaction: the analysis of the SWAT observations, the choice of OBTLC as the robot architecture, and basic considerations for effective interface design. The next section describes different interface modalities used to control automated systems. Of the many different methods that could be used, one has been widely used and refined for systems of multiple entities – direct manipulation (DM). Within direct manipulation, in fact, a particular method of interaction has been developed for dynamic systems of virtual robots – the interface for the genre of video games known as real-time strategy (RTS). There are issues, described in the third section, with extending DM/RTS interfaces for use with actual robots because these interfaces are built with assumptions not valid for field robots. In the final section of the chapter, innovations are presented that broaden the scope of DM/RTS interface applications to field robotics by utilizing an interactive task selection method and software agents.

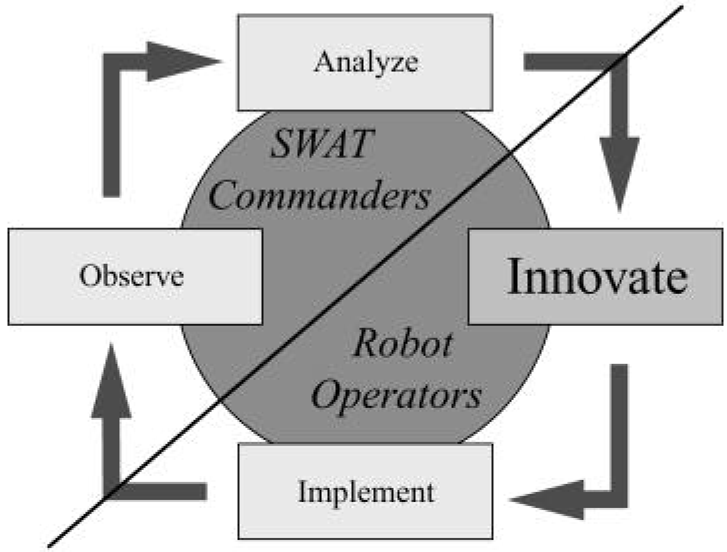

Figure 6.1: **Step Three – Innovate**

6.1 Requirements for development of the human-computer interface

Developing the human-computer interface is the most constrained part of the design process, as it must not only accommodate the best design methods for interface development but also complement the existing robot system and the proposed operator roles and tools.

6.1.1 Follow basic usability guidelines

When designing a system for direct human use such as a computer interface, there are some key considerations that will make the final result more usable. In particular, the designer should try to create a system that suggests the proper action at the proper time by first providing for what is natural, then enabling what is visible, and if that is not possible, providing what is standard [21].

In the case of the control of multiple autonomous robots, there is no well-known and accepted method of operation. Thus, this research has attempted to address this challenge by:

- Providing an interaction that is natural for a similar situation,
- Carefully protecting what is made visible, and

- Enabling the closest possible approximation of a potential standard.

Since there are no existing operational multiple-robot systems to study, the search for a 'natural' interaction required that a close proxy to such a system be found. The essence of supervisory control has often been compared to a supervisor's interaction with subordinate human staff members [1]. An analogue to the experience of controlling multiple field robots might then be, as argued in Chapter Three, the management or command of a distributed team of human individuals. Chapters Three and Four described the observation and analysis of police SWAT teams, which serves as a setting to learn how a natural interaction for field robot operation might be created.

The attention of the human operator is a valuable resource, and interaction designers should be careful to use that attention wisely. The best way for interactions to do so is to make sure that they afford the correct – and only the correct – capabilities to the operator at all times so that she is not distracted by irrelevant information. As a result, the affordances of the interaction could be counted on as a valid representation of the robot state. Simply stated, if a task is visible it should be possible, and if a task is possible it should be visible.

There is no standard for the operation of multiple autonomous robots by a single person. This research did identify, as a likely potential standard, the genre of computer entertainment software known as real-time strategy (RTS) gaming. Millions of potential robot operators are already familiar with the RTS interface as a means to operate virtual field robots, making it as close to a standard for field robot operation as exists today. In these systems, the entities being controlled behave like a simulated robot and in many cases are meant to represent an actual robot. However, as one would expect, there are significant differences between entertainment software and robot operation.

Some military planners have also mentioned the likelihood of the adoption of video game-like interfaces for actual robot operation: "(Current) UCAV (Uninhabited Combat Air Vehicle) operator requirements build on (manned airplane) combat pilot requirements. In the near future, this will change. One of the major changes has to do with crew selection regarding basic operator skills. In the past, combat pilot selection was specific, with high physical and mental requirements. A UCAV operator, however, will need to have good video game proficiency [22]."

6.1.2 Complement existing operator and robot components

In addition to these general interface design considerations, the human-computer interface should be developed within the constraints identified in the previous chapters.

Operator roles

The operator must be able to participate in two important roles: cultivating common ground and coordinating action. The interface should enable these roles through natural means whenever possible. To cultivate common ground, the operator would seem to need to be able to observe the various perspectives and then make connections as necessary. Direct access to the current task model and the ability to call for tasks in real-time appear to be requisites for coordinating action.

Use of objects

The interface must allow the use of objects by both the operator and the robot. Ideally, the interface would not mandate the use of certain types of objects, but instead be flexible to handle whatever types the robot senses and reports. Ready integration with the Object-Based Task Level Control robot architecture is also required.

Integrate external information

The interface has to assist with the integration of external information, such as constraints, time pressures, and creative opportunities, into the conduct of the robots. This will most likely be accomplished by effectively involving the user in real-time operation.

No global source of information

There is no global source of information about the system, and the interface must be able to handle this condition. In comparison, most interfaces depend on a single source of comprehensive and accurate information to provide the data to be displayed.

Maintain local models

As there is no global source of information, data is provided by many local sources of information. As a result, not only must these local models be consolidated into a consistent global model for some needs, they must also be kept separate so that information (e.g. commands) that flow back to the source of the local model are understood.

6.1.3 Suitable interfaces

A variety of potential interface modalities exist that could satisfy many of these stated constraints, ranging from voice-only to complex computer graphics.

Voice-only

As this human-robot interaction was based fundamentally on the conduct of SWAT teams who only use voice to communicate, an interaction could be built for the robots that only uses voice. However, there are some major problems with using voice, particularly related to voice recognition. In addition, voice is not a high-bandwidth communication modality and other alternatives might consequently provide better performance for entities such as robots that can handle high data rates.

Text-only

The voice-only interaction can be reduced to the text of each speech segment, producing a text-only interface. The text could also be augmented with other information that might not have easily been integrated into the voice-only interface.

Web-based

One method of providing a text-only interface is through the Internet using standard web servers and web browsers. An example of how such an interface might appear during the task selection process is shown in Figure 6.2. Standard web-browsing issues, such as forced refresh methods and data currency, would need to be addressed.

Handheld

Another method of providing a text-only interface is via one of the various handheld computers or cellular phones. These interfaces would have less available screen real estate than the web-based interface but some mission environments might benefit from their portability. Figure 6.3 shows an example of such an interface to an object-based interaction conducting the command formation process.

Direct manipulation

The voice-only commands can be further abstracted to a direct manipulation interface that requires a pointing device (mouse or stylus) and a graphics-capable computer screen. Direct manipulation interfaces such as the windows file management system have become

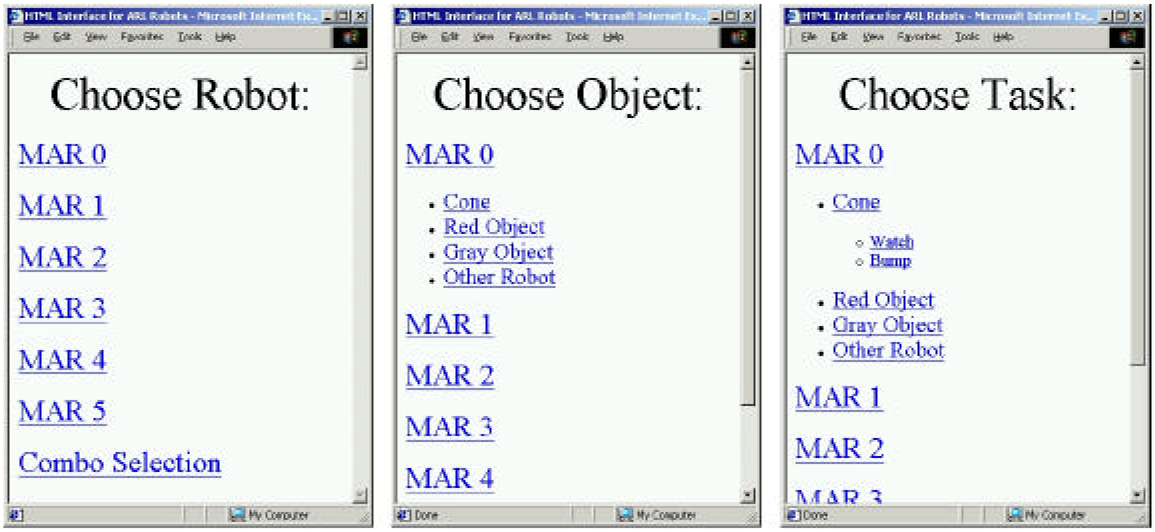

Figure 6.2: **Example Web-Based Interface**

This web-based GUI shows three consecutive screenshots. In the first screen, the operator selects MAR 0, the name of one of the robots. That link goes to a dynamically generated screen where the operator selects the Cone as the object on which to act. Finally, the operator is given a choice of the possible tasks to perform on the Cone using MAR 0.

very widespread and are familiar to many potential users. However, most direct manipulation interfaces make some assumptions about the underlying data that do not hold true for systems of multiple field robots. These assumptions will be explained below in detail.

Choosing an interface for experimental purposes

Many versions of the above interfaces could have been chosen to implement for the purposes of conducting experiments. The direct manipulation interface was chosen because it

- seemed to be most familiar to any operator,
- accommodated the “cultivation of common ground” operator role,
- allowed for both global model and multiple local model display,
- provided the most opportunity for further development, and
- would create the most attractive and impressive result, if successful.

Figure 6.3: **Example Handheld Interface**

This handheld GUI shows three consecutive screenshots that mirror that of the web-based GUI from Figure 6.2. In this case, the keypad is uses the keypad to make her selections.

6.2 Background information

Two topics of background information are useful for further discussion about the utility of direct manipulation interfaces for robot operation – other types of interfaces that have been used to operate field robots, and direct manipulation interfaces that have been used for applications other than robotics.

Interfaces for field robot operation

The number of interfaces for field robot operation is roughly the same as the number of field robots, since most robots have interfaces developed specifically for them. There is no standard interface in use for robots. Chapter Two provides detailed information about previous research and development of field robot interfaces including specific examples. A summary is provided here.

Most field robots have been teleoperated using rate-control joysticks and direct video feedback, from the earliest remotely piloted vehicles to today's underwater explorers or unmanned military vehicles. The interfaces are often the same as if the operator was inside the vehicle and might attempt to provide a sense of telepresence through head-mounted video displays or even physical motion.

Newer interfaces have been enabled through a greater degree of autonomy on the robot and have mostly been control panel or dashboard-like displays. These interfaces allow the output of many sensors to be monitored at one time, sometimes fused into an integrated view that could help the operator manage a greater amount of data. Some displays are also multimodal, giving the operator a variety of control mechanisms (e.g. gestures and voice) [112] and displays (e.g. text, haptic, graphic). Interfaces strictly built for supervisory control have typically included graphical displays, buttons, and other mechanisms to operate robots through the command of sub-tasks. Finally, some novel interfaces have been developed, such as gesture only [96], PDA-based collaboration [29], or web browser interfaces [113].

Direct Manipulation Interfaces

Since the development of the Xerox STAR, the first direct manipulation interface, the popularity of this interaction framework has spread tremendously. The most well-known direct manipulation interface is the "windows" desktop first popularized by Apple and then adopted by Microsoft. Today, people all over the world are comfortable with the ideas of "point and click" and "dragging and dropping" on-screen icons to accomplish tasks on the computer.

Direct Manipulation vs. Agents debate

A debate currently exists about whether or not direct manipulation is the appropriate direction for future interface development. The debate has been led by Ben Schneiderman at the University of Maryland, in favor of direct manipulation interfaces, and Pattie Maes of the MIT Media Lab, who prefers Interface Agents. Schneiderman believes that DMI provides comprehensible, predictable, and controllable methods for working with computers [114]. Maes, on the other hand, suggests that intelligent software could learn users' interest and act autonomously on their behalf to reduce the workload of the user, eventually until the proper actions take place autonomously [115]. Put another way, in direct manipulation a model of the task domain is presented to the user to interact with directly, while with an agent metaphor the computer is treated as an intermediary that responds to user requests [116].

Although both arguments are useful to consider when designing field robot interfaces, the arguments of Schneiderman appear stronger for the field robot application. In particular, the physical-ness of the results of field robot commands requires that the user have more direct control over what actions are taken rather than have the operator taken further out of the loop and replaced by autonomous decisions.

Real-Time Strategy games

As discussed previously, there is no standard interaction for the operation of multiple complex robots. However, a suitable interaction paradigm for the command and control of multiple complex robots has undergone extensive testing and refinement -- although only for simulated ideal robots. This interface genre, used most often in Real-Time Strategy (RTS) genre of computer games, has the following pertinent characteristics:

- Implements an interface between a superior commander and multiple autonomous subordinates
- Provides a third-person perspective of the environment as the dominant viewpoint
- Represents robots and objects as icons that may be selected singly or in groups for subsequent action
- Enables a combination of mouse gestures and keystrokes for operator input

The basic premise of Real-Time Strategy games is that one person is required to command many autonomous entities to achieve strategic goals. Popular examples of the genre include Strifeshadow (http://www.ethermoon.com) shown in Figure 6.4 and Starcraft (http://www.blizzard.com/starcraft/). Future robot applications could benefit from the existence of millions of operators with experience in this interaction paradigm. A Boeing-led study funded by the Department of Defense identified such an approach as the best method to overcome current human-system interface challenges [56]. The use of game interfaces to accomplish real tasks is not new; game interfaces have been used for such mundane tasks as operating system process management [117] and for many training uses in the military [118].

Figure 6.4: **Example Real-Time Strategy Game Interface**

This screenshot taken from the RTS game Strifeshadow shows the four main parts of the RTS interface: the main activity screen in the top two-thirds, status of a selected unit in the bottom left, a 3-by-3 choice of possible tasks for the selected unit in the middle right, and a global map in the bottom right.

6.3 Challenges with extending DMI to multiple field robots

Real-Time Strategy games are examples of a very appropriate implementation of the direct manipulation interface framework for use in multiple robot operation. In particular, the manipulation environment is dynamic even in short time scales, has a spatial relationship component, enables a task-level focus, and presents a world view to the commander. As such, it forms the basis for the best point of reference in how DMI interactions are currently implemented and where these implementations may succeed or fail for real robot operation.

While the implementations vary, most RTS games share the following characteristics:

- User interface and data representation are separate from the control architecture,

- Entire game state is managed as a single simulation; all entities operate within the context of this shared state and contribute no information to the simulation state, and
- Command architecture supports immediate as well as high-latency and non-guaranteed connections to enable networked play.
- Control protocols allow for shared commands and multi-entity interaction so that game objects can easily be instructed to collaborate on certain tasks.

A review of the typical field robot applications might compare favorably to a list of scenarios currently employed in RTS games. One might even reason at first glance that a practical human-system interface might be accomplished by simply "plugging in" a robot system to an RTS game architecture. However, further consideration reveals the following significant differences between the typical RTS and robot environments [119]:

- The real-world nature of robot systems does not allow the shared-state consistency required by the standard RTS environment,
- The sensors of the distributed robots will often disagree and send conflicting data that may not be readily integrated into the comprehensive world model required for RTS,
- Disturbances to the robots and objects in the world may be completely unobservable to the robots and, consequently, to the interface, and
- Relationships between robots, their capabilities, and objects in their world change dynamically.

These are all the result of the most fundamental difference between an RTS game and a robotic system – while the RTS game program creates its world internally and has complete authority over everything in it, the robot interface must construct its perception of an external world from imperfect sensors with limited ability to effect any change.

An RTS application has complete knowledge of the state of all robots and objects in the environment. There is no confusion about an object's identity or location. Many design and implementation issues are greatly simplified by this fact. In an actual robot system, the assumption that robots will sense infallibly and agree on everything is unrealistic. The practical outcome is that there is no conclusive global model of the world that the robot or operator can use for command purposes. Systems used to command robots must thus recognize that robots will disagree about fundamental aspects of their environment yet be able to handle these discrepancies adroitly.

Distributed model consistency

Because RTS games have complete knowledge of system input and system states as well as control over the speed of state changes, they may insure that all representations of the environment are consistent with one another. This allows for straightforward collaboration and coordination. Field robots, on the other hand, sense the environment individually and must communicate the state to one another. From physics first principles, this data is inconsistent with reality by the time it is received. Perfect consistency between the representations of the environment on the robots and by the operator would require the capacity for infinite and immediate communication. This consistency-throughput tradeoff is a fundamental challenge for all distributed systems [111].

Limited local sensing

The RTS games are able to provide complete global information about the entire state of the world to any robot that asks for it. The virtual robots can consequently see infinitely far away and be sure that they are aware of everything that is nearby. Real robots, on the other hand, have limited sensor ranges and cannot even be sure that the sensor is working with complete integrity within the expected sensor range. The interface nonetheless must rely on this incomplete information.

Object correspondence

RTS systems utilize a single data representation that serves as the sole source of information for all robots in the environment. As a result, objects can be uniquely identified and are easily referenced between robots. In a real field robot system, there is no such single data representation and the robots themselves must derive the location of the objects. As a result, for any sort of cooperation task or interaction, the object correspondence problem must be addressed.

Dynamic tasks

Finally, a valuable trait of a well-designed interface is that it only affords commands from the user that the robot is capable of satisfying. However, robot capabilities may change as engineers provide upgrades or as the realities of a deployment effect second-to-second robot state. This can be problematic, since robot capabilities not available on the interface are effectively nonexistent from the perspective of the operator, and capabilities provided by the interface that the robot cannot accomplish will be a source of frustration [52] and may lead to a reduction in trust and subsequent use [67].

6.4 Methods to address the challenges of DMI for field robots

Given certain assumptions about the robotic system and its mission, four innovations address the challenges outlined in the previous section: flexible data acquisition, maintenance of local models, the implementation of an object correspondence agent, and the use of tasking dialogues.

System assumptions

Robot capabilities

The assumed robot capabilities were the same as those listed in Section 5.5. Also assumed is that due to the adoption of the OBTLC architecture, the robot can accept task-level commands about objects of the form "Robot, (task) (object)."

Mission requirements

The major assumptions about the requirements of the mission are that it require coordination among the robots and with events beyond the immediate workspace of the robots, that the determination of the mission be dynamic so that a priori scripts or plans are only marginally useful, and that the tasks conducted by the robots be coupled with one another for successful completion of the overall mission.

Flexible data acquisition

Because RTS games are able to "stop time" as necessary to ensure that all distributed entities remain consistent world models, they are able to avoid the consistency-throughput tradeoff discussed in Section 6.3. All data updates happen simultaneously for all entities so that the entire world remains synchronized. For field robot systems, this is not possible. This research implemented a flexible data acquisition strategy in which a real-time publication/subscription scheme was used to communicate data. Entities that needed high levels of consistency could request frequent publications while others could reduce the overall bandwidth requirements by making infrequent requests. The interface was built to handle information input and output at any data rate and at mixed data rates between publishers by processing data through an event-handling process.

Tasking dialogues

As the potential for time-variant task models for the robots is significant, the interface needed to reflect these changes in as straightforward a manner as possible. The alternative implemented in most field robot interfaces, a static set of robot capabilities from which the operator can choose, is not acceptable in light of the usability principles discussed previously. The most direct solution is to simply ask the robots what its current capabilities are. This approach creates a dialogue between the operator and the robots and is the one taken by this research.

Dialogue with a robot is a very old topic of artificial intelligence, going back to Nilsson's classic SHAKEY robot in the 1960's. Much more recently, robotic researchers have developed many systems for *collocated* human-robot interaction using natural language interfaces, sometimes supported by a GUI [44]. There have been two projects to use some form of dialogue with a field robot – Fong [29] used dialogues to collaborate on object classification, and Zelek [47] directed a robot on navigational tasks. Dialogues have not been used with systems of multiple robots, and no interactions have integrated findings from observations in non-robotic environments of how humans actually use dialogue to accomplish goals cooperatively.

This research does not utilize the latest in natural language dialogues and related fields. Instead, it only implements what is necessary to highlight the insight that a simple dialogue between the user and each field robot is the atomic unit of interaction between these two entities. The term "dialogue" might be overkill, as the interaction is mainly a turn-based constrained query whose main task is to get at the database of information on each robot. The "dialogue" may take place invisibly without the explicit participation of the operator as the back end of the direct manipulation process.

Consequently, a review of the latest dialogue research is not particularly useful. Instead, definitions of a dialogue and an overview of their basic use follows. A thorough discussion of the fundamental challenges of the creation and use of natural language in computer software is presented by Winograd [120].

A brief definition of dialogue: the process of communication between two or more parties. Although most laypeople do think of a dialogue as a sharing of information (data, symbols, context), it is equally a sharing of control between the parties. This sharing of control – in particular, turn taking – is a key feature of the dialogue present in the human-robot interaction developed by this research (Section 7.1.7), and a key mechanism to develop a useful interface for a novice user [121, 122].

Some projects utilize the distinctive qualities of other languages to carry out dialogues, such as the word-ending particles in the Japanese language for the JiJo-2 service robot. This reference is included because it is important to remember to consider the international perspective of any 'dialogue' technology [123]. The dialogue format developed by this research should be extendible to any language but this assertion has not been tested.

Based on the observations of the SWAT TC's dialogues with the tactical teams, the goal of the dialogue is to create valid and relevant commands of the form "Subject Verb Direct Object," such as "Robot A grab Component 1." This is an extension of a well-known method for the operation of single robots based on a conceptual command of "Put that there [27]." The limitation on possible actions in the "Put that there" method – one can only ask the robot to move an object – is both positive and negative, since it means that the operator has little confusion over what actions are possible by the robot, yet is given no choice in the matter. The new method proposed by this research provides the possibility of many more verbs than simply 'to put' and replaces the location focus with a physical object focus, creating a new paradigm for operation that can be succinctly described as "Do what to whom?"

The electronic dialogue starts when a robot (or robots) queried by the operator generates a list of the types and positions of objects it senses. The operator then selects, by clicking, a particular object with which to work. The interface requests a list of tasks that the robot can enact on that object, and this list, when returned, is displayed by the interface. The operator selects a task and the command – valid and relevant – is sent. In effect, the operator is assured of a consistent set of affordances for solving mission problems and accomplishing tasks. An example of this interaction is shown in Figure 6.5, with a screen shot of the computer interface on the left and a text representation of the electronic dialogue on the right.

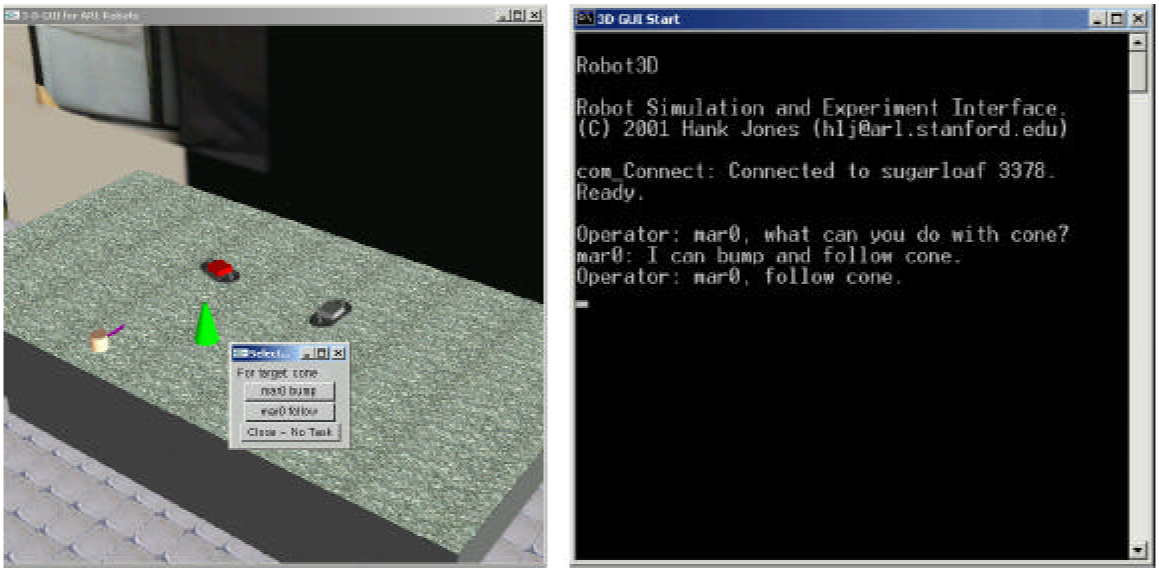

Figure 6.5: **Electronic Tasking Dialogue**

The left screen shows what is displayed to the operator, while the left screen (the last three lines in particular0 shows the electronic dialogue that is being conducted in the background.

The result of this tasking dialogue is the presentation by the robot – through the interface – of accurate, feasible, context-sensitive affordances to the user. Affordances, according to Gibson, are what the environment offers, provides, or furnishes [124]. Applied to field robotics, affordances are the clear options of possible commands that may be given to the robot. The notion of providing affordances at this level and in this sense was discussed by St. Amant regarding artificial intelligence planning algorithms, but the concept of dynamically altering the available affordances has not been implemented on a complete robotic system prior to this dissertation work [125].

Maintaining local models

In the observations of SWAT teams, the tactical commanders constructed a mental model of the current situation and environment from the initial information and subsequent updates from the teams, thus integrating many different perspectives to form an overall world view. In these SWAT situations, such a process is the only way to construct a global model as no external sensor, such as a camera or a position reporting system, is available.

In contrast, existing multiple robot systems currently rely on access to a global model of the world to allow the robots to interact, cooperate, avoid conflicts, and otherwise function

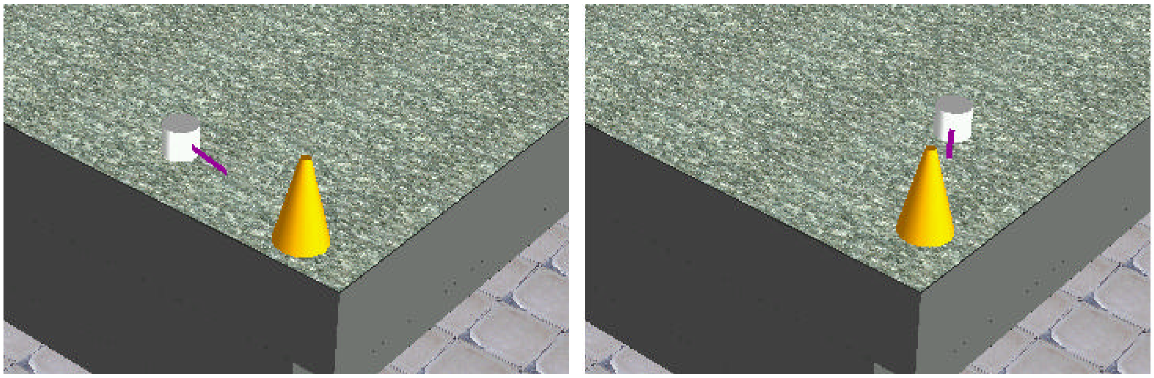

Figure 6.6: **Perception Conflict**

Two robots see the cone in two different locations; the robot on the left sees it near the left edge of the table, while the right robot sees it a few centimeters to the right. In reality, there may be either one cone on the table or two cones in close proximity to one another.

harmoniously. To accomplish this, a remote sensor is used to determine the position of the robots, objects, obstacles, and the structure of the environment [31, 32]. However, for field robot operation, such a sensor is not likely to be available.

This provides a more realistic system, but not without cost. As shown in Figure 6.6, two robots will typically sense their environment slightly differently. To provide a clear example, the two robots are shown only with the cone each senses and not their sensed location of the other robot. The same problem was observed during the SWAT observations, when members of the distributed team referred to the same objects (e.g. the staircase used by the two teams in Exercise 3) from different perspectives. The Tactical Commander resolved this issue by maintaining in his head distinct and separate models that represented the perspective of each team member. The direct manipulation interface handles this problem by displaying all data as it is provided but storing it according to its source. The data can be displayed in many ways without losing the important information contained by association with its source.

Correspondence Agent

The maintenance of distinct local models and the absence of a global information source presents a problem for conducting operations involving more than one robot. In the example shown in Figure 6.6, there are, conceptually, two cones in the world – one sensed by

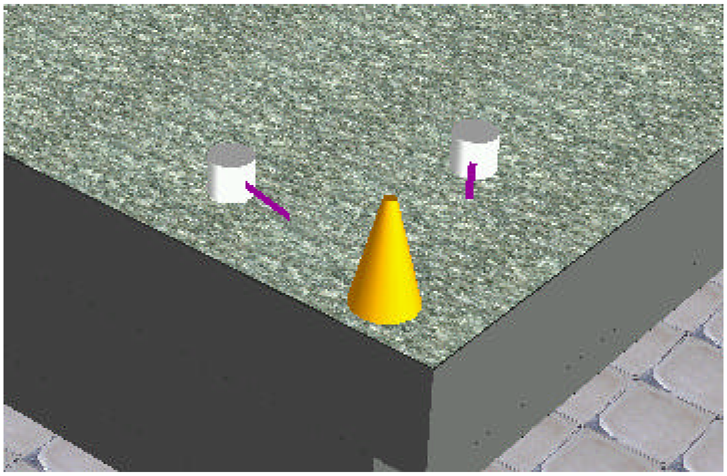

Figure 6.7: **Correspondence Example**

The two robots cannot cooperate unless they agree about the correspondence of objects that each perceives. Here, the interface reflects the assertion by a Correspondence Agent that the cones sensed by the robots are actually the same object.

each of the two cylindrical robots. This research addressed this challenge by introducing an autonomous software agent called the Correspondence Agent (CA). The CA listens to status reports from the robots about objects that are sensed and makes determinations of which objects correspond to which others – basically forming ties between the worlds based on the objects being sensed, just like the SWAT tactical commander. This agent was automated because of the likely scale of the task, which for even a small system of five robots and four objects can require the analysis of up to 1,000,000 (sum of the number of objects and number of robots, to the power of the number of robots) possible correspondence pairs.

The effect of the Correspondence Agent is shown in Figure 6.7. On the left, with the CA turned off, two cones are presented to the operator. When the CA is enabled, the interface is informed that those two objects are actually the same, and the interface presents only one. The operator can now select the two robots and give them commands for cooperative tasks on the cone. Most importantly, the interface will send out commands <u>in each robot's own frame of reference</u> using the cone that it senses, not the unified cone shown to the operator. Again, this is the reason for maintaining local models and devising a Correspondence Agent – to be able to provide a straightforward single model interface to the operator for ease of use, but within an infrastructure that reflects the realities of the robotic system.

One should note that this interaction does not require a 3-D GUI like the one shown in these figures – robot operation based on the same electronic dialogue and using a Correspondence Agent could be conducted strictly via the text-based cell phone or web browser discussed in this chapter's first section.

Chapter Seven

Experimental Demonstration

This chapter describes the novel robotic system that was implemented as the final step in the design process outlined in the first chapter (Figure 7.1), and it presents the results of using this system in an experimental setting. The first of the two sections provides details about the experimental setup, including the robots, the software agents, and the graphical user interface. The robots that were used make up the Stanford Aerospace Robotics Laboratory Micro Autonomous Rover (MAR) testbed. The MAR system consists of five small non-holonomic robots that are directed wirelessly to move about a flat surface. They are capable of various complex tasks and can interact with physical objects in their environment. The software agents include a Correspondence Agent, a Query Agent, and a Planning Agent. The graphical user interface (GUI) utilizes the OpenGL computer graphics library to provide a three-dimensional interactive representation of the experiment environment.

The second section of the chapter describes the demonstration missions carried out using the experimental system and the strengths and limitations of the system that these experiments revealed. The system was shown to allow the operator to participate actively in mission planning through problem solving and task coordination. The lack of a real-time planner to assist the operator with multiple-robot tasks constrained the effectiveness of this implementation.

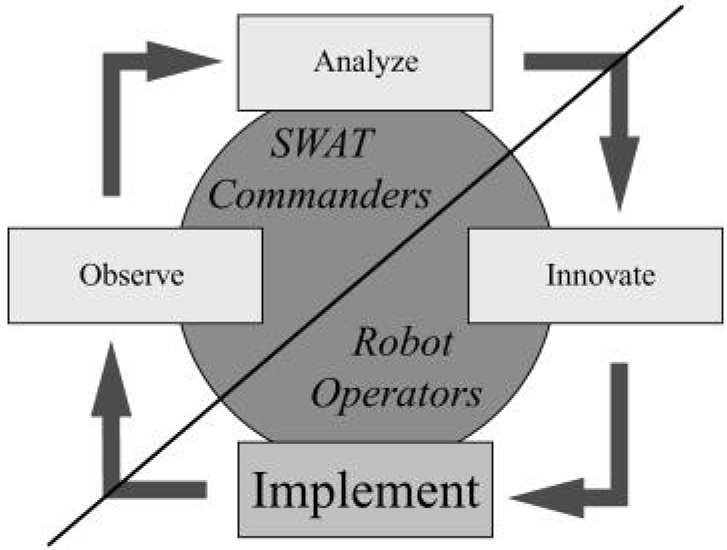

Figure 7.1: **Step Four – Implement**

7.1 Experimental setup

This section describes the experimental system that was used to confirm the feasibility of the hybrid setting design process and the use of Object-Based Task Level Control for multiple-robot missions.

7.1.1 System overview

A basic overview of the experimental system is shown in Figure 7.2. The system is made up of a community of agents, including the Interface Agent to communicate to the operator and Robot Agents who represent each robot. The robots sense objects in the world (see Section 7.1) and communicate this information through the Robot Agents to the community. An OAA Facilitator program moves specific requests and replies between agents, and the NDDS data infrastructure carries state update data packets, which occur at a much faster rate. The robot state data from the overhead vision system passes through the robots to the GUI, so that an architecture appropriate for field robots is retained.

With any experiment, many theoretical and operational issues must be considered. However, only a subset of them will be addressed directly by a given experimental setup. The rest will be simplified through helpful assumptions or altogether ignored. The next

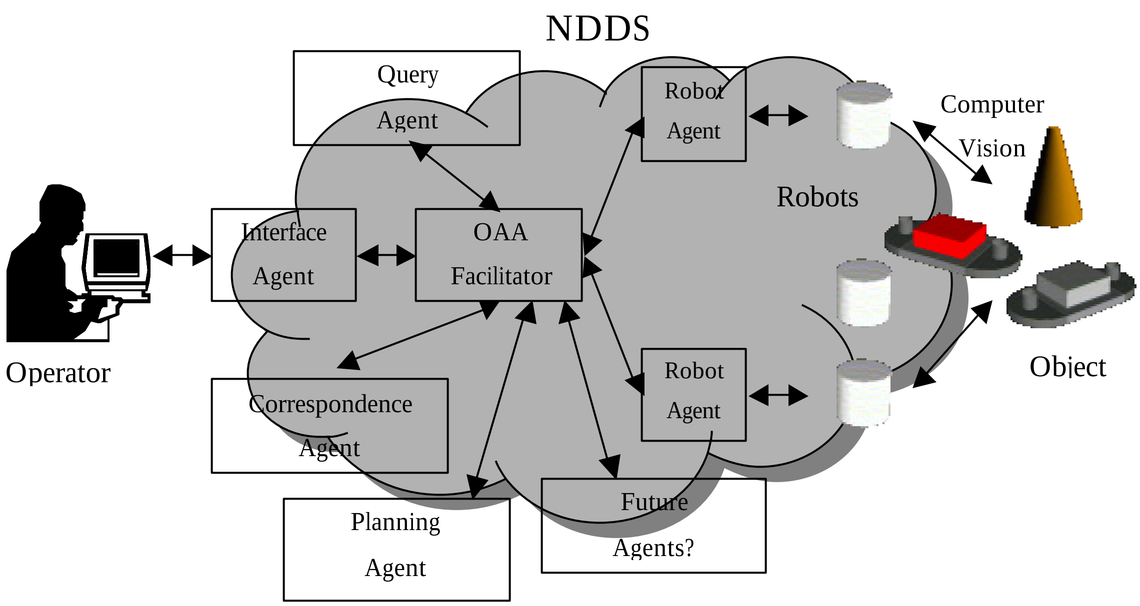

Figure 7.2: **System Overview**

The NDDS middleware layer enables a publish/subscribe communication backbone between agents for information with high update rates. The OAA facilitator handles specific inter-agent requests. The robots sense objects in the environment via computer vision (either onboard or overhead) and pass this information into the system via their respective Robot Agents. The operator gains access to the environment through the Interface Agent.

subsections describe the important issues considered in the context of this experimental demonstration and how they were approached.

Issues addressed

A set of core issues was directly addressed in this experimental demonstration. These issues reflect a focus on integrating the core lessons learned from the SWAT command analysis:

- No global sensor/global model – All information for field robots, by definition, comes from the robots themselves. The experimental setup routed the information from a global vision system through the robots.
- Limited local sensing – Field robots all have limited sensor range. Although the overhead vision system detected all marked objects in the experimental environment,

each robot simulated local sensing by restricting the acquisition of data to the area near itself.

- Correspondence – Objects sensed in the world are not uniquely marked and thus multiple robot systems must determine which objects correspond to those sensed by the robots. A Correspondence Agent was created to address this need.
- Dynamic capabilities – The tasks that the robots are able to perform change depending on the state of the robot. These robots used a first-order theorem prover to determine what tasks were possible for an object given the object type and the robot state.
- External information integration – The human operator must often incorporate new, novel, or unexpected data into the mission requirements for the robots. Using the GUI, the operator could direct the robots according to information outside the sensing capabilities of the robots.
- Communications bandwidth usage – Field robots usually have limited communication bandwidth, and this system took a number of steps to reduce the bandwidth requirements between each robot and the operator.
- Noisy and biased sensing – Robots that rely on relative sensing typically produce noisy sensing data that also may develop biases over time. The environment sensing used by the robot, provided by an overhead vision system, was moderately noisy. The vision system did not typically develop a significant bias, so an artificial one was added by the robots.

Issues simplified

Some issues were simplified by applying reasonable assumptions or simulating more advanced robotic capabilities than were actually implemented:

- Path planning – Although this system sometimes used a motion planner developed by Chris Clark in the ARL [126], path planning generally was kept very simple. Even with the Clark system in use, only entities detected by the vision system were avoided while unmarked objects were truly hazards. Since more comprehensive algorithms for safe path planning exist and the robots were not able to harm themselves or others, the use of this simple path planning was sufficient.

- Self-monitoring – The robot self-monitoring information such as fuel and camera status utilized by the first-order theorem prover was simulated by manual monitoring by the experimenter. Robust self-monitoring is a current area of research and its results could be readily integrated into this system in the place of the manual simulation.
- Object sensing/identification – Object sensing and identification was carried out by an overhead vision system. LED identifiers uniquely marked objects. Object sensing and identification is a significant issue being addressed by many researchers, and their work could be applied directly to this system.
- Position sensing – The robots utilized the overhead vision system rather than determine their position independently. This information was slightly noisy and the robots added some bias to the data to increase its realism.
- Atomic task capability – The number and character of basic field robot capabilities varies widely. This experiment sought only to provide a reasonable variety without particular concern for the tasks themselves. With the MAR system, capabilities are limited to those that only require moving about. To accomplish a manipulation task, the MAR collided with a lightweight object, which resulted in object motion.

Issues ignored

The benefits of addressing some issues were weighed against the roadblocks they might cause and whether their absence fundamentally altered the nature of the experiment. The following issues were judged to be sufficiently secondary to the primary research needs that they could be ignored:

- Computational power limitation – Field robots are usually limited by computational power available. In this system, the computing requirements of the field robots are reasonably low, but no particular measures were taken to minimize or otherwise examine the necessary computational power.
- Robustness – For many field robot deployments, robustness to the potential failure modes is the most important feature for success. The steps necessary to provide robustness for the MAR system would have been very time-consuming and would not have contributed to the outcome of this research.

- Cooperative tasks – One of the most powerful arguments for the use of multiple field robots is the ability to conduct cooperative tasks. Such tasks are the subject of current research and could be integrated into this system with some minor extensions. They were not included because of the significant additional complexity and development time they would have caused.
- Mission planning and management – Some of the tasks carried out by the human operator might be more efficiently planned and carried out using an automated mission manager. At some level, however, the human must be involved in directing the mission. To that end, a mission planner was not used although the operator would conceivably interact with it in the same way as she did with the robots themselves.
- Power and Energy Optimization – Field robots must usually be concerned with their power consumption and energy management. Such issues affect the ability of the robot to perform various tasks over time. The effects of these dynamic capabilities were addressed by this research, but the power and energy management was simulated.
- User efficiency – Each particular field robot deployment will have its unique set of operator requirements and operational tempo. The efficiency of the user is important, but analysis of her efficiency when using this experimental system would not have provided generic results.

7.1.2 Infrastructure

The experimental platform is a part of the research infrastructure of the Aerospace Robotics Laboratory at Stanford University.

Laboratory equipment

The foundation for the experimental setup is a large high-quality granite table located in Room 010 of the Durand Building at Stanford University. The table dimensions are 2.4 meters by 3.7 meters. Mounted in the ceiling over the table are three cameras used for the overhead vision system described in a subsequent section.

Network Data Delivery System

The network backbone for fast data rate system communication was the Network Data Delivery System (NDDS) (http://www.rti.com/products/ndds/ndds.html) produced by Real Time Innovations (RTI) (http://www.rti.com). NDDS is network middleware that simplifies the development of distributed, real-time applications. This package uses a publish/subscribe architecture to create a virtual real-time data bus over a TCP/IP network. In a publish/subscribe system, data-creating programs publish NDDS packets of a given name and type, and programs that need data subscribe using the same name and type. A central manager administers the publication and subscription requests. The NDDS publish/subscribe network allows data sharing among multiple applications on heterogeneous platforms, and provides an elegant method for publishers or subscribers to enter or leave a system [127].

Open Agent Architecture

A well-defined structure is often beneficial for exchanging information and knowledge in distributed systems. Agents (typically defined as any software process that meets the conventions of a particular agent society) are an increasingly common way to conduct this information exchange. Standards bodies such as the Foundation for Intelligent Physical Agents (FIPA) work to provide standards so that heterogeneous agents can communicate with one another [128]. One of the most well known such standards is the Knowledge Query and Manipulation Language (KQML). A programming structure in widespread use, based on KQML, is the Open Agent Architecture (OAA) developed at SRI. This research utilized KQML and OAA for a substantial part of its necessary communications and database management infrastructure, particularly for the electronic dialogues between agents which were subject to frequent change throughout the system development.

The Open Agent Architecture (http://www.ai.sri.com/~oaa) developed by SRI International (SRI – http://www.sri.com) is used as the software agent communication infrastructure. OAA is an open, extensible framework that allows distributed agents to communicate either directly or anonymously. Software agents can pose queries to the OAA society and receive answers from one or all possible respondents. Use of OAA requires a facilitator as the central location for managing agent requests and responses.

Figure 7.3: **Micro Autonomous Rover Testbed**

The robots are the cylindrical objects in the foreground; the objects that are manipulated are in the back row.

7.1.3 Hardware

The hardware used in the experiments consists of a group of robots and a set of heterogeneous objects, shown in Figure 7.3.

Robots

The Micro Autonomous Rovers (MAR) system is made up of five small, wheeled cylindrical robots. Each robot (sometimes referred to as a "micro rover") is independently driven by standard radio-controlled (RC) servomotors that have been modified to follow speed, rather than position, commands. Two wheels are mounted underneath the robot on the left and right sides of the robot. Two small posts with Teflon caps are mounted perpendicular to the wheelbase in order to balance the vehicle.

Four of the five robots are 9.8 cm tall and 10.2 cm diameter. A fifth robot, shown in Figure 7.4, has an additional wireless camera mounted on top that adds to its height. The camera is an Xcam2 wireless color camera integrated with a 2.4GHz wireless transmitter. The camera is mounted such that is looking slightly downward in the direction of positive robot motion. For the most part, this camera was not used in this research.

A thin Teflon case covers the robot interior for protection and cosmetic reasons. The case hides a 4-cell 4.8 V NiCad battery and the radio-control receiver, whose antenna exits

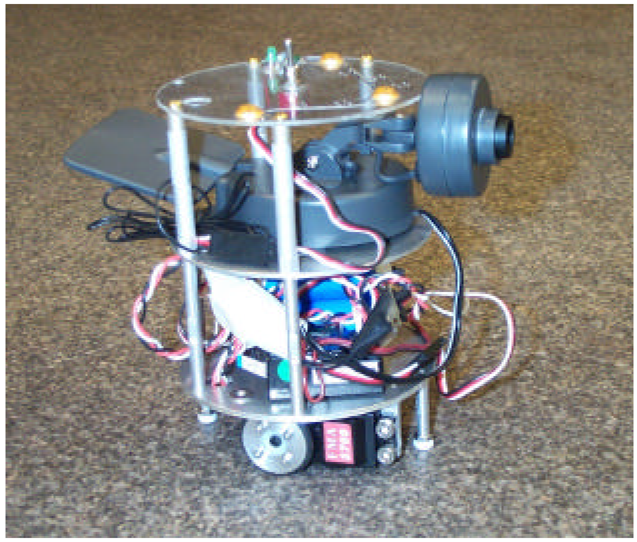

Figure 7.4: **Micro Autonomous Rover with Onboard Camera**

the vehicle at the top and hangs by its side. The robot is topped with a Plexiglas plate that holds three infrared LEDs that are used by the overhead vision system to sense the vehicle's position and velocity. All vehicle processing and control is done off-board and is described in a subsequent section.

The robots operate on a large granite table, although the available workspace is limited by the range of the overhead vision cameras to an area of 2.0 by 3.7 meters.

Objects

Four passive objects exist on the table – one orange plastic cone, one "discus," and two "floaters." The cone has been fitted with a small wooden plate on top to hold an LED pattern, but otherwise has not been altered. The discus is a round metal plate of significant mass with a small air pump contained in a box on top with an LED pattern attached to the box top. The floaters are oblong plates of much less mass, also with air pumps in boxes on top with their own unique LED patterns. One of the floaters has a red box (the "red object") and one has a gray box ("the gray object"). The air pumps on the discus and the floaters are to able to cause the objects to float basically frictionless on the tables by directing air through small holes in the bottoms of the objects, but this functionality was not necessary for this research. Without the air pumps on, the discus and floaters are unmovable by the MAR robots. The cone, however, easily is moved if a robot runs into it.

7.1.4 Sensing

The robots primarily used an overhead vision system for determining the state of themselves and the objects, but an onboard vision system was also tested.

Overhead Vision

The position, orientation, and velocity of the micro rovers are sensed by an overhead vision system. Three Pulnix 440 CCD cameras are mounted directly overhead on the ceiling, looking downward on the table. Their views of the table overlap and provide full sensing of the robot workspace. They each have an infrared filter that allows them to only see the infrared LEDs mounted on the robots.

The outputs of the Pulnix cameras are fed into a Matrox Meteor 2 board on a Pentium III 1GHZ computer running Windows 2000. The three black and white signals from the cameras are input into the vision-processing computer on the red, green, and blue (RGB) channels of the Meteor board.

The vision processing software consists of several subsystems. The first splits the digitized image into RGB components, which actually correspond to the black and white channels of the three different cameras. Because the three Pulnix cameras have infrared filters mounted on them, the only objects in the images are the infrared LEDs on the robots and objects. Once the image is digitized, the LED blobs are segmented and their locations within the image are passed to the next processing section. The LEDs are then located within each image and the unique LED groups are identified. The vision software knows these patterns and is able to recognize and distinguish between them. Once the LEDs have been grouped, the position and velocity of the associated vehicle can be determined.

The overhead vision system outputs robot position and velocity information through the NDDS network backbone at 30 Hz. The LEDs can be located within the image with an accuracy of approximately 1/4 of a pixel. The main source of error comes from lens distortion and the camera calibration used to project the vision measurements back into world coordinates. Using a grid of LEDs, a third order polynomial fit is used to correct for the lens distortion. The system is able to achieve global accuracy of approximately 5 mm across the entire table.

Onboard Vision

For a small number of trials, the onboard vision system of the robot fitted with a camera was used to determine the position of another robot. The straightforward integration of the onboard system into the interaction developed by this research, described more completely by Frew [129], confirms that the use of the overhead vision system did not create any unintentional or unrealistic dependencies on the overhead vision system.

7.1.5 Robot software

The software necessary to control the MAR robots is composed of four parts: a controller to handle low-level motion commands, a Robot Agent to decide which task capabilities were possible, a "nervous system" to provide trajectory commands to the controller and process communication between the robots and the rest of the system, and a motion planner to provide obstacle avoidance.

Controller

All control processing for the micro rovers is done off-board on a dual Pentium III 550MHz computer running the RedHat Linux 7.3 operating system. Each robot is controlled by a separate process that runs on the control computer. The control signals for each robot are output as voltages that are converted to pulse width modulated signals (PWM) and sent via a radio link to the individual robots. The conversion electronics were developed at the ARL. Each robot uses two of the 12 total channels broadcast by the controller transmitter. By using commercial radio-control components, the micro-rovers can also be driven by hand using a standard commercial radio-control transmitter.

The servomotors on each robot have been converted from position to velocity control. By driving each wheel independently, the speed and angular velocity of the vehicle can be commanded. System identification was performed in order to derive a map between servo voltage and wheel speed. The effects of battery voltage were not included in the system identification so the performance degrades as the battery is drained.

Several different modes have been developed to control the rovers. A simple LQR regulator is used to keep the vehicle stationary or to turn in place. A nonlinear algorithm is used to follow smooth paths. All commands to the robots come with time parameterization.

Robot Agent

The Robot Agent (RA) is a text-only Java application that wraps the Java Theorem Prover first-order theorem prover (http://ksl.stanford.edu/software/jtp/) by Gleb Frank at Stanford University to allow it to work in an OAA agent society. The JTP loads a Knowledge Interchange Format file at startup that defines the relationships between the robot's states, its basic capabilities, and object types. An instance of the Robot Agent application is run for each robot, and the KIF file for each robot is unique to create heterogeneity between robots. When queried, the Robot Agent responds with a list of true statements. For example, when queried jtpask('(mar0 ?x redOB)') or literally "What can MAR #0 do to the red object?" the reply might be '(mar0 watch redOB)(mar0 locate redOB)(mar0 guard redOB)'. A screenshot of the Robot Agent providing this query response is shown in Figure 7.5.

Nervous system

The "Nervous System" (NS) software application is responsible for being the central point of communication for each robot and for generating the trajectories that carry out the robot tasks. The user interface for the Nervous System, used for testing and debugging purposes and not robot operation during the experiment, is shown in Figure 7.6. The NS subscribes to the data from the overhead vision server and gathers information about the state of the robots and the objects in the workspace. The robot then takes this data, adds source information identifying it for the GUI, and publishes these new data packets. A file loaded at startup provides the list of objects and other robots that a particular robot can sense.

The "Relative" checkbox on the Nervous System user interface tells the NS to add bias to the state data it publishes. A random seed is determined at program start and a different random bias is then added to each object's state. This step simulates one of the effects of the use of relative sensing by a field robot.

The Nervous System program can also examine the state of objects and other robots and compare them to a simulated sensor range. For most experiments, this range was an angle 40 degrees wide and 2 meters long extending from the front of the robot. This field of view approximates that of the camera mounted on one of the robots. The sensor limits could be turned on or off using the "Limited" checkbox on the NS user interface.

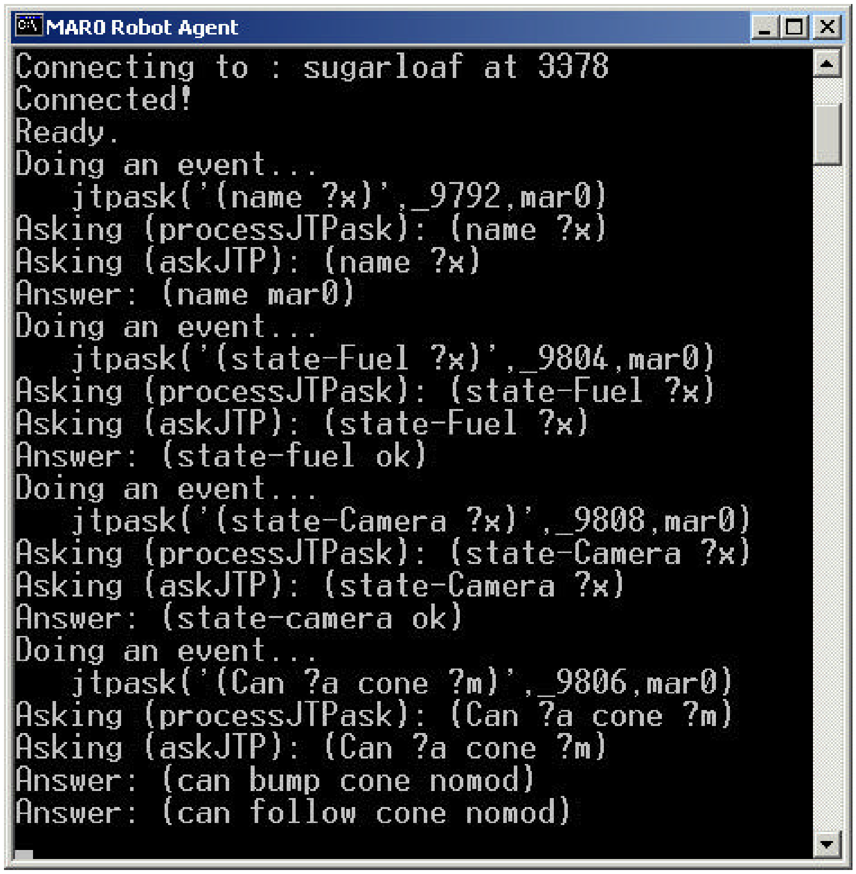

Figure 7.5: **Robot Agent Screenshot**

The Robot Agent answers requests of the form "jtpask('(QUERY)', variable placeholder, Robot Agent name)" and sends back an answer or answers that are true for that query.

For a small number of trials, the motion planner developed by Clark [126] was used to conduct robot motion while avoiding obstacles. This motion planner is based on randomized path planning, which does not provide optimal routes but does enable real-time motion and operation to take place. The trajectories created by the motion planner are arcs, rather than the straight lines created by the Nervous System program. The "Ext Plan" checkbox indicated whether the Clark planner was in use.

Two checkboxes on the Nervous System user interface reflect the fuel and camera states of the robot as determined by the Robot Agent. The user could change these states within the Robot Agent by clicking on these checkboxes.

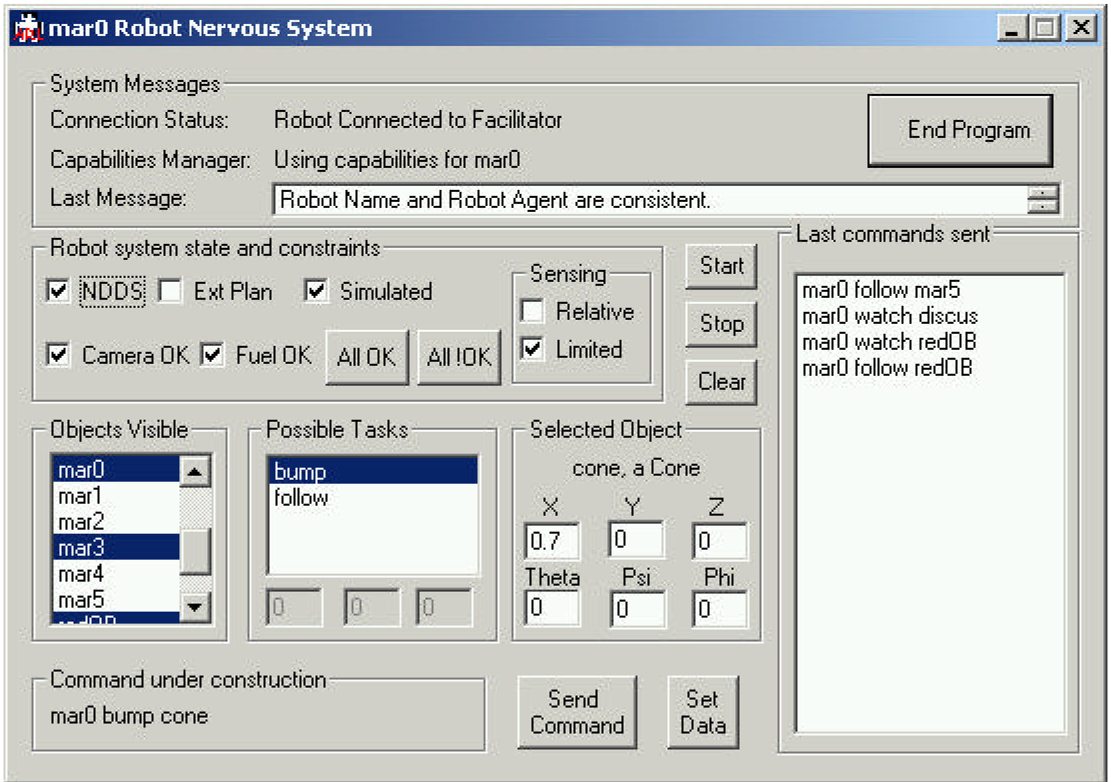

Figure 7.6: **Robot Nervous System Screenshot**

This application was used to manage the functionality of the robots during development and subsequent experimentation. This "engineering interface" was not used to operate the robots directly.

Robot tasks

The Nervous System is also responsible for managing the execution of tasks by the robot. Whether or not a task is currently possible given the capabilities of the robot and the type of object is determined by that robot's Robot Agent, which provides a list of these possible commands when queried. Consequently, only tasks that should be able to be executed are commanded to the NS. These commands typically come to the robot from the operator by way of the user interface, but could also be requested by an automated mission planner.

The tasks are all developed as subclasses of a CTask( ) parent class, which has the basic functionality necessary for all tasks. The Nervous System application has a placeholder for an active task of the CTask family, and the appropriate child class is instantiated when a command is received.

Translating

To move from one spot on the table to another, the robot must follow a three step process: rotate in place to the direction of travel, move forward in a straight line to the final position, and then rotate in place to the final orientation. This basic functionality was built in a CTaskTranslate( ) class, and then more specific actions could be developed by inheriting from this class. The Robot Agent application ensures that the fuel state of the robot is OK before any translation is allowed.

- Move-to: possible when the object selected is the tabletop, simply moves the robot to the location specified.
- Return-home: provided when the selected object is the robot itself. This task instantiates a Move-to task with a predefined location on the edge of the table where batteries were recharged. This task was most useful to the experimenters at the end of an experiment.
- Bump: possible if the selected object was the plastic cone. In this case, the robot would go to the sensed location of the cone, consequently bumping it out of the way. This was a very entertaining and popular task for most new operators.
- Follow/Guard: possible for any entity (robot or object) if its camera state was "OK," a move was requested to a position 0.4 meters away from the entity towards the center of the table. If the object moved more than 0.3 meters, the robot would be told to move again. At the end of each move, the robot turned toward the object being followed. This task was sometimes called "Guard" to agree with the context of the mission description.

Rotating

Tasks that required only rotating in place were subclasses of the CTaskRotate( ) class, which implemented a smooth rotation trajectory. The Robot Agent allows rotation tasks to be executed even if the fuel state is not OK.

- Watch: Any robot or object on the table might be watched, meaning that the robot would stay in place but continuously turn towards the watched entity. There was a dead band of 10 degrees to prevent constant movement by the robot. Watch tasks were only possible if the state of the robot's camera was "OK."

- Pan-CW / Pan-CCW: If the object selected was the robot itself, Clockwise or Counter-Clockwise Pan tasks would be offered. The pan would continue until the operator told the robot to stop. This was useful when the sensing range was limited and the operator needed to learn what objects surrounded the robot.

Other

Experience through the first steps of the implementation showed that a few other capabilities were required to facilitate robot operation.

- Halt: If the robot needed to be stopped quickly, the Halt command could be used. This set the desired state of the robot to its current state, which halted its movement but did not turn off the closed-loop control of the robot.
- Get-help: When the robots were being operated from another room, some method of communication with the experimenter was necessary. The NS was run on a computer near the robots, with the user interface readily seen by anyone nearby. The Get-help task, in this implementation, caused the NS user interface to flash. The experimenter's attention would be attracted by the flashing so that he could then take appropriate action. This task is also an example of how a multiple robot system might be run in real life, when an operator responsible for many robots might send problem robots to a troubleshooter.

7.1.6 Helper Agents

Through the course of developing and testing the system, and in comparing the result with the existing commercial productivity and game interfaces, many extensions were identified that would be very helpful for the user. Some of these improvements require some limited intelligence that did not necessarily need to be confined to one particular application, such as the GUI. Due to the nature of the use of OAA and the NDDS publish/subscribe system, it was easy to add agents to accomplish tasks.

Correspondence Agent

The analysis of the SWAT teams revealed that establishing common ground between the distributed members provided a critical foundation for coordinating action. This is true with multiple field robots as well, as the effects of noisy and biased sensing often result in disagreement about the location of objects. In human-only teams, information only comes when asked, and is typically filtered to only include new information. With robots, however, the amount of information about the environment is repetitive, can be extensive, and could completely overwhelm the operator. Therefore, a Correspondence Agent was create do assist the operator by continually merging information from the robots about the locations and identifies of objects. The processing would be useful for the operator as well as for the robots when performing cooperative tasks.

Determining correspondence is not a trivial task. Just to determine if a pair of two objects are actually the same requires a number of comparisons on the order of N^R where N is the number of robots and objects and R is the number of robots. For this system, with five robots sensing up to four objects each, using pairs was not acceptable from a computation perspective. In addition, all five robots may see the same object, making pairs less useful as the information storage unit.

Correspondence algorithm

This research took the approach of forming artificial entities called groups that could hold from one to any number of objects. Each object was then just compared to the existing groups, and added if it should be included or otherwise used to start a new group. In this way, the order of the correspondence problem was typically reduced to the order of R*N. This was much more computationally tractable.

The rules used to determine inclusion in a group are dependent on the details of the robotic system, and a very wide range of variables could be used. For the Correspondence Agent implemented here, entities of the same type were compared by their reported positions and compared to a difference threshold. The orientations of the entities were ignored. This simple set of rules worked adequately for this environment.

Manually asserting correspondence

In some cases, the Correspondence Agent would not automatically determine a correspondence that the user could readily see should exist. In these cases or others where correspondence could be determined outside the CA, a method for externally asserting correspondence was needed. The Correspondence Agent subscribed to a particular NDDS topic for correspondence assertions that would link two objects together. Whenever one of these packets was received, the two objects were put into the same group. If both objects were already in groups, these groups were merged.

Query agent

The Query Agent (QA) was developed to maintain lists of capabilities possible to perform on a given object. The Query Agent learned this information through eavesdropping on electronic exchanges, typically between the Robot Agents and the Interface Agent. If the agent did not have any information about an object, or if the information was deemed old, the QA itself would initiate a conversation with the Robot Agent. This ensured that the Query Agent had reasonably recent information, although there was no guarantee that the information would not be obsolete. For the most recent information, the operator needed to ask the robot explicitly.

The Query Agent was queried by requesting information about a given object referred to by the ID number given it by the robot that sensed it. The QA would then publish a list of tasks that were possible on that object and the relevant names of the robots capable of carrying out those tasks. An example of the result of a Query Agent interaction is shown in Figure 7.7. The dialog box shows robot and task combinations from which the operator could choose.

In many cases, the object was a member of a Correspondence Agent group. The Query Agent subscribed to all CA publications and used the same group-based internal data structure. Thus, when a query came in about one object, the results consisted of all the tasks possible for all robots that sensed what the Correspondence Agent considered to be one object.

Planning agent

To assist with the execution of compound tasks by multiple robots simultaneously, a Planning Agent (PA) was developed. Mission or complex task planning is a diverse topic with many possible options for planner implementation. For this system, a simple planner that looked for a small set of preconditions to ascertain that a given complex plan was possible. When queried, the Planning Agent would respond with a list of currently possible complex plans given the current robot and object states. When the Planning Agent was directed to carry out one of the plans, it would send the appropriate command simultaneously to the robots involved. The conduct of the plan was not monitored; the

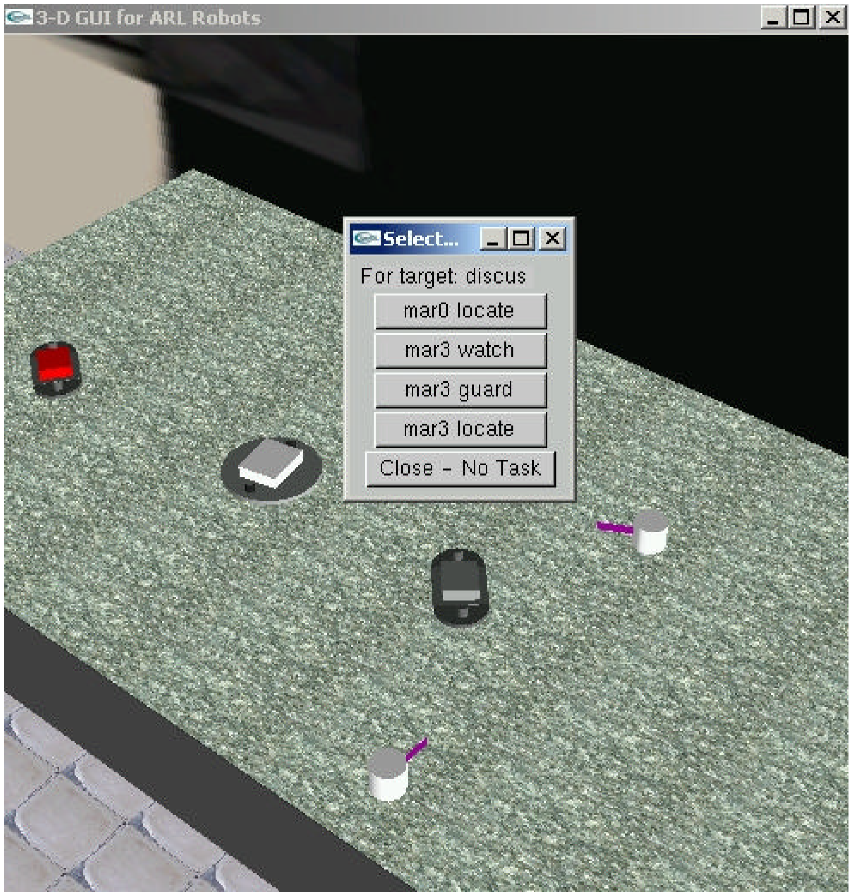

Figure 7.7: **Query Agent in Use**

The operator engaged the Query Agent when clicking on the discus object, resulting in a list of the tasks available to be selected. The robot that would carry out each task is listed next to the name of the action.

planner acted in an open-loop manner. The use of a Planning Agent in this sense was analogous to the training step of the SWAT commanders' efforts to establish common ground.

7.1.7 Interface Agent

Figure 7.8 shows the Graphical User Interface (GUI) that is presented to the operator. The objects on the screen are manipulated by the user to coordinate robot activity. The MARs are the small white cylinders. The other entities are mobile or immobile objects. Further description of the objects on the screen is given in upcoming sections.

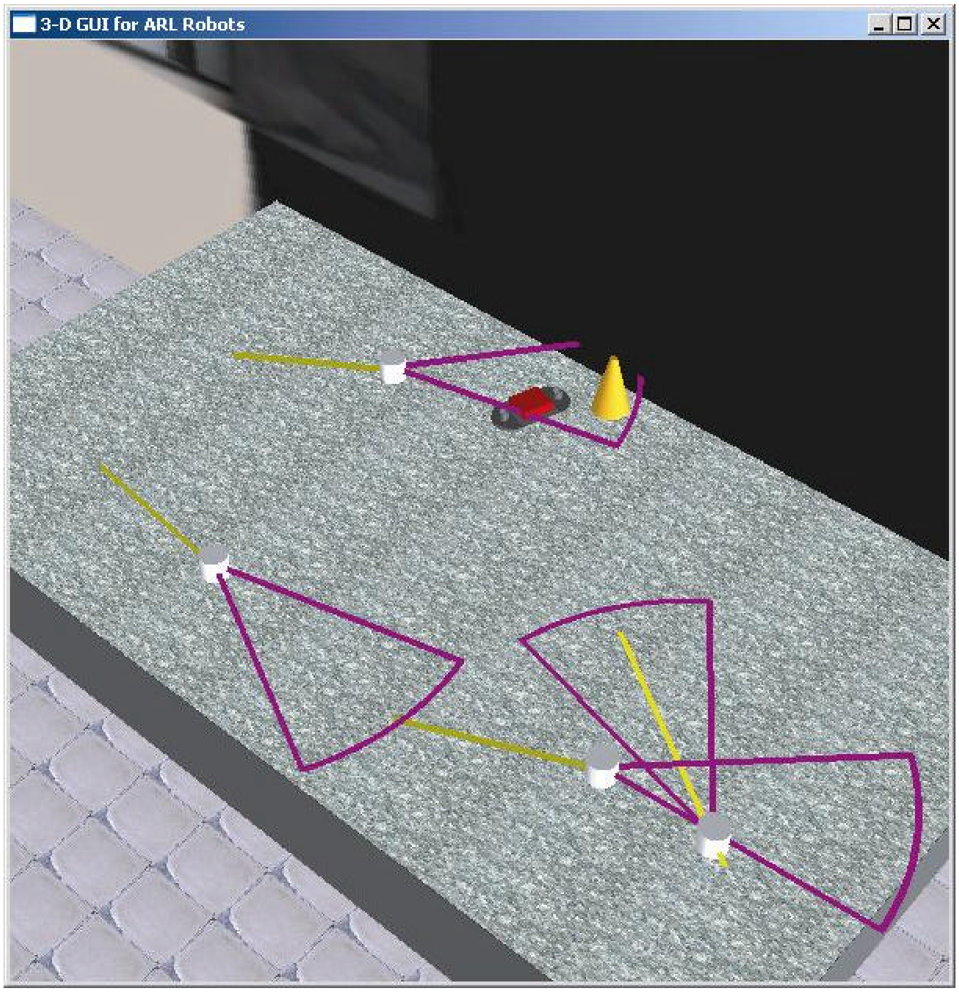

Figure 7.8: **Three-Dimensional Graphical User Interface Screenshot**

This screenshot shows four active robots in search of the cone and one of the floater objects. The sensor ranges of the robots are delineated by the pie-slice shapes. The dim straight lines are the current trajectories of each robot.

Architecture

The GUI was based primarily on OpenGL graphics, with the use of user interface-oriented software libraries and a tree representation of the object structure.

OpenGL

OpenGL (http://www.OpenGL.org) is the most widely adopted environment for developing portable, interactive 2D and 3D graphics applications. Since its introduction in

1992, OpenGL has become the computer industry's most widely used and supported 2D and 3D graphics application programming interface (API), bringing thousands of applications to a wide variety of computer platforms. OpenGL allows application developers to implement a broad set of rendering, texture mapping, special effects, and other powerful visualization functions. OpenGL works well on many desktop, laptop, and workstation platforms either through specialized hardware or software emulation. OpenGL is written in C and the function calls are in C.

GLT and GlutMaster

To use a C++ object-oriented methodology for the OpenGL GUI application, this implementation used the GLT and GlutMaster (http://www.nigels.com/glt/) libraries by Nigel Stewart (http://www.nigels.com). GLT is a C++ class library for programming interactive 3D graphics with OpenGL. GlutMaster is a C++ interface to the GLUT library. GLUT (OpenGL Utility Toolkit) is a C++ toolkit for writing OpenGL programs that implements a simple windowing API for OpenGL that is independent of the window system in use. GlutMaster consequently provides a portable window, keyboard, mouse and menu environment for OpenGL programs. GLT, GlutMaster, and GLUT are extremely portable. They depend only on OpenGL, and have been compiled and tested on Windows NT and Linux.

GLUI

In addition to the more mainstream OpenGL user interaction issues such as mouse handling, this research also required popup dialog boxes with buttons and text. For this purpose, the GLUI (http://www.cs.unc.edu/~rademach/glui/) software package, a C++ user interface library written by Paul Rademacher (http://www.cs.unc.edu/~rademach), was used.

Object tree

Because a C++ architecture had been enabled through the use of GLT and GLUT, the use of C++ class inheritance feature made data management and object drawing straightforward to implement. A class named CGUIEntity( ) was created as a union of two subclasses, CEntityInfo( ) and CShape( ). This class was modeled after the Composite design pattern [130]. CEntityInfo( ) is ultimately responsible for data collection and dissemination, while the CShape( ) class draws the entity. CShape( ) also contains a list of children objects.

This child list allowed the drawing application to make one Draw( ) call to the root object and all branches of the object tree would be drawn recursively. In addition, the application could send data to the tree root for processing and it would be quickly handled recursively. Consequently, no master lists of objects were necessary.

Correspondence groups

To handle the correspondence information generated by the Correspondence Agent, the GUI used the same parent classes and overall group structure used by the CA. Each group consisted of a list of pointers to its members, instantiated in the object tree described above.

Display of Room 010, Durand Building, Stanford University

Although OpenGL could generate any sort of scenario or setting, the environment modeled was that of the actual robot environment – Room 010 of the Durand Building at Stanford University. Other settings, such as oceanic search and rescues or space station servicing, also could have been modeled and used in simulated experiments.

Dimensions and Scaling

One of the major benefits of using OpenGL is that there are no limits on the scale of the modeled environments. The interface application was able to use the actual dimensions of the room and the structures and entities in it. Consequently, viewpoints, locations, relative distances, and other measurements would all yield results that reflected their real-life quantities exactly. The meter was the unit of measurement of choice because it was used in the coordinates generated by the overhead vision system.

Structures

The size of the room was measured so that the floor and walls would appear to be consistent with reality. The relationship between the structures in the room, such as the height of the large granite table from the floor, were measured and used. The north wall of the room was not implemented since it restricted the view of the room from the standard isometric viewpoint.

Textures

To provide additional realism, pictures of the walls of Room 010 were taken to create large mosaics. The GltTexture( ) member functions inherited through CShape( ) were used to show these pictures at the appropriate places in the room. Small image files found on the Internet created the appropriate look for the floor and granite table surfaces.

Robots

Drawing the MAR robots was straightforward. Each body was drawn using a white cylinder via gluCylinder( ) with a gluDisk( ) disk on top. After some use, it became apparent that heading information would be helpful. A small purple line was drawn in the direction of heading. In some missions, an estimate of the sensor range was also helpful. For these cases, a simple purple pie slice-shaped polygon was drawn from the middle of the cylinder as shown in Figure 7.8.

Objects

Robots could sense and interact with four different types of objects: an orange cone, a small-mass oblong floater, a large-mass circular floater, and other MAR rovers.

- Cone: GluCylinder( ) with a top radius of 0.01 yielded a cone with the proper dimensions, shown in Figure 7.9.
- Floaters: The “red object” and “grey object” of the “floater” type, shown in Figure 7.11, used a composite of eight separate objects. The base plate consisted of a flat box in the center with a cylinders and disk cover on each end. The color of the center power box identified the object. Two cylinders on each end simulated the manipulation sockets.
- Discus: The “discus” shown in Figure 7.10 was made up of five separate objects: a cylinder and disk for the base, a center power box, and cylinders for the manipulation sockets.
- Other MARs: Because some distinction was necessary between MARs that the operator could command and which ones the active MARs merely sensed, the method of drawing the sensed MARs needed to be slightly different. The removal of the purple heading indicator or sensor limit estimate proved to work well.

Figure 7.9: **Cone**

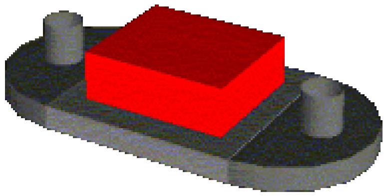

Figure 7.10: **Floater**

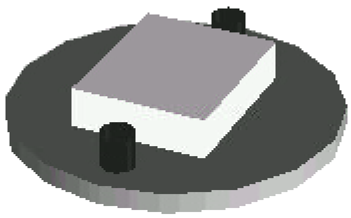

Figure 7.11: **Discus**

Moving the viewpoint

In some cases, the operator would want to change the viewpoint of the three-dimensional workspace. A common method of calling for this used in other three-dimensional workspaces is to use the mouse while holding down the right mouse button. When the right button was pressed, the mouse movements were captured and sent to a viewpoint-handling routine. The intent of the mouse-motion mapping was that if a point in the scene were clicked on, mouse movements would move that point around. For instance, if the mouse were moved up, the scene would rotate away from the operator's perspective with the axis of rotation in the tabletop plane. Mouse movements to the left caused the scene to rotate to the left with the axis of rotation perpendicular to the tabletop plane. The viewpoint stayed at the same distance from the center of the top of the table such that the viewpoint moved about a sphere in space.

To zoom in and out, the operator could hold both left and right buttons down and move the mouse up or down. Up caused the viewpoint to zoom in, and down caused it to move out. This action would effectively change the size of the sphere on which the viewpoint moved.

When the viewpoint was moved to its maximum height on the sphere (directly above the center of the table), rotation of the viewpoint along an axis in the tabletop plane stopped. If this rule had not been implemented, the viewpoint would start back down the other side of the sphere, making the entire environment appear upside down.

The viewpoint always focused on the center of the surface of the table; the (0,0,0) spot in the environment coordinate frame. This constraint was chosen because it greatly simplified instruction and proved to be sufficient for control. Many other methods for navigating a viewpoint in three-dimensional space exist [131] and some have begun to establish themselves as the standard through video games such as Homeworld. (http://homeworld.sierra.com). Users experienced in navigating three-dimensional worlds had no problems using the viewpoint movement methodology.

Commanding the robots

The command method desired for the robots was an electronic dialogue that determined the abilities of each robot with respect to the physical objects it sensed. This dialogue manifested itself through a selection of robots and objects on the GUI screen using the mouse and a selection of tasks from a popup dialog box.

Selecting robots

If the state of the GUI is such that no dialogue is being conducted (i.e., no robots have been selected), then a dialogue is initiated when the operator selects one robot by clicking directly on it or selects many by using a drag box. The click or drag box is translated using modified OpenGL "picking" calls into a list of entities 'underneath' that click location on within that drag box in order of their proximity to the viewpoint. These entities are processed in order to determine if the object is an active robot which the operator has the authority to command. If so, then that entity is told to draw itself orange starting with the next redraw and the state of the GUI is changed to reflect that a robot or robots are selected. A pointer to that robot or list of robots is stored for future use.

If the state of the GUI is not ready for a robot to be selected, such as when the task dialog box is being shown, or if there are no robots at the location of the click, a low beep sounds and no robot is selected.

Selecting objects

After the operator selects a robot or robots, she may then select a single object on which to perform a task. That object may be the robot itself, one of the objects or other robots on the table, or the tabletop. As with the robot selection, the click is translated into a list of the objects underneath the click. The pointer to the selected robot is retrieved and each object class instance on the list is sent to the robot class instance to see if that object is sensed by the robot. Only objects sensed by the selected robot may be chosen for manipulation.

If a selected object is sensed by the robot, it is told to turn itself green and the state of the GUI is changed to reflect that a target for manipulation has been selected. The GUI then takes the name of the selected robot and target object and constructs a query for the robot brain. This query is sent to the OAA facilitator and is routed to the proper Robot Agent.

Selecting tasks

The provision of task choices to the operator in most field robot interfaces is done by hard coding in the abilities of the robots into buttons or some other control mechanism on the screen, and then if the ability isn't applicable either gray out the button or (more likely) to just do nothing when that button is selected. These practices are not acceptable from the standpoint of modern human-computer interaction design. Buttons that may or may not be useful at a given point in time should not be taking up screen real estate and distract the user from the task at hand. An alternative possibility created to address such a situation is the pop-up dialog box.

When the Robot Agent receives a request to find out what tasks are possible, it forms the proper answer using its first-order theorem prover. The RA then sends the list of tasks to the OAA facilitator who routes the answer back to the agent making the request – in this case, the GUI.

The GUI receives the list of capabilities and processes them for display in the dialog box. Since multiple robots might have been selected, lists from multiple Robot Agents may come in. Once the GUI is sure that the entire list has been processed, a flag is changed that tells the GLUI library to show a dialog box using the task capability list. The dialog box consists of four parts, as shown in Figure 7.12: a title bar, descriptive text, the task buttons, and a "Close - No Task" button. The descriptive text includes the name of the object so that the user can double-check that the correct object was chosen. The task buttons have text that shows the name of the robot and the name of the task in a subject-verb command. When a single robot is selected as in the example shown, this information is somewhat redundant. However, if many robots had been selected, this information provides the proper affordances to the user. The "Close - No Task" button is available in case the operator changes her mind.

If the operator selects a task, the GUI takes the names of the robot, task, and object and creates a command to send to the robot. This command is processed by the Nervous System application and is executed by the robot.

Interacting with agents

As described in Section 7.1.6, the Correspondence Agent determined which reports of objects from the robots actually corresponded to the same physical entity. Sometimes, the CA did not decide that two objects were the same although the operator could readily tell that they were. In such an instance, the GUI was programmed to take instruction from the operator and publish a correspondence assertion. The operator asserted this by holding down the CTRL key and then clicking on two objects in sequence. There were times when the operator would want to know which objects had been grouped together by the CA. In such a case, the operator could hit the 'C' key on the keyboard to toggle whether the GUI hid all but the first item in the correspondence groups. The GUI sorted the groups so that if one item was its own source (i.e. a robot reporting its position), then that object was the one which would be displayed.

The number of objects, robots, and possible tasks could reach a high number when all the robots in the environment were active. This provided the user, conceptually, with a large

Figure 7.12: **Dialog Box**

number of choices of possible tasks. One way that the operator could find out the complete list of tasks possible on an object was to select all the robots and then click on the object (which would have been grouped by the Correspondence Agent as a single object sensed by all the robots). The GUI would then conduct a dialogue with each the robots. To assist the operator, the Query Agent (QA) was developed as described in Section 7.1.6. To access the Query Agent information, the operator could hold down the ALT key and then click on an object in the GUI. The results would then be displayed in a single dialog box, just as if the operator had asked the robots explicitly. There could be other users of this information, such as an advanced mission planner that needed to know which capabilities the robots had at a given time. Of course, there may be inconsistencies when using the QA between what is actually possible and what the QA reports, since the Query Agent updates the list from time to time rather than constantly. When the operator needs the most recent information, she should ask the robots directly and not use the Query Agent.

7.1.8 Operators

Many different types of users were able to operate the GUI easily and with very little instruction: university professors, undergraduates and graduate university students (in robotics and in other fields), high school and junior high students and teachers, and a SWAT Tactical Commander. All of them were very satisfied with the experience and were able to readily accomplish the tasks required of them.

In addition, there are conceivably millions of other people with the proper training to use this type of interface due to the ubiquity of direct manipulation interfaces and of real-time strategy games in particular. Such widespread familiarity with this method of field robot operation suggests that it is a strong candidate to become a standard for the field.

7.2 Observations of system use

Twenty-four operators used the experimental hardware and software over one hundred times to achieve various goals for the robots. This section discusses these operations and the lessons learned from them. These observations and preliminary analysis presented in this section form the first two steps of the next iteration through the four-step design process.

7.2.1 Typical system use

In a typical experiment, the operator would be assigned control of all five robots and verbally given a mission to execute. The complexity of the mission depended on the experience of the operator with direct manipulation interfaces, particularly those utilizing three-dimensional worlds.

The goal of these experiments were to ascertain the strengths and weaknesses of the system through use by a wide variety of operators and to exercise some of the novel capabilities of the experimental system:

- Multiple objects and tasks
- Heterogeneous robots
- Limited local sensing
- External information integration by the operator
- No global data model
- Dynamic task models

Simple mission

For first-time operators or novice computer users, a simple mission was assigned. The most common simple mission goals were to use some robots to follow moving objects and to move the other robots out of the way or to open places on the table. The purpose of these missions was mainly to allow operators to explore the capabilities of the robots and the user interface.

Complex mission

Experienced users were given a more complex mission. One of the most common mission goals was to have one robot watch another as it bumped into the cone. The experiments were set up such that the limited sensor ranges of the robots did not detect the cone location initially. In addition, the robot states were altered so that they had heterogeneous capabilities such that some robots could only bump into the cone, others could only watch other robots, and others could do both. The task was further complicated when the use of the surrogate relative sensing system introduced position and orientation errors, highlighting the usefulness of the Correspondence Agent. Finally, some aspect of the mission required the integration of external information beyond what was available at mission start. For example, the operator was told that an observed and moving object ahd been determined to have a tendency to move toward the unobserved goal object (i.e. the cone). By recognizing that a Follow task on that moving object would lead her to the cone more quickly than a formal search pattern, the operator could accomplish the mission more quickly.

Remote operation

In a most cases, the operator used the GUI in the same room as the robots. The robot workspace was visible from the operator workstation, but not from a perspective that was useful for operation. However, sounds from the robots could be heard which could provide information that would not be available in an actual field robot deployment. Consequently, for five experiments the operator was put in another room and required to conduct a complex mission. There did not seem to be a marked difference in performance for new users who did not know how to incorporate the noises from the robots into useful feedback. Experienced operators, particularly those who had worked with the development of the robots, definitely noticed the lack of the noise cue for determining whether the robots were operating as expected.

Multiple operators

To highlight the capability of the interaction to accommodate multiple operators, a simplified game of “Capture the Flag” was played by two operators sitting at separate computers and each commanding two robots. The operators were told they were on opposing teams and needed to capture the flag of the other person by touching it with a robot. The "flag" was the cone for one team and the discus for the other team.

7.2.2 Successes

Utilization of the system was generally very successful. The two most significant observations were that users learned rapidly how to use most of the system functionality and they could thus quickly accomplish complex missions. Although the interface's knowledge of the robots' characteristics was very limited compared to other field robot interfaces, the interface accomplished its role by affording all of the potential robot functionality to the operator.

Rapid learning

Most operators were instructed in the use of the interaction in less than five minutes. Some users, typically younger people with extensive video game experience, had no trouble utilizing all of the interface's functionality after only a cursory explanation. These operators had the most success in accomplishing the "mission," although they were also the most vocal about the various improvements that could have been made to the user experience.

Complex mission success

The number of robots, objects, and tasks and the variety of the mission goals and constraints was higher than most field robot systems. Consequently, the missions accomplished by the operators were the most sophisticated that a real multiple robot system is known to have completed successfully.

Ease of use

Even operators who did not have much experience with direct manipulation interfaces were able to operate the robots. The general concepts were found to be intuitive and the fundamental actions to accomplish goals were easy to learn quickly.

External information integration

Integrating external information into the mission for the coordination of action, and to a lesser extent for the cultivation of common ground, was expected to be an important capability of the operator. The experiments showed that not only was this capability utilized through this object-based interaction, it was one of the most vital aspects of the system for successful mission completion. Even in this experimental setting, the conduct of real missions inevitably required the integration of information not perceptible to the robots directly.

Usefulness of agents

Without the Correspondence Agent, many more errors were made in attempting to accomplish cooperative tasks and the experience was very frustrating for the operator. Some operators found the Query Agent particularly useful and used it for all robot command formation, while others only used it for complex multiple-robot situations.

7.2.3 Limitations

The experiments also pointed out some limitations in the interaction, the interface, and the robots.

Compound tasks

Accomplishing compound tasks – that is, a sequence of simple tasks – proved to be difficult and consumed the operator's attention while she waited for the first simple task to be completed. A command queue would be relatively simple to create, as they are very common structures in RTS games and should fit within the overall interaction framework. However, implementing a queue on this actual robot system would have been contrary to one of the central assumptions, which is that the capabilities of the robots change dynamically. Compound tasks are difficult to carry out if a second or subsequent task is not possible once the robot gets to the point where that task needs to be executed. There might be a way to elegantly integrate the dialogue concept by initiating a follow-up dialogue at the proper time. Another option might be to develop a real-time complex task planner that considers the changing task models in its plans and serves as the point of contact for the operator interface instead of the robots themselves. This problem is a prime area for future research.

Robot identification

The individual robots were not explicitly identified by the display during their operation unless a dialogue was conducted. This could have been addressed with some changes to the interface, but exactly how they would have been identified is not clear. Colors might have been used, although then the operator would have needed to remember which color was for which robot. Numbers might have been used but would have either needed to float over the robot or somehow be on the robot in a direction always visible to the operator. The operators tended to find out robot identities by double-clicking on the robots to get the list of self-tasks and then choosing "Close - No Task."

Failure notification

One regular complaint was that there was no explicit notification that the state of the robot had changed and that consequently the capabilities of the robot had changed. This was only observable by the operator if she chose a certain object for a robot and the returned list of possible tasks was different from what was expected. There are many different ways to implement such a notification procedure, but none of them were implemented because the best notification method is likely to be mission-dependent and an attempt at a generic solution might have detracted from the primary goals of this research.

There was no direct way to determine robot state other than to see the effect of the state on the robot capabilities. This was an explicit usability-oriented design decision – the important information ultimately for the operator should not be that the robot has a particular state but what the current mission-relevant capabilities are. As an analogy, it is not as important to know how much fuel is in a car but rather how many miles it can go. A fuel gauge is just a convenient way to provide information that the operator can then derive into the desired data. Nonetheless, a more experienced field robot operator might be able to look at the raw robot state and determine whether or not to ask a robot for something immediately without being forced to wait for the dialogue to take place. Perhaps a compromise between the usability design goal and the expert operator expectations would be to develop an additional capability for the Query Agent that determines all the possible capabilities of a robot and displays them when the robot is selected, as compared to its current method of organizing capabilities only by which objects can receive them.

Chapter Eight

Applications and Future Work

This is the concluding chapter of this dissertation. In the first of the four sections, some possible applications for this research, both within field robotics and beyond, are discussed. The number of robotic systems with multiple robots will likely increase dramatically over the next few years, and the role of the human within these systems will evolve. Since the concepts developed through this research were based on a fundamental analysis of human roles in distributed team command, the object-based interaction should be able to adapt to this evolution. The second section in this chapter discusses areas of future research that would either be useful to expand the functional capabilities of the current system or to extend this work into related fields. In the related areas of distributed work and human interaction with automated systems, the basic findings of this research add to the existing bodies of knowledge. In the final two sections, the contributions stated in Chapter One are reiterated and a brief conclusion is given.

8.1 Applications

There is a wide range of applications for this object-based interaction. The choice to implement the research using the Micro Autonomous Robot testbed was made mainly to simplify development without a significant sacrifice in realism. The basic principles of MAR operation were consistent with the operation of other types of robots, and the infrastructure developed is applicable to any robot system. Nevertheless, some applications are better suited for an object-based interaction than others, since the user of objects requires some computational overhead that is not always necessary. This section discusses the system qualities that promote the use of this interaction framework and those that discourage it.

System qualities that highlight usefulness

Ability to focus on high-level robot requirements

Most robot operators would prefer to focus on tasking the robots and using the current robot capabilities to solve problems and accomplish the mission. These roles allow the unique creative aspects of the human component of the system to be engaged. The operator's responsibility for all for the robots at once also means that the low-level control and troubleshooting issues for each individual robot are handled by some other aspect of the system. There are at least two ways to accomplish this – robust robots or specialized troubleshooting operators.

Robots that are able to take care of themselves are the best solution to the problem of managing low-level issues. They should handle most problems that come up and accordingly change their responses to the operator about their current capabilities. When in trouble, they might provide a list of task choices to start various self-monitored troubleshooting methods. Methods of increasing robustness of field robots vary from contingency management systems [132] to autonomous self-monitoring and diagnosis [133].

Providing robust robots will be a significant research and development task, and the current solution to handling the common problems that plague robots is to let humans perform the troubleshooting. This expectation is not reasonable when the operator is responsible for many robots, but if she can pass off the troublesome robots to another tier of operators who are solely responsible for that role, then she can continue to work with the functional robots. With a well-designed management system, the object-based interaction could be used at a high level with any size robot population, supported by a calculated number of troubleshooters based on a predicted failure rate.

Use of objects

This interaction is especially useful in environments where the utilization of physical objects as points of reference is natural, straightforward, and necessary. In the SWAT environment, physical objects were commonly used as references between team members and between the team and the commander. Many robotic missions are for exploration or assembly, and physical objects play a natural role in such situations. When physical objects can be readily identified, particularly in situations like construction tasks were all objects can be explicitly labeled, the implementation of this interaction becomes straightforward. Even when the objects are not labeled, the interaction should work well if the robots can identify objects. For missions such as a Mars exploration, where the large number of rocks might lead to confusion, the model used by humans in similar situations should prove useful. Rocks or other objects with significant distinctive attributes could be given a name and that knowledge could be passed to the robots, creating surrogate object identification. If manipulation of objects is required, this interaction supports a broad spectrum of possible object-based tasks.

Multiple robot coordination necessary

This interaction was shown to be useful for coordinating many robots at once, particularly for tasks that benefit from centralized coordination. Other interactions that do not provide global perspectives of the environment or mechanisms for multiple robot tasking do not appear to support robot coordination as well. Use of scripts, one current alternative, might result in faster execution of some commands but are difficult to use when mission goals and resources change rapidly.

Integration of external information

By keeping the operator actively involved, a direct route exists for external information to be integrated into the planning and the execution of the mission. If the entire environment, current and future, could be completely described beforehand, a computerized planner might be able to provide a more optimal plan than one generated by an operator. Such planners do not typically have a mechanism for incorporating new types of external information during their execution.

System qualities that reduce need to use

There are costs incurred when implementing a generic interaction of this nature. Some of its components are relatively complicated so that their capabilities are flexible enough to handle a wide range of prospective robots and missions. Some systems may have qualities that reduce the need to implement some aspects of this interaction.

Comprehensive global sensor

If a comprehensive global sensor is available, then a large part of this interaction's infrastructure is not necessary. For example, the robots could get accurate and consistent information about their location and the location of objects from the global sensor, immediately making the Correspondence Agent obsolete. The architectures of the robot nervous system program and the 3-D GUI interface application were also based on the assumption that each robot produced its own information. Although the current interaction could adapt to this change in information source without any modification, a significant amount of programming and processing overhead could be eliminated.

One of the most common comprehensive global sensors is the operator herself when she is collocated with the robots. In such a scenario, some of this interaction's infrastructure is particularly redundant and definitely should be eliminated to avoid possible confusion between the operator's perception and the interaction's output. One should note that a robot system collocated with the operator ceases to fulfill the definition of a field robot, and other human-robot interactions have been developed for such systems.

Operator must periodically shift to teleoperation level

If the operator periodically has to shift her attention to lower-level aspects of the robotic system, then this interaction will need to be augmented at best and replaced at worst. The interaction does not provide a natural way to move down to a low level for direct control of a robot. This was an explicit design decision based on the conceptual viewpoint that this would not be an attractive way to conduct operations since the other robots would be left without an operator. One possible counter-argument is that the operator may not be busy dealing with the robots at a high level and would have time to move to the lower level. If this were the case, a better solution would be to add robots to her responsibility until the operator is sufficiently engaged. If there were no more robots in the system, the operator should be given other tasks consistent with the high-level planning role that could be integrated easily with the high-level interaction. Low-level control or troubleshooting tasks could be passed on to other operators with human-robot interactions specifically designed for those roles.

Single simple robots

If there is only one single robot, then much of the interaction's infrastructure would go to waste. In addition, if there is no manipulation of objects required (even in a broad sense of 'manipulation' where any action on an object such as 'watch' is considered a manipulation), then the infrastructure necessary to support this interaction could be considered excessive. Nonetheless, the system still functions very well for one robot, and much of the initial system testing and debugging was conducted using just one robot.

Example applications

Given the preceding breakdown of traits that encourage or discourage use of this interaction, this section lists a few real-world example applications that appear to be good fits for this object-based human-robot interaction framework.

Space solar power construction

Some of the proposed space solar power construction missions are expected to use 1000 robots operated by just 100 humans. Besides the obvious need for an interaction that would allow the operation of many robots by a single operator, many assemblies and transports would need to be performed cooperatively, and undoubtedly there would be new external information throughout mission execution that would affect the overall plan of action.

Search and rescue

Future search and rescue efforts will use many autonomous systems to provide around-the-clock search capability. Such efforts require the integration of a great deal of external information, which means that robotic systems would benefit from having a human operator involved in their planning and execution at a high level.

SEAD

The suppression of enemy air defenses (SEAD) is one of the most dangerous missions for military pilots today. These missions have recently been conducted by cruise missiles and stealth airplanes, and plans call for unmanned combat air vehicles to be used. These missions would be a natural fit for this interaction, either using a GUI interface at a computer terminal or using a voice-only interface in conjunction with pilots in the air.

8.2 Future work

Human interaction with robotic systems is a rapidly growing field, and the main research issues in this area are drawing increasing attention as field robot systems are deployed. This section describes the future work that should be conducted to continue the progress made by this research.

8.2.1 Improvements

Just as this research described the conduct of one loop through the design process, future efforts should continue to iterate to refine the human-robot interaction. Based on informal observations of the results of the implementation of this experimental system, prospective areas for innovation and improvements in the next version could be identified.

Robot improvements

The robots of the MAR test bed are not currently able to conduct cooperative tasks. There are other robots in the Aerospace Robotics Laboratory that can conduct such tasks, and either these should be integrated into the interaction or the MAR robots should be programmed to carry out some cooperative tasks.

The robots also did not actually monitor their own battery condition or camera status, so this capability had to be simulated manually. Consequently, there were times when the batteries were actually getting low or the camera was malfunctioning and the benefit of self-monitoring and reporting was clear.

Graphic User Interface (GUI) improvements

Users experienced with Real-Time Strategy video game interfaces had quite a few recommendations to make the GUI better. A small icon over each robot with some form of identification and robot battery status was often recommended. In addition, many games provide a ‘hotkey’ mechanism that allows users to bind certain robots or groups of robots to a key on the keyboard, allowing the user to quickly select robots. Finally, some users suggested that the robots send the task lists along with each object state packet so that the dialogue procedure would not be necessary. Four factors (communications bandwidth, time delay, list size, and time pressure) determine whether the task lists should be sent, as shown in Figure 8.1. The experimental setup, because it exists in an environment with no time delay and high communication bandwidth, exists in the category where the task lists could have been sent, but this was not done to highlight the utility and the cost of carrying out the dialogue through explicit turn-taking.

Infrastructure improvements

One of the most useful infrastructure improvements would be a more intelligent Correspondence Agent. The CA that was built for the experiment utilized simple rules that were sometimes fooled by the conditions of the system. A CA that took into account the velocities of the objects would likely perform better than the position-only rules used by the CA.

At times, the messages from the GUI to the robots did not seem to reach their destination. The NDDS communication infrastructure permits reliable publications and subscriptions, which might have solved this problem, but this capability was not used in this implementation. The OAA library, while useful in the early stages of the project as a method for agents to communicate and as a pre-built wrapper for the first-order theorem prover, created a large number of memory leaks within the applications and required extra operating overhead that might have been as readily addressed by creative use of NDDS.

Finally, the modular application approach worked well during development of the robot component of the experimental system, with separate applications for the servo controller, the trajectory generator, the task manager, and the capabilities prover. This created some overhead in managing the startup, initialization, and interprocess communication that would be reduced if all were combined into one single robot-control program.

8.2.2 Evaluation

One of the main questions answered by the successful operation of this experimental system is "Will a human-robot interaction framework based on observations of a human-human work setting be effective?" Qualitatively, this question was answered affirmatively, but quantitative results are needed to strengthen this conclusion.

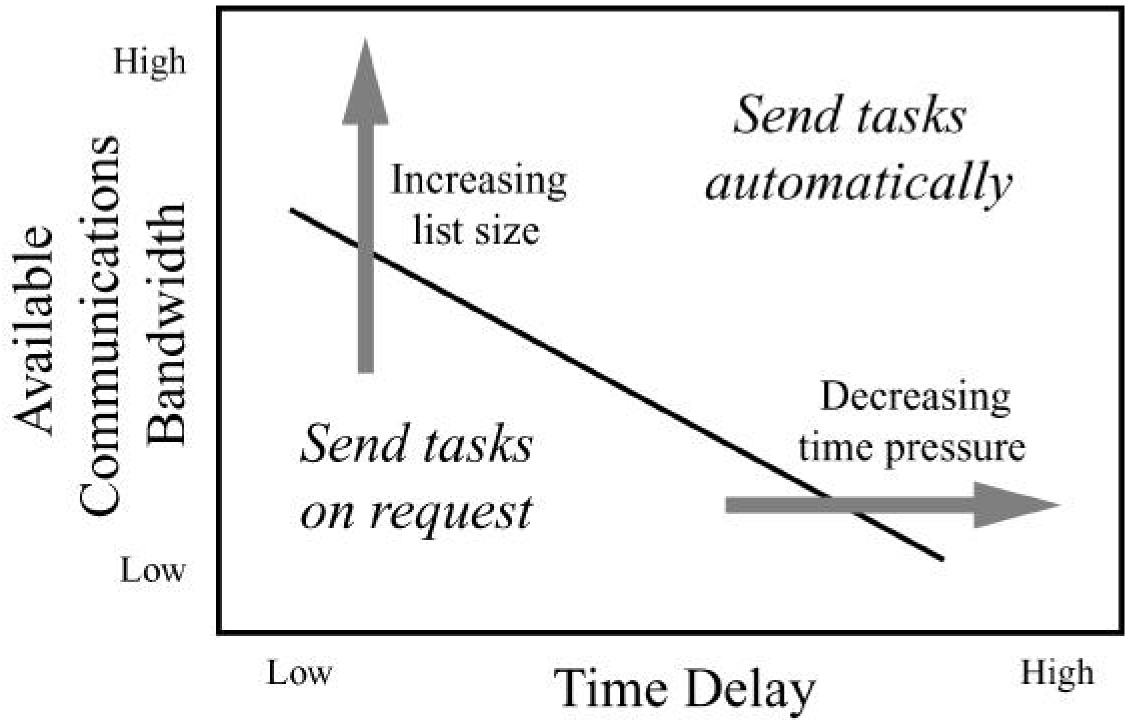

Figure 8.1: **Sending the Task List**

Determination of whether the task list should be sent automatically or upon request is a function of a few factors. The line that separates the two choices is moved up when the list size increases because communications bandwidth must be larger. The line moves to the right when there is more time to make decisions based on the time list, therefore reducing the time pressure and the negative effect of time delay.

Scalability

Observations of a dynamic direct manipulation interface such as those used for RTS games show that operators are comfortable with over one hundred entities under their command. Given robots with a considerable level of robustness as discussed in Section 8.1, the robot operators should be able to operate robots on a similar scale. Tests of this scale should be conducted with simulated robot systems and preferably extended to actual robots. In addition, workload optimization plans may further increase the scalability of the system [134]. The addition of robots, objects, or tasks to the experimental setup should have some effect on system performance, and their relative effect would be interesting to analyze.

Performance comparisons

Other methods for operating multiple robots were identified in Chapter Two. These systems did not utilize physical objects but rather explicitly commanded robot movement. A fundamental comparison between these other operation methods and the object-based method would be informative. However, the differences between the interactions would probably mean that the nature of the mission and robots used would have a great effect on the experiment's outcome and consequently confound the comparison. This is a common but difficult problem in many fields and their approaches should be investigated for possible application here. One possible comparison technique, used in a US Army experiment, would be to compare results of humans-only and human-robot teams, rather than between robot systems, since that comparison is more useful for officials assessing the utility of robots [135]. The expectation might be that the humans would perform much better, but that may not be true for a set of tasks and environments not comfortable for humans but accessible by robots.

Usability

Future research is needed in the form of usability studies to assess the impact of this design on usability and determine the extent to which it is an improvement over existing single- or multiple-robot system interactions. Such studies would be a substantial contribution because there are currently few usability studies performed with field robots. Even basic user interaction analyses such as GOMS [136] have never been done in a field robotics context.

Not only would the pursuit of a more complete usability understanding for field robots do more than generate a measure of whether a system was built well once it was complete, but it would help with decisions throughout the design process. Good usability investigations apply pressure to the assumptions that form the foundation of the design, such as who will be the ultimate users and the extent of their regular and exceptional uses. The usability queries will often uncover issues with the use of the system that may lead to further design questions, even about internal processes of the system not directly accessed by the user.

With very limited previous research documenting the usability of field robot systems [66], much work remains to build even basic knowledge to serve as a foundation for design studies. Without usability information for current implementations, comparisons with future systems are impossible. Basic investigations of existing human-robot interactions (even those interactions that are not considered to be usable) would be useful to characterize the positives and negatives that could be addressed in future designs. Such studies might also be able to determine the 'costs' in operator training, operational overhead, and errors to justify further usability-oriented investigation.

Even with the challenges caused by the lack of previous usability work with field robots, the methods of determining useful usability metrics are numerous. The choice of experimental setup depends only on the information being sought. Some possible usability comparisons might be:

- New interface vs. old interface. Even if the old interface is obviously inferior, how much of an improvement is the new one?
- Two different new designs. Requires more work to develop two separate designs, although this could be used to study small differences within a new design.
- Team of human operators vs. one human operator. The choice of mission may have a large effect on the outcome of such an experiment.
- Cross-mission and cross-platform effects. For a given interface, usability will likely vary depending on the type and capabilities of the robot as well as the mission setting.

To measure usability, a number of measures are possible. Besides basic analyses such as GOMS [136], one could study the amount of mental workload, the number of errors, the degree of task completeness, the reported comfort level or other subjective feedback, or the extent to which the interface use matched the designer's intent. Within these measures, conditions could be altered by varying the required task for each operator or varying operator skill to accomplish the same task. The cognitive load could be conditioned by requiring that the operator perform a high- or low-demand complementary task or through modulation of the required speed of the task. The tasks themselves could possibly have embedded measures to determine stress or task performance errors. Ideally, such studies would not only create a more usable interaction for a particular robotic system, but contribute to the scant existing theory as well.

8.2.3 Studies of other distributed groups

For reasons discussed in Chapter Three, this research studied SWAT teams as an analogous setting for human-robot teams. SWAT teams, as extreme distributed work teams, are unlike most other teams in several ways – their work is dangerous, they are extremely and unavoidably interdependent, and they are temporary. Although these factors make SWAT teams ideal for understanding how one might coordinate field robots and how teams are structured in extreme settings, they also may limit the generalizeability of these results for more traditional teams.

Many questions were generated by this research that would need to be answered to determine the degree to which extreme distributed teams are different from other teams. For example,

- Are objects used in other teams? Because objects encapsulate so much information and are natural tools for humans to use in their communication, one would expect that all teams use them (though no studies were found to confirm this). However, are objects used in the same way in 'normal' distributed teams, since the team members do not share a physical space? How are objects used in collocated teams?
- How do the roles assumed by the extreme distributed work team leader change on other teams? Does the importance of the roles diminish? Do the responsibilities migrate among the team members, or remain with one or two people throughout the team's existence? Are there available technologies that support these roles in normal distributed teams?
- What technology could be used to support extreme distributed teams? Jones et al [70] found that the SWAT team applied a strict standard that rejected any new technology that might increase the amount of uncertainty in its environment. The technology must be robust to failure, easy to use, and perform some task that significantly offset whatever overhead costs the technology use incurred.
- How many subtypes of extreme distributed work teams are there? In what ways do they vary? Are the results of this dissertation, particularly the roles played by the commander and the utilization of physical objects, true for other teams?

- Is the commander the sole source of information sharing among the distributed team members? What were the exact techniques used by the commander for information sharing, including the frequency of exchanges, the cues to request or provide information, and the extent of the shared information. In what instances do the distributed agents communicate directly?

In Section 3.3.4, alternative ethnography settings were suggested, such as football teams, orchestras, and military commanders. The lessons learned through the study of the leaders in these environments would be expected to provide different and interesting results. The methods of orchestra conductors, for instance, might inform better field robot design when the robot operator is responsible for monitoring the conduct of many heterogeneous robots simultaneously with minimal input on their conduct.

8.3 Contributions

The primary contribution of this research is an object-based human-robot interaction that enables the command of multiple robots by a single operator. The secondary contribution of this research is the utilization and validation of a hybrid setting approach to user-centered design.

This section reiterates the contributions stated in Section 1.3 and expands on the discussion in light of the information presented in the intervening chapters.

8.3.1 Created a new object-based human-robot interaction that allows a single operator to operate multiple complex field robots

By basing an interaction on the objects sensed by the robots and the tasks afforded by those objects, the operator is able to coordinate the action of multiple robots simultaneously. Previous human-robot interactions for multiple-robot systems (Section 2.2.4) were designed to give the operator control over the positions of the robots only. Although the position-based method is sufficient for some applications, it does not readily integrate new robot autonomous capabilities and requires operator input for all robot activity. The use of objects and object-based tasks encapsulates more intelligent robot behaviors so that field robots are capable of working without constant human direction.

First user-centered development of a human-robot interaction to demonstrate the operation of multiple heterogeneous field robots by one person

Very few robot interaction frameworks have been developed with user-centered design approaches, and those that have been were not for multiple heterogeneous robots. As was discussed in Chapter Two, the function of the operator in the multiple-robot system has typically been addressed only at the end of the system design, yet the importance of integrating human capabilities – such as creative thinking and decision-making – in these systems is increasing. This tendency to push user interaction to the background is not only a concern in robotics; many other technological fields have similar histories during their early development [137]. Only two other field robotics research efforts have considered usability metrics during development: the testing of supervisory control methods for multiple simple homogeneous robots by Ali at Georgia Tech and the implementation of user-monitoring software for operating a single field robot in DARPA's Tactical Mobile Robotics program. Many more usability-oriented studies must be done in the course of subsequent robotics development, but in the short term this research has attempted to raise the profile of usability considerations during robot design.

This research approach was experimentally validated (Section 7.2) through demonstrations on a number of robots in a variety of situations, showing that it is a useful and flexible method for the operation of multiple field robots. Such validation is not common in field robotics research due to the costs associated with developing experimental platforms.

Finally, the robots in other multiple field robot systems were homogeneous in type and capability, which significantly reduces the complexity necessary for the operator interface and subsequently reduces the complexity presented to the operator. The robots in this research had heterogeneous capabilities, yet this heterogeneity was handled elegantly through the interface design.

Extension of an existing effective human-robot interaction (OBTLC) to a multiple-robot, multiple-task, multiple-object environment

This research made significant extensions to the Object-Based Task-Level Control (OBTLC) paradigm to allow operation of multiple complex, heterogeneous robots. This new functionality required innovations with both the robot and computer interface components of the system, as described in Chapters Five and Six. Prior to this research, almost all OBTLC implementations had been single-robot, single-object, and single task, with the remaining few either emphasizing the single-task collaboration of two robots with a single object or the ability of one robot to distinguish between multiple objects. While OBTLC had the promise of allowing an operator to control many robots by decreasing the workload for each robot, no experiments had shown that this would actually be possible. In addition, although previous OBTLC research has been carried out on many types of robots, no experiment had included more than one type of robot.

An object-based dialogue ("Do what where?") that transcends an existing object-based manipulation paradigm ("Put that there")

This research developed a dialogue interaction metaphor for field robots that is flexible, dynamic, and task-oriented, based on a new "Do what where?" dialogue paradigm (Section 6.4). The fundamental metaphor for the interaction between the operator and the field robot is that of a conversation between humans, particularly one where the parties are seeking to establish shared references to allow collaborative action. Although interactions based on dialogues are not new to some human-computer interactions, they had not been applied to the control of remote, complex, and physical (non-virtual) machines such as they have in this research.

"Put that there" is a well-known interaction paradigm developed by Cannon at Stanford University and further advanced by Cannon and others [27]. However, there are implicit limitations to a "Put that there" methodology. The first is that the verb "Put" implies that motion of something must take place, when in fact field robots have many potential tasks that do not explicitly require the motion of an object. In addition, there is no prologue to the "Put that there" dialogue for the discovery of what a particular robot is able to see or do before the operator commands are given. As a result, "Put that there" can be seen as a subset of the greater "Do what where?" paradigm; in fact, "Put that there" might often be the literal answer to the robot's query of "Do what where?"

Context-sensitive affordances of multiple task choices

Context-sensitive affordances, a familiar concept for human-computer interactions, were applied to a robot interaction for the first time. Existing interfaces for field robots typically use different modes of operation depending on robot state and operator intentions.

However, the difficulties of mode changes and their adverse effects on human operation have been well documented [67]. As field robots serve as physical agents to conduct missions remotely for their human operators, the robots should be as flexible as the operator's creativity and resourcefulness requires. In addition, one of the greatest challenges during the field robot development and operation process is when the actual robot capabilities and the interface affordances are no longer synchronized. In response to these issues, this research identified and implemented an interface design that adapts gracefully to the state of the robotic system via context-sensitive affordances.

The result was an interface that only affords the actions that are possible at that point in time, and are sensible for the combination of robot and object that have been selected by the operator. These context-sensitive affordances eliminate much of the confusion created by existing robot interfaces that afford all capabilities at all times or rely on various operation modes. By providing only the tasks possible for a given object, the full extent of the robot capabilities are passed to the operator in a simple, straightforward communication structure. Other items of interest to the operator, such as lists of all possible tasks, all sensed objects, or shared tasks or objects between robots, can be constructed from this atomic unit of interaction.

Integration of software agent paradigm to provide user assistance

Software agents were added to the OBTLC framework (Section 7.1.6) to assist the user with the more complex interaction requirements presented by a multiple-robot system. The ease of operation that the use of OBTLC enables does not come without cost somewhere else in the system. The primary manifestation of this cost is in a higher requirement for robot onboard intelligence and the dependence on certain types of communication with the operator. As more complex robotic environments with greater numbers of smarter robots are created, the operator needs help to manage the additional information.

Identified the implications of using physical objects as the basis for an interaction with multiple field robots

While the use of physical objects for references is natural for the operator and should produce a more effective human-robot interaction, this focus creates distinct challenges for implementation on real robots as discussed in Sections 5.4 and 6.3. For example, relative sensing of objects by multiple robots leads to certain difficulties in establishing centralized comprehensive world models. Each robot will have its own model of the world, which may disagree with all other robots' models. Therefore, while relative sensing is the only method available for field robots, all known multiple-robot interactions rely on some form of global sensing for determining robot and object location. As another example, a single global or comprehensive data source makes the interaction with the operator much more straightforward, and it is very helpful for coordination between robots. Nonetheless, such a source is not available in field robot deployments. Consequently, operator control and robot cooperation both require a facility to determine correspondence between objects sensed by the robots. This correspondence may be done manually by the operator or automatically as part of the overall robotic system.

8.3.2 Utilization and validation of a novel hybrid-setting approach to user-centered design

By utilizing a hybrid setting, the first interaction implementation could be informed about the roles and tools required by the operator even though no operators were available to study. Previous early-stage interaction development in robotics either was not concerned with the user or waited until the entire system could be built and tested. As a result, progress in the area of human-robot interaction for field robots has been slow, particularly when compared to advances in sensing or planning or to the current state of human-computer interaction. The hybrid setting approach, through its use of a non-robotics source for observation and analysis, promises to accelerate the understanding of the inclusion of the human in the robot system. Furthermore, this research validates the hybrid setting approach by applying the analysis from a non-robotics setting to the development of a human-robot interaction and testing its feasibility with an actual multiple-robot system.

Conducted field studies of leadership in extreme distributed work teams

Research in leadership of distributed teams of all types is still in its early stages, and this was the first research of its kind to study the leadership of 'extreme' distributed work teams. The teams studied were police Special Weapons and Tactics teams (Chapter Three), who are spatially distributed once they are deployed and are led by a Tactical Commander who is located remotely from the team members.

Recognized the importance of physical objects as tools in communication and coordination

The most important communication and coordination tool for the team were references to the physical objects in the environment. As was discussed in Chapter Four (Section 4.3), objects provide unique common points in the world that can be used to facilitate the reconciliation of various perspectives and the locations of the participants. In addition, objects may serve as focal points for cooperative action since they encapsulate much more information than their appearance and state, such as the actions that may be conducted using them.

Identified two vital roles of a team leader and observed how those roles are conducted

Recognizing and defining roles of team members is a key step in research into teamwork, and no information previously existed for the leaders of extreme distributed work teams (either humans-only or human-robot). The analysis of the SWAT team observation (Section 4.3) revealed two important roles of the leader:

- Cultivation of common ground: Common ground is an important feature of distributed teams, but the leader's role in maintaining it had been mostly unstudied. This research found that the distributed team leader was in a unique position, due to his remote location and ability to focus solely on team leadership, to cultivate common ground throughout the team. He acted as a filter so that each member was not overwhelmed with information but knew as much as possible relevant to that member's particular perspective. Prior to the incident, the leader prepared the team by establishing standard procedures and then calling for certain ones as the incident began.
- Coordination of action: Action must be coordinated in any team, but this is more difficult in distributed teams when communication between members is limited. This research discovered that not only was the responsibility for coordination within the team heightened for the distributed team leader, but he also served as the sole integrator of external information and constraints into team activity. The leader coordinated actions among team members by providing commands in the frame of reference of the addressee, enabling coordinated action between team members without the requirement of a shared mental model. With the leader integrating

external information, the individual team members could focus on their own tasks and not the larger context of the mission.

Advanced the state of understanding of the role of the operator in field robotics

The roles of the robot operator and the tools she should be given to operate are not well understood nor have they yet become the subject of much research. The design requirements due to these roles and tools will have a significant effect on the entire robotic system. Consequently, it is important to understand the fundamental functions of the operator early in the system design process.

Identified the atomic component of field robot interaction framework: the determination of valid tasks to perform on an object and the subsequent command of a valid task by the operator

No previous work has attempted to identify the fundamental communication requirement of a field robot interaction framework, given the bandwidth, sensing, and other constraints posed by a field robot system. Based on observations and analysis of an analogous setting, this research proposed (Section 4.4.2) that the atomic component of the human-robot interaction framework should be the discovery of valid tasks to perform on an object and the subsequent command of a valid task by the operator. “Atomic” in this sense means that it is the smallest unit of interaction, from which interactions that are more complex may be created but smaller interactions should not be attempted. This conclusion was reached based on the observations of the SWAT commander and consideration of the constraints of typical field robot deployments. The operator prefers, if possible, to achieve mission goals by providing commands at a task level rather than a direct motion-control level, and in field robot deployments, motion-level control is difficult if not impossible to conduct with the operator in the loop.

Highlighted areas for focus in robot autonomy research to enable better integration with an operator

The level of autonomy and the role of the operator are tightly coupled, yet data is scarce on which further autonomy developments are likely to enable for better operator performance. Three of the most useful such areas for autonomy research identified by this research are:

- Identification and manipulation of physical objects,
- Self-monitoring and intelligent notification when problems occur, and

- Independently organized cooperation and collaboration on high-level tasks.

The management of physical objects by robots must be improved to enable the basic requirements of this research. Recent work in computer vision has made great strides in recognizing various types of objects, even in cluttered backgrounds [138]. This recognition can also be augmented by a human operator dedicated to that task [106]. Once the objects are perceived, the robots must maintain the persistence of each object over time [139]. Most elements of the object management have been addressed, but integration of the best systems to create a capability approaching that of humans will require much more work.

By robustly self-monitoring themselves and acting appropriately when failures occur, robots allow the operator to spend less time as a monitor (a menial task) and more as creative planner (a meaningful role). A few different strategies for this issue exist, from contingency management systems to reactive behaviors [13]. The best choice is usually application-dependent, determined by the complexity of the robot and the level of structure in the environment. In some cases, the operator should be notified of changes in robot health; in others, this would be an unnecessary distraction.

Independently-organized collaboration would allow the robots to receive individual commands that require collaboration and then work out between themselves how these tasks would be accomplished, without relying on a central source of instructions that may have incomplete or inaccurate information.

Integrated research in the two fields of geographically distributed work and robotics for the first time

Research in geographically distributed work had not previously been extended to include field robotics, and the use of concepts from this interdisciplinary field had not been a formal part of any previous robot interaction design. For example, “remote work,” a standard term in the distributed work field, had been used to refer to a situation in which a human worker worked remotely from the significant support and resource structure common in a collocated workplace. With the addition of field robotics to the disciplines contributing to the study of geographically distributed work, “remote work” may now also mean the work done, for a human by a robot, in a location where that human is not and may not ever go.

This is the first application of any body of humans-only teamwork research to teams with robots. Although the concept of the “human-robot team” is often given as a goal in robotics research, most robot designers do not have significant training in the theory underlying teams and teamwork. Their definitions are formed by their personal experience with teams, which likely lack the objective perspective, scope, and insight that formal research in this area provides. This research identified the field of geographically distributed work as a source of valuable teamwork research and applied its concepts to the development of a human-robot interaction framework.

8.4 Conclusion

Many current educational and recreational computer applications mimic the relationship between field robots and their operators. Future field robot operators will have thus grown up interacting with computer simulations of field robots, and they will expect to be able to interact with real robots as naturally as they had with these simulations. Yet, the fundamental differences between a virtual world and the real world cause problems not previously addressed by research in human-system interaction for robots, and the proper role of the operator in such systems has now only begun to be understood. This dissertation opens the door to further study on the identification and resolution of these important research issues.

Bibliography

[1] T. Sheridan, *Telerobotics, Automation, and Supervisory Control.* Cambridge, MA: MIT Press, 1992.

[2] A. Mikhailov, "Automation or Astronaut?," *Acta Astronautica*, vol. 1, pp. 557-559, 1974.

[3] B. Wilcox, L. Matthies, D. Gennery, B. Cooper, T. Nguyen, T. Litwin, A. Mishkin, and H. Stone, "Robotic vehicles for planetary exploration," presented at IEEE International Conference on Robotics and Automation, Nice, France, 1992, pp. 175-180.

[4] J. Rife and S. Rock, "Field experiments in the control of a jellyfish tracking ROV," presented at IEEE OCEANS Conference, 2002, pp. 2031-2038.

[5] J. Wilson, "Unmanned helicopters begin to deliver," in *Aerospace America*, vol. 37, 1999, pp. 38-42.

[6] B. Iannotta, "UCAVs prepare for battle," in *Aerospace America*, vol. 39, 2001, pp. 28-32.

[7] H. Wang, "OTTER: The design and development of an intelligent underwater robot," *Autonomous Robots*, vol. 3, pp. 297-320, 1996.

[8] J. Casper, *Human-Robot Interactions during the Robot-Assisted Urban Search and Rescue Response at the World Trade Center*, Masters thesis, Computer Science and Engineering, University of South Florida, 2002.

[9] R. Arkin, T. Collins, and Y. Endo, "Tactical Mobile Robot Mission Specification and Execution," presented at Mobile Robots XIV, Boston, MA, 1999.

[10] J. Hornbuckle, "Unmanned Aerial Vehicles in the U. S. Coast Guard," presented at AUVS-93, Washington, DC, 1993, pp. 79-94.

[11] W. Doggett, "Robotic Assembly of Truss Structures for Space Systems and Future Research Plans," presented at IEEE Aerospace Conference, Big Sky, MT, 2002.

[12] D. Shirley, "Touching Mars: 1998 Status of the Mars Robotic Exploration Program," *Acta Astronautica*, vol. 45, pp. 249-265, 1999.

[13] G. Sukhatme and M. Mataric, "Robots: intelligence, versatility, adaptivity," *Communications of the ACM*, vol. 45, pp. 30-32, 2002.

[14] B. Hall and B. Wasel, "TCS: A Common Control Interface," presented at AUVSI '98, Huntsville, AL, 1998, pp. 453-475.

[15] C. Dominguez and D. Hoagland, "Focus on Integration: Myths and Admonitions for UAV Designers," presented at AUVSI '98, Huntsville, AL, 1998, pp. 411-429.

[16] T. Sheridan, "Speculations on the Future Relations Between Humans and Automation," in *Automation and Human Performance: Theory and Applications*, R. Parasuraman and M. Mouloua, Eds. Mahwah, NJ: Erlbaum Associates, 1996, pp. 449-560.

[17] "Uninhabited Air Vehicles," National Research Council, Washington, DC, 2000.

[18] B. Boehm, "A Spiral Model of Software Development and Enhancement," in *SIGSOFT Software Engineering Notes*, 1986, pp. 22-42.

[19] D. Gentner and J. Grudin, "Why good engineers (sometimes) create bad interfaces," presented at CHI '90, Seattle, WA, 1990, pp. 277-282.

[20] D. Norman and S. Draper, *User centered system design: new perspectives on human-computer interaction*. Hillsdale, NJ: Lawrence Erlbaum Associates, 1986.

[21] D. Norman, *The Design of Everyday Things*. New York, NY: Doubleday, 1990.

[22] C. Monson, M. Hornsby, and D. Britton, "The Development of Specifications and Guidelines for the Design of Crew Stations for UAV Systems," presented at AUVSI '98, Huntsville, AL, 1998, pp. 395-410.

[23] S. Schneider and R. Cannon, "Experimental Object-Level Strategic Control with Cooperating Manipulators," *The International Journal of Robotics Research*, vol. 12, pp. 338-350, 1993.

[24] H. Wang, *Experiments in Semi-Autonomous Underwater Intervention Robotics*, PhD thesis, Aeronautics and Astronautics, Stanford University, Stanford, CA, 1996.

[25] H. Schubert, *Impedence Control of Flexible Macro/Mini Manipulators*, PhD thesis, Aeronautics and Astronautics, Stanford University, Stanford, CA, 2000.

[26] G. Pardo-Castellote, *Experimental Integration of Planning and Control for an Intelligent Manufacturing Workcell*, PhD thesis, Electrical Engineering, Stanford University, Stanford, CA, 1995.

[27] D. Cannon, *Point-and-Direct Telerobotics: Object Level Strategic Supervisory Control in Unstructured Interactive Human-Machine System Environments*, PhD thesis, Mechanical Engineering, Stanford University, Stanford, CA, 1992.

[28] "Tactics," in *United States Marine Corps Handbook*. Washington, DC: United States Marine Corps, 1955, pp. 91-94.

[29] T. Fong, *Collaborative Control: A Robot-Centric Model for Vehicle Teleoperation*, PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 2001.

[30] *IEEE Standard Computer Dictionary*, 92 Edition ed: IEEE Standard Office, 1991.

[31] R. Arkin and T. Balch, "AuRA: Principles and Practice in Review," *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 9, pp. 175-189, 1997.

[32] L. Parker, "Multi-Robot Team Design for Real-World Applications," in *Distributed Autonomous Robotic Systems 2*, H. Asama, T. Fukuda, T. Arai, and I. Endo, Eds. Tokyo, Japan: Springer-Verlag, 1996, pp. 91-102.

[33] H. Surmann and M. Theissinger, "ROBODIS: A dispatching system for multiple autonomous service robots," presented at Field and Service Robots '99, Pittsburgh, PA, 1999, pp. 168-173.

[34] K. Dixon, J. Dolan, W. Huang, C. Paredis, and P. Khosla, "RAVE: A Real and Virtual Environment for Multiple Mobile Robot Systems," presented at International Conference on Intelligent Robots and Systems, Seoul, Korea, 1999.

[35] K. Ohkawa, T. Shibata, and K. Tanie, "Method for controlling a group of robots by an operator," in *Distributed Autonomous Robotic Systems 3*, T. Lueth, R. Dillmann, P. Dario, and H. Worn, Eds. Berlin, Germany: Springer-Verlag, 1998, pp. 309-318.

[36] Y. Anzai, "Human-Robot-Computer Interaction in Multiagent Environment," presented at 5th International Conference on Human-Computer Interaction, Orlando, FL, 1993, pp. 2-7.

[37] S. Fleischer, S. Rock, and J. Lee, "Underwater vehicle control from a Virtual Environment Interface," presented at Symposium on Interactive 3D Graphics, Monterey, CA, 1995.

[38] H. Jones, E. Frew, B. Woodley, and S. Rock, "Human-Robot Interaction for Field Operation of an Autonomous Helicopter," presented at Mobile Robots XIII, Boston, MA, 1998, pp. 244-252.

[39] M. Adams, S. Kolitz, and S. Rasmussen, "An Automation-Centered Human-System Integration Architecture for Autonomous Vehicles," presented at AUVSI '98, Huntsville, AL, 1998, pp. 385-394.

[40] H. Wang, R. Marks, S. Rock, and M. Lee, "Task-Based Control Architecture for an Untethered, Unmanned Submersible," presented at 8th Annual Symposium of Unmanned Untethered Submersible Technology, Marine Systems Engineering Laboratory, Northeastern University, 1993, pp. 137-147.

[41] H. Stevens, E. Miles, S. Rock, and R. Cannon, "Object-Based Task-Level Control: A hierarchical control architecture for remote operation of space robots," presented at AIAA/NASA Conference on Intelligent Robotics in Field, Factory, Service, and Space, Houston, TX, 1994, pp. 264-273.

[42] D. Lees, *A Graphical Programming Language for Service Robots in Semi-Structured Environments*, PhD thesis, Mechanical Engineering, Stanford University, Stanford, CA, 1994.

[43] E. Guglielmelli, C. Laschi, G. Teti, R. Fontanelli, and P. Dario, "A modular and distributed supervisory system for a semi-autonomous personal robot for household applications," presented at International Conference on Autonomous Robotics (ICAR) '97, Monterey, CA, 1997, pp. 45-50.

[44] A. Green, H. Huttenrauch, M. Norman, L. Oestreicher, and K. Eklundh, "User-Centered Design for Intelligent Service Robots," presented at IEEE International Workshop on Robot and Human Interactive Communication, Osaka, Japan, 2000, pp. 161-166.

[45] Interview with Staff Sargeant Antonio Mitchell, US Army, conducted by H. Jones on July 10, 2002 in Orlando, FL.

[46] T. Collins, R. Arkin, M. Cramer, and Y. Endo, "Field Results for Tactical Mobile Robot Missions," *Georgia Tech Mobile Robot Lab* online article, www.cc.gatech.edu/ai/robot-lab/online-publications/field_results.pdf, 2000.

[47] J. Zelek, "Human-Robot Interaction with a Minimal Spanning Natural Language Template for Autonomous and Tele-operated Control," presented at International Conference on Intelligent Robots and Systems, Grenoble, France, 1997, pp. 299-305.

[48] Y. Cao, A. Fukunaga, and A. Kahng, "Cooperative Mobile Robotics: Antecedents and Directions," *Autonomous Robots*, vol. 4, pp. 1-23, 1997.

[49] R. Gilson, C. Richardson, and M. Mouloua, "Key Human Factors Issues for UAV/UCAV Mission Success," presented at AUVSI '98, Huntsville, AL, 1998, pp. 477-484.

[50] T. Ishikawa, K.Kawabata, Y. Ueda, H. Asama, and I. Endo, "Graphical user interface for collaborative system of human and mobile robots with sensors," in *Distributed Autonomous Robotic Systems 3*, T. Lueth, R. Dillmann, P. Dario, and H. Worn, Eds. Berlin, Germany: Springer-Verlag, 1998, pp. 319-327.

[51] T. Suzuki, T. Fujii, H. Asama, K. Yokota, H. Kaetsu, N. Mitomo, and I. Endo, "Cooperation between a human operator and multiple robots for maintenance tasks at a distance," in *Distributed Autonomous Robotic Systems 2*, H. Asama, T. Fukuda, T. Arai, and I. Endo, Eds. Berlin, Germany: Springer-Verlag, 1996, pp. 50-59.

[52] C. Miller, M. Pelican, and R. Goldman, ""Tasking" Interfaces to Keep the Operator in Control," presented at Fifth Annual Symposium on Human Interaction with Complex Systems, Urbana-Champaign, IL, 2000.

[53] D. Gage, "Command Control for Many-Robot Systems," in *Unmanned Systems*, vol. 10, 1992, pp. 28-34.

[54] N. Geddes and R. Lee, "Intelligent Control for Automated Vehicles," presented at AUVSI '98, Huntsville, AL, 1998, pp. 755-764.

[55] T. Payne, K. Sycara, and M. Lewis, "Varying the User Interaction with Multi-Agent Systems," presented at 4th International Conference on Autonomous Agents, Barcelona, Spain, 2000, pp. 412-418.

[56] D. Brock, D. Montana, and A. Ceranowicz, "Coordination and Control of Multiple Autonomous Vehicles," presented at IEEE Conference on Robotics and Automation, Nice, France, 1992.

[57] R. Moore, J. Dowding, H. Bratt, M. Gawron, Y. Gorfu, and A. Cheyer, "CommandTalk: A Spoken-Language Interface for Battlefield Simulations," presented at Fifth Conference on Applied Natural Language Processing, Washington, DC, 1997, pp. 1-7.

[58] "MAGIC2: Multiple Aircraft GPS Integrated Command and Control System," presented at AUVSI '98, Huntsville, AL, 1998, pp. 235-249.

[59] J. Gonzalez, J. Hollinger, J. Martin, and A. Iasiello, "BUGS: A Successful Team of Small Robots for Ordnance Disposal," presented at AUVSI '02, Orlando, FL, 2002.

[60] R. Aylett, A. Coddington, D. Barnes, and R. Ghanea-Hercock, "Supervising multiple cooperating mobile robots," presented at Autonomous Agents 97, Marina Del Rey, CA, 1997, pp. 514-15.

[61] J. Bay, L.Borelli, K. Chapman, and T. Harrold, "User Interface and Display Management Design for Multiple Robot Command and Control," presented at Mobile Robots XV, Boston, MA, 2000, pp. 57-66.

[62] M. Saptharishi, K. Bhat, C. Diehl, J. Dolan, and P. Khosla, "CyberScout: Distributed Agents for Autonomous Reconnaissance and Surveillance," presented at Mechatronics and Machine Vision in Practice 2000, Hervey Bay, Austrailia, 2000, pp. 93-100.

[63] W. Dickson, *Experiments in Cooperative Manipulation of Objects by Free-Flying Robot Teams*, PhD thesis, Aeronautics and Astronautics, Stanford University, Stanford, CA, 1993.

[64] H. Schubert and J. How, "Space construction: an experimental testbed to develop enabling technologies," presented at Telemanipulator and Telepresence Technologies IV, Pittsburgh, PA, 1997.

[65] T. Fong, C. Thorpe, and C. Baur, "Collaboration, Dialogue, and Human-Robot Interaction," presented at International Symposium of Robotics Research 2001, Victoria, Austrailia, 2001.

[66] K. Ali, *Multiagent Telerobotics: Matching Systems to Tasks*, PhD thesis, College of Computing, Georgia Institute of Technology, Atlanta, GA, 1999.

[67] R. Parasuraman and V. Riley, "Humans and Automation: Use, Misuse, Disuse, and Abuse," *Human Factors*, vol. 39, pp. 230-253, 1997.

[68] B. Muir and N. Moray, "Trust in Automation. Part II. Experimental studies of trust and human intervention in a process control simulation," *Ergonomics*, vol. 39, pp. 429-460, 1996.

[69] "SWAT Training Course: Command Post Information Processing," Alameda County Sheriff's Department, Oakland, CA, 1998.

[70] H. Jones, S. Rock, D. Burns, and S. Morris, "Autonomous Robots in SWAT Applications: Research, and Design, and Operations Challenges," presented at AUVSI '02, Orlando, FL, 2002.

[71] I. Greif, "Computer-supported cooperative work: A book of readings." San Mateo, CA: Morgan Kaufmann, 1988.

[72] J. Blomberg, J. Giacomi, A. Mosher, and P. Swenton-Hall, "Ethnographic Field Methods and Their Relation to Design," in *Participatory design: Principles and practices*, D. Schuler and A. Makioka, Eds. Hillsdale, NJ: Erlbaum, 1993, pp. 123-156.

[73] J. Van Maanen, *Tales of the field*. Chicago: University of Chicago Press, 1988.

[74] B. Whiting and J. Whiting, "Methods for observing and recording behavior," in *Handbook of method in cultural anthropology*, R. Naroll and R. Cohen, Eds. New York: Columbia University Press, 1970, pp. 282-315.

[75] J. Jackson, ""Deja entendu": The liminal qualities of anthropological fieldnotes," *Journal of Contemporary Ethnography*, vol. 19, pp. 8-43, 1990.

[76] H. Beyer and K. Holtzblatt, *Contextual Design: Defining customer-centered systems*. San Francisco, CA: Morgan Kaufman, 1998.

[77] S. Fussell, R. Kraut, and J. Siegel, "Coordination of communication: Effects of shared visual context on collaborative work," presented at Conference on Computer-Supported Cooperative Work, Philadelphia, PA, 2000, pp. 21-30.

[78] J. Herbsleb, A. Mockus, T. Finholt, and R. Grinter, "Distance, dependencies, and delay in a global collaboration," presented at Conference on Computer-Supported Cooperative Work, Philadelphia, PA, 2000, pp. 319-328.

[79] G. Mark, "Merging multiple perspectives in groupware use: Intra- and intergroup conventions," presented at SIGGROUP, Phoenix, AZ, 1997, pp. 19-28.

[80] T. Allen, *Managing the Flow of Technology*. Cambridge, MA: MIT Press, 1977.

[81] R. Kraut, C. Egido, and J. Galegher, "Patterns of contact and communication in scientific research collaborations," in *Intellectual Teamwork*, J. Galegher, R. Kraut, and C. Egido, Eds. Hillsdale, NJ: Lawrence Erlbaum, 1990, pp. 149-172.

[82] H. Clark and D. Wilkes-Gibbs, "Referring as a collaborative process," *Cognition*, vol. 22, pp. 1-39, 1986.

[83] R. Kraut, S. Fussell, S. Brennan, and J. Siegel, "Understanding the effects of proximity on collaboration: Implications for technologies to support remote collaborative work," in *Distributed Work*, P. Hinds and S. Kiesler, Eds. Cambridge, MA: MIT Press, 2002, pp. 137-164.

[84] G. Olson and J. Olson, "Distance matters," *Human Computer Interaction*, vol. 15, pp. 139-179, 2000.

[85] D. Armstrong and P. Cole, "Managing distances and differences in geographically distributed work teams," in *Diversity in work teams: Research paradigms for a changing workplace*, S. Jackson and M. Ruderman, Eds. Washington, DC: American Psychological Association, 1995, pp. 187-216.

[86] C. Cramton, "The mutual knowledge problem and its consequences for dispersed collaboration," *Organization Science*, vol. 12, pp. 346-371, 2001.

[87] J. Olson and S. Teasley, "Groupware in the wild: Lessons learned from a year of virtual collocation," presented at Conference on Computer-Supported Cooperative Work, Boston, MA, 1996, pp. 419-427.

[88] R. Grinter, J. Herbsleb, and D. Perry, "The geography of coordination: Dealing with distance in R&D work," presented at SIGGROUP, Phoenix, AZ, 1999, pp. 306-315.

[89] K. Eisenhardt, "High reliability organizations meet high velocity environments: Common dilemmas in nuclear power plants, aircraft carriers, and microcomputer firms," in *New Challenges to Understanding Organizations*, K. Roberts, Ed. New York, NY: Macmillan, 1993, pp. 117-135.

[90] K. Weick, "The collapse of sensemaking in organizations: The Mann Gulch disaster," *Administrative Science Quarterly*, vol. 38, pp. 628-652, 1993.

[91] S. Weisband, "Maintaining awareness in distributed team collaboration: Implications for leadership and performance," in *Distributed Work*, P. Hinds and S. Kiesler, Eds. Cambridge, MA: MIT Press, 2002, pp. 311-333.

[92] J. Keegan, *The Mask of Command*, 1st American ed. New York, N.Y.: Viking, 1987.

[93] "Command and Control," in *United States Marine Corps Handbook*. Washington, DC: United States Marine Corps, 1955, pp. 91-94.

[94] J. Keegan, *The Face of Battle*. New York: Viking Press, 1976.

[95] H. Jones and P. Hinds, "Extreme Work Groups: Using SWAT Teams as a Model for Coordinating Distributed Robots," presented at Conference on Computer-Supported Cooperative Work, New Orleans, LA, 2002, pp. 372-381.

[96] M. Becker, E. Kefalea, E. Maeel, C. Malsburg, J. Triesch, J. Vorbrueggen, R. Wuertz, and S. Zadel, "GripSee: a gesture-controlled robot for object perception and manipulation," *Autonomous Robots*, vol. 6, pp. 203-221, 1999.

[97] S. Dockstader and A. Tekalp, "On the tracking of articulated and occluded video object motion," *Real-Time Imaging*, vol. 7, pp. 415-432, 2001.

[98] K. Valavanis, D. Gracanin, M. Matijasevic, R. Kolluru, and G. Demetriou, "Control Architectures for Autonomous Underwater Vehicles," *IEEE Control Systems*, vol. 17, pp. 48-64, 1997.

[99] M. Cox and M. Veloso, "Controlling for Unexpected Goals when Planning in a Mixed-Initiative Setting," in *Progress in Artificial Intelligence: Eighth Portuguese Conference on Artificial Intelligence*, E. Costa and A. Cardoso, Eds. Berlin, Germany: Springer, 1997, pp. 309-318.

[100] J. Crandall and M. Goodrich, "Experiments in adjustable autonomy," presented at IEEE Conference on Systems, Man, and Cybernetics, Tucson, AZ, 2001, pp. 1624-1629.

[101] J. Hinchman, "Why Go Distributed? An exploration into distributed control," presented at AUVSI '01, Baltimore, MD, 2001.

[102] M. Ullman, *Experiments in Autonomous Navigation and Control of Multi-Manipulator Free-Flying Space Robots*, PhD thesis, Aeronautics and Astronautics, Stanford University, Stanford, CA, 1993.

[103] C. Bowers, R. Oser, E. Salas, and J. Cannon-Bowers, "Team Performance in Automated Systems," in *Automation and Human Performance: Theory and Applications*, R. Parasuraman and M. Mouloua, Eds. Hillsdale, NJ: Lawrence Erlbaum, 1996, pp. 243-261.

[104] D. Miles, *Real-Time Dynamic Trajectory Optimization with Application to Free-Flying Space Robots*, PhD thesis, Aeronautics and Astronautics, Stanford University, Stanford, CA, 1997.

[105] R. Kindel, D. Hsu, J.-C. Latombe, and S. Rock, "Kinodynamic Motion Planning Amidst Moving Obstacles," presented at IEEE Conference on Robotics and Automation, San Francisco, CA, 2000.

[106] E. Miles, *A Real-Time Human Perception Interface for Task-Level Control of a Robot in Unfamiliar Environments*, PhD thesis, Aeronautics and Astronautics, Stanford University, Stanford, CA, 1997.

[107] L. Alder, *Control of a Flexible-Link Robotic Arm Manipulating an Unknown Dynamic Payload*, PhD thesis, Aeronautics and Astronautics, Stanford University, Stanford, CA, 1993.

[108] J. Russakow, *Experiments in Manipulation and Assembly by Two-Arm, Free-Flying Space Robots*, PhD thesis, Mechanical Engineering, Stanford University, Stanford, CA, 1995.

[109] A. Robertson, *Control System Design for Spacecraft Formation Flying: Theory and Experiment*, PhD thesis, Aeronautics and Astronautics, Stanford University, Stanford, CA, 2001.

[110] T. Corazzini, A. Robertson, J. Adams, A. Hassibi, and J. How, "GPS Sensing for Spacecraft Formation Flying," presented at Institute of Navigation GPS-97 Conference, Kansas City, MO, 1997.

[111] S. Singhal and M. Zyda, *Networked Virtual Environments: Design and Implementation*. New York, NY: ACM Press, 1999.

[112] O. Lemon, A. Bracy, A. Gruenstein, and S. Peters, "Information States in a Multi-Modal Dialogue System for Human-Robot Conversation," presented at Bi-Dialog, 5th Workshop on Formal Semantics and Pragmatics of Dialogue, 2001, pp. 57-67.

[113] K. Goldberg, S. Bui, B. Chen, B. Farzin, J. Heitler, D. Poon, R. Solomon, and G. Smith, "Collaborative Teleoperation on the Internet," presented at IEEE International Conference on Robotics and Automation, San Francisco, CA, 2000.

[114] B. Schneiderman, "Direct manipulation for comprehensible, predictable, and controllable user interfaces," presented at ACM International Workshop on Intelligent User Interfaces '97, Orlando, FL, 1997, pp. 33-39.

[115] P. Maes, "Agents that reduce work and information overload," in *Software Agents*, J. Bradshaw, Ed. Cambridge, MA: MIT Press, 1994, pp. 145-164.

[116] M. Lewis, "Designing for Human-Agent Interaction," in *AI Magazine*, vol. 19, 1998, pp. 67-78.

[117] D. Chao, "Doom as an Interface for Process Management," presented at CHI 2001, Seattle, WA, 2001, pp. 152-157.

[118] D. Laprad, "Real War," *Adrenaline Vault* online article, http://www.avault.com/featured/realwar/index.asp, 2001.

[119] H. Jones and M. Snyder, "Supervisory Control of Multiple Robots based on a Real-Time Strategy Game Interaction Paradigm," presented at IEEE Conference on Systems, Man, and Cybernetics, Tucson, AZ, 2001, pp. 383-388.

[120] T. Winograd, "Computer Software for Working with Language," in *Scientific American*, vol. 251, 1984, pp. 230-245.

[121] S. Young and C. Proctor, "The design and implementation of dialogue control in voice-operated database inquiry systems," *Computer Speech and Language*, vol. 3, pp. 329-353, 1989.

[122] N. Yankelovich, "How Do Users Know What to Say?," in *Interactions*, vol. 3, 1996, pp. 32-42.

[123] J. Fry, H. Asoh, and T. Matsui, "Natural Dialogue with the Jijo-2 Office Robot," presented at International Conference on Intelligent Robots and Systems, Victoria, BC, 1998, pp. 1278-1283.

[124] J. Gibson, *The Ecological Approach to Visual Perception*. Hillsdale, NJ: Lawrence Erlbaum, 1986.

[125] R. St. Amant, "Planning and User Interface Affordances," presented at Intelligent User Interfaces 1999, Redondo Beach, CA, 1999, pp. 135-142.

[126] C. Clark and S. Rock, "Randomized Motion Planning for Groups of Nonholonomic Robots," presented at 6th International Symposium on Artificial Intelligence, Robotics, and Automation in Space, Montreal, Canada, 2001.

[127] G. Pardo-Castellote and S. Schneider, "The Network Data Delivery Service: A Real-Time Connectivity System," presented at AIAA/NASA Conference on Intelligent Robots in Field, Factory, Service, and Space, Houston, TX, 1994, pp. 591-597.

[128] Y. Labrou, T. Finin, and Y. Peng, "Agent Communication Languages: The Current Landscape," in *IEEE Intelligent Systems*, vol. 14, 1999, pp. 45-52.

[129] E. Frew and S. Rock, "Exploratory Motion Generation for Monocular Vision-Based Target Localization," presented at 2002 Aerospace Conference, Big Sky, MT, 2002.

[130] E. Gamma, R. Johnson, and G. Booch, *Design Patterns*. New York, NY: Addison-Wesley, 1994.

[131] R. Zeleznik, J. La Viola, D. Feliz, and D. Keefe, "Pop through button devices for VE navigation and interaction," presented at IEEE Virtual Reality Conference, Orlando, FL, 2002, pp. 127-134.

[132] M. Pimentel, "Design of a Contigency Management on the UCAV," presented at AUVSI '02, Orlando, FL, 2002.

[133] K. Reichard, E. Crow, and J. Stover, "Self-Awareness, Monitoring and Diagnosis for Autonomous Vehicle Operations," presented at AUVSI '02, Orlando, FL, 2002.

[134] S. Thayer and A. Morris, "Scalable Command and Control: Optimal Configuration of Multi-Robot, Multi-Operator Work Systems," presented at AUVSI '02, Orlando, FL, 2002.

[135] M. Howell, "Teaming of Manned and Unmanned Systems," presented at AUVSI '98, Huntsville, AL, 1998, pp. 669-689.

[136] B. John, "Why GOMS?," in *Interactions*, vol. 2, 1995, pp. 80-89.

[137] J. Grudin, "The computer reaches out: The historical continuity of interface design," presented at CHI '90, Seattle, WA, 1990, pp. 261-268.

[138] F. Pipitone, B. Kamgar-Parsi, and R. Hartley, "Three dimensional computer vision for micro air vehicles," presented at Enhanced and Synthetic Vision, Orlando, FL, 2001, pp. 189-197.

[139] N. Kruger, D. Wendorff, and G. Sommer, "Two modules of a vision-based robotic system: attention and accumulation of object representations," presented at International Workshop on Robot Vision, Auckland, NZ, 2001, pp. 219-226.