Dialogue-Based Human-Robot Interaction for Space Construction Teams[1, 2]

Hank Jones, Stephen Rock
Aerospace Robotics Laboratory
Stanford University
250 Durand Building
Stanford, CA 94305
650-723-3608
hlj,rock@arl.stanford.edu

Abstract—This paper describes a human-robot interaction that uses dialogues as a basis for the operation of multiple robots in space construction. The dialogues, which are conducted by the operator and a community of software agents, consist of explicit and implicit queries and responses regarding the state of the robots and their environment.

A dialogue enables a high-level but active role for the operator in resource management and task planning for space construction missions with multiple robots. The dynamic nature of such a scenario will be challenging for the operator, but a dialogue interaction provides valid, up-to-date information about robot capabilities that make up the tools that the operator may use to solve problems creatively.

This interaction enables the management of large teams of multiple robots through methods that are natural for the operator. Experiments demonstrate the utility of this method of robot operation, and point out some of the challenges that remain for future research.

TABLE OF CONTENTS

1. INTRODUCTION

Future space structures will likely be constructed by many humans and robots working together as a team. Whether the humans work alongside the robots or from remote stations on Earth, the robots will require continuous observation and direction from ground operators. Current robotic systems have many operators for one robot. Future systems will have one operator for many robots. The development of this capability requires research in many areas, including the development of the interaction between the operator and the robots.

We propose an interaction based on dialogues between the human and the robots as an effective method for operating multiple robots. However, developing a robot system capable of conducting a dialogue with an operator is a challenging task. There are many issues to be addressed, including:

- *Establishing the structure and scope of the dialogue*
- *Creating a robot infrastructure capable of conducting an effective dialogue*
- *Determining methods for dealing with the social conventions of dialogues*
- *Developing an interface that allows the operator to carry out the dialogue with the robotic system.*

The purpose of this paper is to describe our implementation of a dialogue interaction that demonstrates the utility of this approach. Section 3 outlines the challenges of developing a robot that is able to conduct a dialogue and then provides details of the steps we took to create a capable robotic dialogue partner. Section 4 describes other issues inherent in implementing any dialogue and our methods used to resolve these problems. Sections 5 and 6 describe the results of this implementation of a dialogue interaction and summarize our findings and ideas for future work. Relevant concepts from non-robotics fields are discussed in Section 2, and the remainder of this section describes the related research in robotics.

Related Work

There have been a wide variety of human-system interfaces for single complex robots. Autonomous helicopters have been controlled using point-and-click [1] and virtual dashboard [2] techniques; autonomous underwater vehicles and space vehicles have been directed using virtual environments [3] and high-level tasking [4, 5]; intelligent arms have been instructed using gestures [6] and graphical icons [7]; and many complex robots have been fully teleoperated [8].

[1] 0-7803-7231-X/01/$10.00/© 2002 IEEE
[2] IEEEAC paper #322, Updated November 13, 2001

However, if one operator were expected to command **multiple** complex robots, none of these methods would readily scale to accommodate the additional requirements for information display and operator input. Direct teleoperation, for instance, would either overstress the operator or underutilize the robots [9]. The robot interfaces of more automated systems, such as those using control panel or dashboard metaphors, are reproductions of physical control mechanisms for single entities, and they do not appear to extend naturally to multiple robots.

Current research regarding emergent or reactive control of multiple robot systems is concerned mostly with answering the challenge of getting these new systems to operate successfully, and has not yet been able to address fully the role of the human in system operation. Most architectures, such as AuRA [10] and ALLIANCE [11], focus on strengthening the autonomous capabilities of the robot teams rather than their operation by humans. The research that has incorporated the human operator, such as ROBODIS [12], RAVE [13], MokSAF [14], MissionLab [15], and Fong's dialogue-based queries at CMU [16], have largely concentrated on methods of cooperative motion and task planning for surveillance and exploration, with the user utilized either for initial planning or for perception assistance during operation.

There have been a small number of research programs that have focused on the human-system interaction for systems with multiple robots that can accept more complex mission goals than behavior-based robots. Purely virtual but complex robots were operated in DARPA's SIMNET [17], and a high-level tasking "playbook" interface has been developed and tested for future operation of uninhabited combat air vehicles [18]. The MAGIC2 system, developed for operational control of unmanned air vehicles [19], and the MACTA hybrid agent/reactive architecture [20] have demonstrated operation of multiple complex robots experimentally. MAGIC2 combines control panels for the control of unmanned aerial vehicles but appears to be limited to a maximum of four vehicles per operator. MACTA focuses on behavior scripts and their ability to satisfy human-designated goals.

Two research programs have addressed the design and implementation of a human-system interaction for field robots from a human factors or usability perspective. Ali at Georgia Tech [21] ran more than 100 people through tests that measured the safety, effectiveness, and ease-of-use of operational paradigms that vary the amount of automation and the robot group size. He concluded that supervisory control was effective for multiple robot systems, although its utility was affected by the nature of the task. However, system constraints limited the depth of the study.

The second program, the DARPA TMR program at Georgia Tech, expanded their MissionLab development environment to accommodate formal usability testing [15]. By recording user actions during pre-mission planning, they have generated data that can be used to refine the design interface itself. However, the results of the usability tests have not yet been fully analyzed or incorporated into subsequent systems.

2. TEAMWORK, AUTOMATION, AND DIALOGUES

We propose a method of multiple robot operation by one person by patterning the interaction between operator and robots after the task-oriented dialogues common in human teams. This section outlines the research that supports the utility of dialogues in teams of humans. Our hypothesis, supported by the interaction implementation this paper describes, is that dialogues can also play a useful role within human-robot teams.

Research in related non-robotics fields

To form effective teams, human team members must communicate clearly about their goals and abilities. Studies of cooperation among spatially distributed teams of human workers have shown that frequency of communications regarding task achievements and plans is a strongly determining factor in team success. This dialogue boosts performance by increasing trust among the team members [22]. The utility of dialogues in human-robots teams is to similarly increase the trust of the operator in the robots under command.

Another study of teams of humans [23] characterized the steps of the team performance process as Forming (determining who would be on the team), Storming (finding out the strengths and weaknesses of team members, and characterizing the tasks to be done), Norming (distributing tasks to the team members for execution), and Performing (execution of responsibilities). Frequent communication establishes the roles that team members are capable of playing and determines role assignments. In the case of a field robot deployment, the Forming and Norming steps would proceed iteratively throughout the robot deployment through an ongoing dialogue with the operator.

Studies of human use of automation, particularly supervisory control of flight control systems and power plant processes, also provide suggestions about how a robot interaction might be designed for effective use. Trust, a variable that may be increased by a dialogue-based interaction, was shown to be a significant factor in determining automation use [24, 25]. Research regarding the impact that automation has on teams is scarce but suggests that automation not explicitly designed for interaction with the team will lead to decreased overall performance [26]. These studies indicate that mechanisms to increase trust could play an important role in increasing the performance of human-robot teams.

Based on the conclusions of these studies in human teamwork and automation use, a reasonable expectation would be for robots designed with a dialogue-based interaction to engender greater trust in the robots under command, and consequently lead to more effective and productive human-robot teams.

3. USABILITY PRINCIPLES IN ROBOT DESIGN

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" [27]. 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. However, the application of usability principles is a new direction for field robot development.

Affordances

One of the most important aspects of good usability design is the application of proper affordances [28]. An affordance provides the user with obvious and simple access to the sort of operations and manipulations that can be done to a particular object. A field robot example of an affordance is a point-and-click procedure that commands the robot to move to a certain location. With most field robot systems, the affordances are set once the code for the interface to the robot has been written. However, in the example above, mechanical problems or obstacles may prevent the robot from moving to the location specified. A more adaptable interaction would allow the actions afforded to the operator to change dynamically as the abilities of the robot change.

Although robotics researchers have explored many ways to express task- and mission-level instructions to robots, we have identified this remaining fundamental issue -- the team leader should be provided with task choices that are legitimate for the robots. A dialogue-based interaction addresses this issue by providing a natural method to create valid, appropriate affordances that allow operators to give orders consistent with the robot state.

Mental models

Mental models, another usability design concept, should be incorporated into human-robot interaction design. Leaders of human teams naturally form mental models of an ongoing mission and its anticipated results, and of the capabilities and expected performance of subordinates. The leader then uses these mental models to efficiently distribute current and future tasks among the team.

For multiple robots, the operator will likely consider the robots under command as agents to be used to cause changes to the world, yet the changes that can be made are limited to what the robots are actually capable of doing at that instant. Consequently, the goal of the interaction is to help the operator's mental model remain as consistent as possible with the actual robot and environment state.

4. CREATING A CAPABLE DIALOGUE PARTNER

This section will focus on the issue of developing a robot able to conduct effective dialogues with the human operator. This issue actually consists of three separate challenges:

- Creating a means to understand incoming requests and statements (the robot needs to 'listen')
- Developing a structure that the robot can use to develop a proper response (the robot needs to 'think')
- Building a mechanism to allow the robot to convey information to other participants in the dialogue (the robot needs to 'speak')

Open-ended dialogues with computers are a topic of intensive study, but the purpose of this project was not to implement the state-of-the-art in this field. Instead, we identified a way to simplify the dialogue to transfer the appropriate information without unnecessary complication. This allowed us to more quickly satisfy our goal of developing a dialogue interaction involving field robots.

Knowledge representation

One of the most challenging issues for knowledge representation is deciding what to represent. Knowledge conveyed in dialogues, as a whole, could conceivably consist of an entire language, with representation necessary not only for the words but the meta-information such as tone and context. The use of such a dialogue was beyond the scope of this project. Instead, we took steps to make the dialogue more manageable.

We declared that the objective of all dialogues would be the construction of an imperative sentence of the form Subject-Verb-Direct Object (i.e. "Robot Three pick up the green block") that would serve as a command to the robot. This constraint has little effect on the operator's ability to provide an appropriate command to the robot, but it eliminates many types of conversations that might otherwise be attempted.

Furthermore, the dialogue is simplified because the subject in this sentence is always known – it is the robot. Maintaining the subject as a command component is important, though, for the operator since he may be operating many robots at once and need to be able to specify the robot to be used.

However, this sentence structure does imply that the robot is aware of objects to some extent. Classifying objects in a hierarchical structure is an intelligent method for representing objects. For example, in our system a cone is an inanimate (non-robot) object, and inanimate objects are

```
;; The robot itself is an Object
(=> (is_a FreeFlyer ?x ?p) (is_a Object ?x ?p))
;; Cones are Objects
(=> (is_a Cone ?x ?p) (is_a Object ?x ?p))
;; This information can be accessed via the following call by an agent, which
;; would return the names and properties of all objects known to this robot:
(is_a Object ?name ?props)

;; created to designate the most specific class of an object ;;;;;;;;
(=> (instance_of ?type ?name ?props) (is_a ?type ?name ?props))
(instance_of FreeFlyer huey (props(grey pos(0 0 0 0 0 0))))
(instance_of Cone cone-red (props(red pos(0 0 0 0 0 0))))
;; This information can be accessed via the following call by an agent, which
;; would return the specific class and properties of a given object:
(instance_of ?type cone-red ?props)

;; Can-watch something if it is an object and the camera is working
(=> (and (is_a Object ?x ?p) (state-Camera ok)) (CanWatch ?x))
;; Abstract over all actions  (Can action object)
;; Define Can as another form of the Able statement
(=> (Able (?a ?x)) (Can ?a ?x))
(=> (CanWatch ?x) (Able (watch ?x)))
;; Initial state
(state-Camera ok)
;; This information can be accessed via the following call by an agent, which
;; would return the actions (such as 'watch' in the example above) possible:
(Can ?action cone-red)
```

Figure 1 – Sample KIF text

obstacles. Consequently, a cone is an obstacle. Such a structure enables logical access to classes of objects. For instance, a logical statement can be applied to all obstacles ("obstacles have specific locations") or just to cones ("cones cannot be moved") without additional clarification.

The only remaining sentence element to be determined is the verb. However, this component is the most constrained of the sentence parts. Certain verbs only make sense for a subset of all the objects, and external information (such as fuel status) determine whether or not that task is possible at that instant. To create a dialogue partner that was able to reason about such constraints, we created logical sentences to describe the relationship between objects, robot state, and capabilities.

All of this information regarding the robot is written to a file in the Knowledge Interchange Format (KIF). KIF was chosen for its flexibility (it was designed for use in the interchange of knowledge among disparate computer systems), its readability by readers not familiar with logic syntax, and its status as an emerging standard.

An example of the KIF language used is given in Figure 1. Although a full description of KIF is available at http://logic.stanford.edu/kif/kif.html, the background necessary to read this code fragment (of one of the KIF knowledge bases used in this research) is given by three rules:

- A double semicolon (;;) serves to mark the remainder of the line as a remark.
- The format (=> (A) (B)) defines A as true if B is true.
- Simple statements such as (A B) equate all elements in the statement.

Thus, in the first declarative statement in Figure 1, a FreeFlyer is defined as being an Object, as is a Cone in the subsequent statement. This shows a simple example of the hierarchical structure used.

In the second set of sentences, the singular instances of the FreeFlyer object 'huey' and the Cone object 'cone-red' are created. This structure is used to allow subsequent calls to determine the class of an object by its unique name. This functionality is necessary because the name serves as the identifier for the object in many processes, and the object's class might be important information to retrieve. Likewise, the properties (color, position, orientation) of an object are acquired by explicitly asking the robot using the unique object name.

In the last set of sentences in Figure 1, the relationship between robot state, object types, and robot capabilities are explicitly defined. In the simple definition shown, the robot is told that it can watch an object if that object is defined as an Object and the camera is operational. More complicated definitions are possible, but have not yet been necessary.

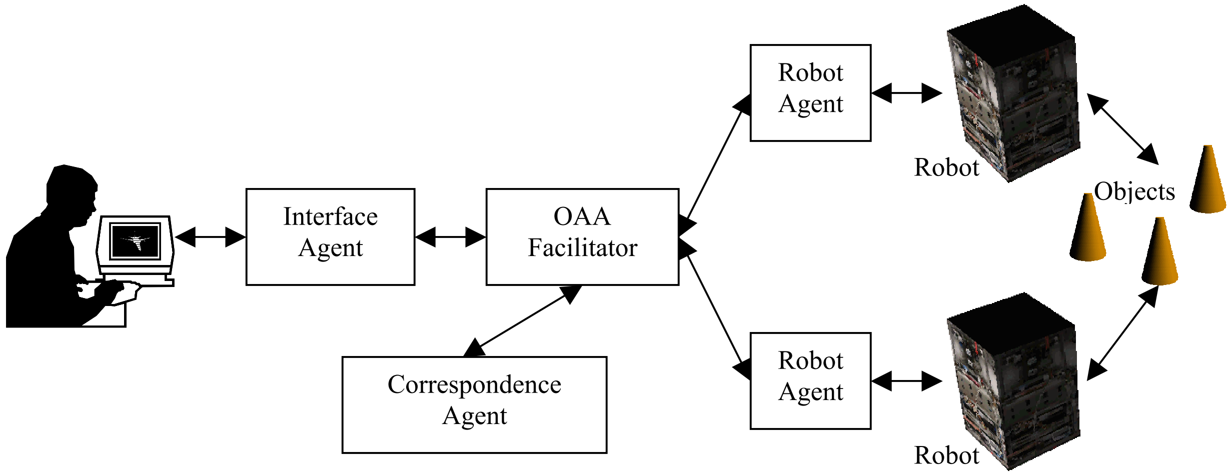


Figure 2 – System diagram

Theorem prover

At robot start, the KIF knowledge base file describing the robot is read and subsequently managed by the Java Theorem Prover (JTP) written by Gleb Frank from the Knowledge Systems Laboratory at Stanford University (http://www.stanford.edu/~gkfrank/jtp/). JTP is a full first-order logic theorem prover that allows statements to be given, tested and sets of solutions produced. JTP was chosen because it works readily with knowledge bases written in KIF, and assistance with its use was easily available.

Robot agent

To implement the 'listening' and 'speaking' components of the dialogue, we wrapped the JTP prover with a software agent that resides on each robot. This agent is responsible for maintaining knowledge of the state of the robot, the knowledge of which objects the robot senses, and the tasks that the robot is capable of performing given the robot state and the objects present. This agent should directly contact low-level monitoring systems for information about robot state and the objects that are sensed by the robots. However, currently this state information is supplied directly to the agent through statements to the JTP prover.

The JTP knowledge base can be queried by any member of the agent community using a call to the agent architecture described below. For example, a JTP query of the form (is_a Object ?name ?props) will inquire about any objects in the KIF knowledge base that satisfy this form, which in this case would be anything that is an Object (a question mark before a term indicates that it is a variable). The response comes in the form of a delimited text string with all solutions from each robot agent that could process the request.

5. Implementing a Dialogue

Our resolution of the other significant issues for the development of the dialogue-based interaction, such as the communication method, turn-taking rules, and a process for clarification of ambiguous information, are described briefly in this section.

Communications infrastructure

The Open Agent Architecture (OAA) developed at SRI International (http://www.ai.sri.com/~oaa) was selected as the communications infrastructure for its combination of open distribution, extensibility, distributed computing capability, and the availability of agents for logic statement and natural language parsing. OAA also allows requests to the network to be open to any agent capable of satisfying the request, or closed and sent only to specific agents.

A diagram of the system is shown in Figure 2. In the center of the diagram is the OAA facilitator, which is the central repository of basic information about each agent and also the router of requests to and responses from the appropriate agents. The operator is presented with an interface that encapsulates an interface agent that communicates with the other agents as necessary. A robot agent represents each robot in the agent community. The Correspondence Agent, explained below, interacts with the interface and the robots through the OAA facilitator.

Interface agent

An interface agent is responsible for constructing the queries that are passed via OAA to the various robot agents according to interface actions by the operator. In the current system, the interface agent automatically sends out periodic requests to the OAA network for lists of objects that are sensed. The agent then keeps track of which robots have responded with which object names. Subsequently, the interface agent determines the task capabilities of the robot by sending a query to the network for a particular robot and object combination. A computer interface encapsulates this interface agent and provides the operator with the necessary functionality to instruct and receive info from the agent.

Computer interface

The interface is implemented using OpenGL to provide a three-dimensional view of the robot environment. The basic robot, object, and environment shapes were predefined during the coding of the interface, but the elements that are displayed are dynamically dependent on which robots are active and which objects they sense. An example screen with the components labeled is shown in Figure 3.

Objects are selected by clicking on them on the screen. The OpenGL 'picking' mechanism is used to determine the object displayed on the top of the graphical object stack under the cursor. The interface then resolves the identity of the object and the robot that sensed the object. If appropriate, the interface agent places a request for task information to the robot.

The interface waits until a response from the correct robot has been returned in the form of a list of tasks that the robot can accomplish on that object. This list is then displayed in a dialog that pops up next to the object. The user can select from this valid list of tasks, and the complete command of robot/task/object is sent to the robot for execution.

Taking turns

Determining who should be speaking in a dialogue – called the 'inference of illocutionary force' by linguists – is a very complex subject. Although humans might find conversations full of explicit declarations of illocutionary force (i.e. May I ask you a question?) unnecessarily cluttered, it dramatically reduces the complexity of a human-robot dialogue. In earlier artificial intelligence research, a dialogue utilizing such an explicit device was conducted [29] to show how to successfully program one robot to talk to another.

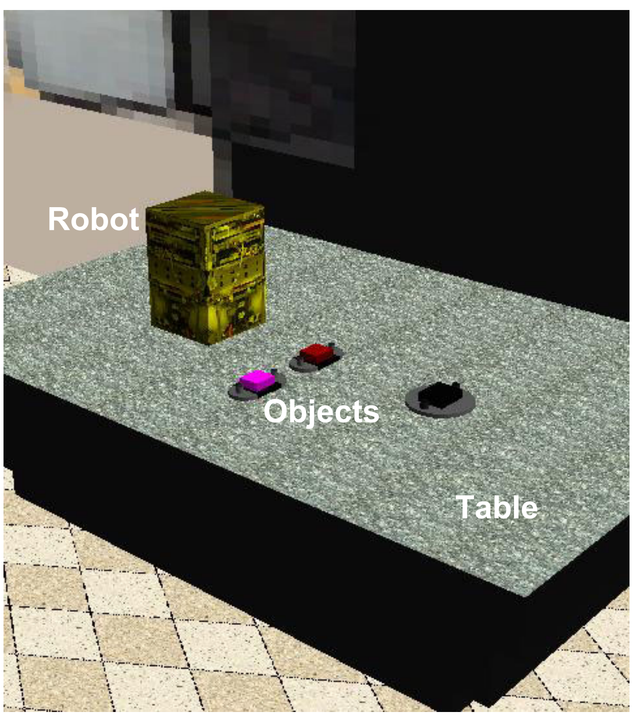


Figure 3 – OpenGL 3-D interface

In our dialogue, the taking of turns is explicit and known by all participants. The dialogue consists of the following steps:

1. All robots that can hear an open request (by being connected to the OAA facilitator) are asked for a list of objects that they sense by the interface agent representing the operator.
2. Robots reply back to the interface agent with their information.
3. The interface agent asks a particular robot for task information regarding a specific object.
4. The robot responds with a list of tasks possible with that object.
5. The interface agent provides a task and object to the robot, completing the dialogue.

Robots only speak once spoken to, and they only expect a limited variety of queries. This is quite sufficient for the purposes of this dialogue, and makes the implementation significantly easier than a full dialogue would be.

Correspondence agent

One of the significant challenges for this architecture, which relies heavily on object perception, is the potential disagreement between object information sensed by more than one robot. This is particularly important to address if the intent of the operator is for multiple robots to perform tasks on one object together or if robots will work independently with similar objects in close proximity to one another.

Since there is no global source of information about the objects, commands must be given in each robot's own context. From a human-computer interaction perspective, this creates an additional challenge to display the objects in a way such that the user does not have to click within each individual context, but can have one click that is then decomposed into the proper context behind the scenes. A Correspondence Agent automatically compiles and distributes this object information.

However, there will often be situations where an automated method for determining object correspondence will not work robustly. In such a case, it is important to give the operator information about object information inconsistencies and allow the user to determine what the correspondence status of each robot is. To handle this possibility, the agent was written to accept either manual or automatic suggestions of correspondence.

Other situations, such as cooperative tasks, highlight a need for a device to determine correspondence. Because this is such a useful tool, the Correspondence Agent answers requests for object context decomposition from any source whenever needed.

We did not address the perception of the objects themselves. This is a significant area of active research, and our plan is to incorporate advances in this field as they become available. For this system, we use a system of LED markers on the objects and a vision system that uses the markers to positively identify and track objects.

6. RESULTS

The experimental platform used was the Free Flying Space Robots (FFSR) in the Stanford University Aerospace Robotics Laboratory [5]. This robotics test bed consists of three self-contained (on-board computer, power, propulsion, wireless communication) air cushion vehicles floating on a polished granite plate, although only one robot has been used with this interaction thus far. This system simulates in two dimensions the drag-free environment of space. An overhead vision system is used for position sensing of the robot and the objects in the environment, but the object info is filtered by perceptive sensor range and distributed to the robots. Each FFSR has arms that have piston-type grippers on the end that enable the robots to grasp specially designed objects that also float on the table. For static objects, small plastic cones are used.

Figure 4 – Example interaction

Figure 5 – Change in robot capabilities

Figure 4 shows the dialogue-based interaction in operation with one FFSR. The process that has preceded this screenshot is the manifestation of the dialogue steps described in the previous section:

1. The interface sent out a request for information.
2. The robot responded with a list of the position and object types of the objects it sensed (including itself).

3.a The operator clicked on the robot to select it. This step could be eliminated since there is only one robot, but is included for completeness since operations with multiple robots would include this step. At this point, the interface keeps the robot in memory as selected, and waits for another selection that makes sense for this context.

3.b The operator clicked on the purple object to select it. The interface agent sends a query to the robot agent to ask what tasks are possible on that object.

4. The robot agent responds with a list of tasks.
5. The interface displays the list of tasks for selection by the user. The user selects a task and it is sent as a command to the robot.

The utility of this method is shown in Figure 5. The operator and the agents used the same process, but the state of the robot has changed. The robot is no longer able to move laterally, but only rotate, making ‘Watch’ the only task it is capable of doing. Consequently, this is the only option displayed to the user. The user is thus only receiving valid affordances from the robot regarding what operations are possible.

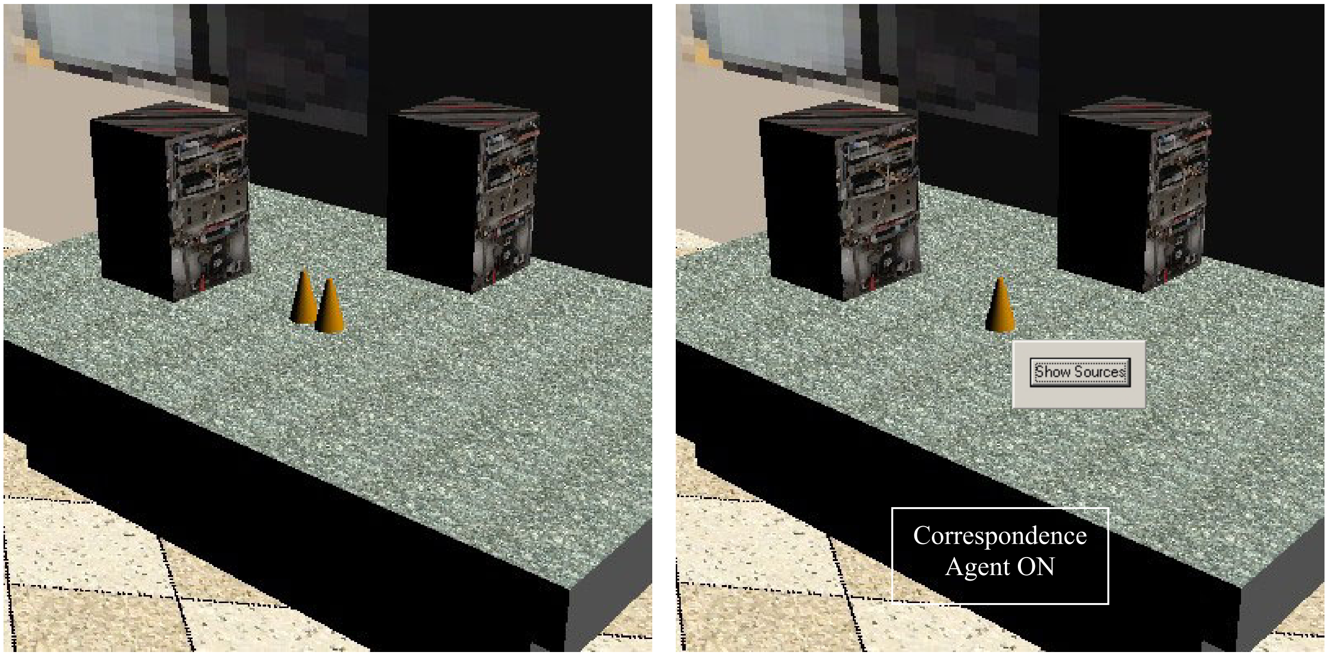


Figure 6 - Correspondence Agent in use

The impact of the Correspondence Agent is shown in Figure 6. Two simulated robots are observing a cone on the table, but the perceived locations are not consistent. Consequently, two separate cones appear on the table in the left pane when the Correspondence Agent is turned off.

The right pane shows the effect of turning on the Correspondence Agent. In this instance, the Agent was given a rule that only one cone exists, so it automatically associates the cones sensed by the two robots, and only submits a single cone for display.

Basically, the Correspondence Agent takes over ‘ownership’ of the cone object, and suppresses the display of the cones sensed by the robots. The interface can be instructed to display the raw locations of the object by clicking on the object and selecting ‘Show Sources’ from the resulting dialogue box, the step shown in the right pane of Figure 6.

7. Summary

This research has shown that it is possible to build a dialogue-based interaction that enables the control of multiple robots. This interaction, as implemented in a virtual three-dimensional world, provides an intuitive point-and-click method for determining the capabilities of the robot in the appropriate context, and enabling the operator to participate in the resource management and task planning for the robots.

References

[1] H. Jones, E. Frew, B. Woodley, S. Rock, “Human-Robot Interaction for Field Operation of an Autonomous Helicopter,” *Mobile Robots XIII*, 1998.

[2] M. Adams, S. Kolitz, S. Rasmussen, “An Automation-Centered Human-System-Integration Architecture for Autonomous Vehicles,” *AUVSI '98*, 1998.

[3] S. Fleischer, S. Rock, J. Lee, “Underwater vehicle control from a Virtual Environment Interface,” *Symposium on Interactive 3D Graphics*, 1995.

[4] H. Wang, *Experiments in Intervention Autonomous Underwater Vehicles*, PhD thesis, Stanford University, 1996.

[5] H. Stevens, E. Miles, S. Rock, R. Cannon, “Object-Based Task-Level Control: A hierarchical control architecture for remote operation of space robots,” *AIAA/NASA Conference on Intelligent Robotics in Field, Factory, Service, and Space*, 1994.

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

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

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

[9] R. Gilson, C. Richardson, M. Mouloua, "Key Human Factors Issues for UAV/UCAV Mission Success," *AUVSI '98*, 1998.

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

[11] L. Parker, "Multi-Robot Team Design for Real-World Applications," *Distributed Autonomous Robotic Systems 2*, 91-102, Tokyo: Springer-Verlag, 1996.

[12] H. Surmann, M. Theissinger, "ROBODIS: A dispatching system for multiple autonomous service robots," *Field and Service Robotics '99*, 1999.

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

[14] T. Payne, K. Sycara, M. Lewis, "Varying the User Interaction within Multi-Agent Systems," *4th International Conference on Autonomous Agents*, 2000.

[15] R. Arkin, T. Collins, Y. Endo, "Tactical Mobile Robot Mission Specification and Execution," *Mobile Robots XIV*, 1999.

[16] T. Fong, C. Thorpe, C. Baur, "Collaboration, Dialogue, and Human-Robot Interaction," *International Symposium of Robotics Research*, 2001.

[17] D. Brock, D. Montana, A. Ceranowicz, "Coordination and Control of Multiple Autonomous Vehicles," *IEEE Conference on Robotics and Automation*, 1992.

[18] C. Miller, M. Pelican, R. Goldman, "'Tasking' interfaces to keep the operator in control," *5th Annual Symposium on Human Interaction with Complex Systems*, 2000.

[19] "MAGIC2: Multiple Aircraft GPS Integrated Command and Control System," *AUVSI '98*, 1998.

[20] R. Aylett, A. Coddington, D. Barnes, R. Ghanea-Hercock, "Supervising multiple cooperating mobile robots," *Autonomous Robots 97*, 1997.

[21] K. Ali, *Multiagent Telerobotics: Matching Systems to Tasks*, PhD thesis, Georgia Institute of Technology, 1999.

[22] S. Iacono, S. Weisband, "Developing Trust in Virtual Teams," *30th Annual Hawaii Int'l Conf on System Sciences*, 1997.

[23] B. Tuckman, "Developmental sequence in small groups," *Psychological Bulletin #63*, 384-399, 1965.

[24] B. Muir, N. Moray, "Trust in Automation," *Ergonomics* **39** (3), 429-460, 1996.

[25] J. Lee, N. Moray, "Trust, control strategies and allocation of function in human-machine systems," *Ergonomics* **35** (10), 1243-1270, 1992.

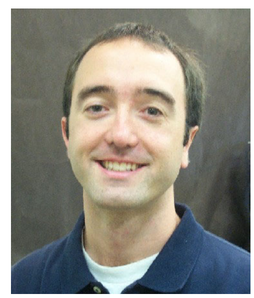
[26] C. Bowers, R. Oser, E. Salas, J. Cannon-Bowers, "Team Performance in Automated Systems," in *Automation and Human Performance: Theory and Applications*, edited by R. Parasuraman, M. Mouloua, 243-261, 1996.

[27] *IEEE Standard Dictionary*, 1990.

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

[29] R. Power, *A computer model of conversation*, PhD thesis, University of Edinburgh, 1973.

Hank Jones *is a graduate researcher in the Aerospace Robotics Laboratory at Stanford University, where he has performed varied research on the role of the human in the deployment and operation of field robots. He received a B.S in Mechanical Engineering from the University of Mississippi in 1995 and an M.S. in Aeronautics and Astronautics from Stanford University in 1996. His research interests are in the areas of human-system interaction design and the application of teamwork research to robotics.*



Stephen Rock *received his S.B. and S.M. degrees in Mechanical Engineering from MIT in 1972. He received his Ph.D. in Applied Mechanics from Stanford University in 1978, and joined the faculty of the Aeronautics and Astronautics department of Stanford in 1988. He is the Director of the Aerospace Robotics Laboratory where his research focus is to extend the state-of-the-art in robotic vehicle control. His interests include the application of advanced control techniques for robotics and vehicle systems. Areas of emphasis include remotely operated vehicles for both space and underwater applications. Dr. Rock teaches several courses in dynamics and control at Stanford. Prior to joining the Stanford faculty, Dr. Rock managed the Controls and Instrumentation Department of Systems Control Technology, Inc.*