USER INTERACTION WITH MULTI-ROBOT SYSTEMS

David Kortenkamp, Debra Schreckenghost and Cheryl Martin
NASA Johnson Space Center – ER2
Metrica Inc./TRACLabs
Houston TX 77058
kortenkamp@jsc.nasa.gov, ghost@ieee.org, cheryl.martin@jsc.nasa.gov

Abstract There has been very little research on multiple human users interacting with multiple autonomous robots. In this paper we present some of the requirements of such user interaction. We present a prototype architecture for collaborative interaction. This architecture is put into the context of multiple space robots monitoring a space structure to assist human crew members.

Keywords: Multi-robot systems, human-robot interaction

1. Introduction

Historically, human-robot interaction has focused on single humans interacting with single robots. For many deployed robot systems, there are multiple humans interacting with a single robot. Current research in multi-robot systems has often focused on robot-robot interaction and very rarely has it focused on human to multi-robot interaction. Even more rare is research focused on how multiple, cooperating humans interact with multiple, cooperating robots. In this paper we look at some of the key research issues in user interaction with multi-robot systems. An early architecture for handling these interactions is presented.

To provide context to the issues raised in this paper, we will use examples drawn from a specific class of NASA robots: mobile monitors. These are free-flying robots that have no manipulation capabilities. They can be used to perform inspection, monitoring and sensing tasks. Two examples of mobile monitors are currently under development at NASA. The first is the Personal Satellite Assistant (PSA), which is designed to be used inside of a space vehicle. The second is the Autonomous EVA

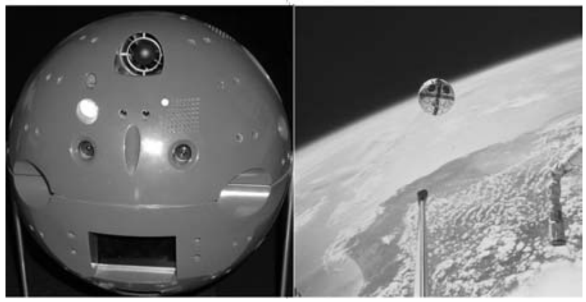

Figure 1. Left: The Personal Satellite Assistant (PSA) robot. Right: The AERCam robot during a teleoperated space mission.

Robotic Camera (AERCam), which is designed to be used outside of a space vehicle (see Figure 1). We will work from a scenario in which there is one PSA, one AERCam and two crew members cooperating to perform a diagnosis task.

Our discussion will be based on a distributed interaction architecture currently being developed at NASA JSC. This architecture is meant to allow NASA crew members to interact with a wide range of autonomous systems (robots being just a subset). The architecture is still being defined and implemented, so this document simply reflects our initial direction. We will first present the distributed interaction architecture and then identify specific issues for robotic interaction.

2. Concepts for Human-Robot Interaction

We have defined an operations concept for human-robot interaction that describes how humans and robots "should" be able to work together and identifies the challenges in achieving such interaction. We identify three main research areas: (1) monitoring and control of mostly autonomous robots; (2) managing the tasks of multiple, distributed agents (crew and robots); and (3) aiding distributed agents in exchanging information while operating remotely and asynchronously. Each of these research areas is described in this section.

2.1 Adjustable Autonomy

Our concept of robot operations includes robots that operate autonomously most of the time. Thus, an important user activity is maintaining situational awareness of autonomous robot activities and their effects. This supervisory task requires occasional, remote monitoring of the robots and user notification when interesting or unusual events occur or a need for manual action arises. Data summarization and visualization techniques are needed to support status-at-a-glance on the control situation across multiple, distributed robots. When situations arise that fully autonomous robots cannot address, it is necessary to support some level of user interaction and intervention with the robot. Our policy for such intervention is to provide for the interactive adjustment of autonomy Kortenkamp et al., 2000; Barber et al., 2000. Techniques for such adjustment include reallocating autonomous tasks to the user, temporarily modifying autonomous procedures for unique situations, and overriding autonomous actions. Interactive adjustment of autonomy may also be needed when the user and robots work closely together to accomplish a shared mission objective. A key issue is determining whether/when/how to interrupt the robot.

2.2 Distributed Task Management

The coordination of user and robot activities will be a key component of the architecture. Coordination requires the capability to automatically track the activities of the users and the robots, to synchronize schedules when distributed robots and users share a common task objective, and to remind the users of pending tasks and task deadlines. When contingency situations with the robots arise, the user needs support in handling interruptions when they respond to the contingency. This includes assistance in managing multiple ongoing tasks, notification of task deadlines, and assistance in updating the activity plan.

2.3 Distributed User Communication

Distributing users and robots throughout a facility (such as space station) or outside the facility for extra vehicular activities (EVA) isolates users from indirect communication resulting from close proximity to the robots (e.g., they may not be able to observe an action) and can limit the information bandwidth available. Supporting operations where users and robots are occasionally out of communication further constrains the ability of each to maintain awareness of relevant operational changes. Finally, the increased use of autonomous robots takes the user out of the control loop, which can result in less awareness of on-going robot actions. In such an environment, the user needs tools for effective information exchange in support of remote, asynchronous operations.

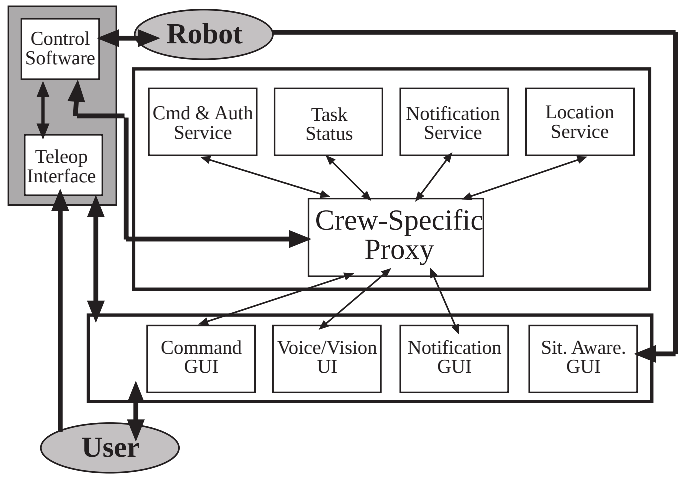

Figure 2. The components comprising our distributed interaction architecture. This figure shows one user and one robot. Each additional user would have their own proxy, interface and services. Each additional robot would have its own control software and teleoperation interface.

2.4 Robot Interaction Issues

The distributed interaction architecture applies equally to user interaction with robotic agents and with non-robotic software agents. In this section we look at a few issues that distinguish between "physical" agents (robots) and software agents.

- Teleoperation: There will need to be facilities for the user to take direct control of the robot, perform some task, and then turn control back over to the autonomous controller.

- Multi-modal input: The physical embodiment of the robot opens up new interaction possibilities, such as drawing a path to be followed on a graphical tablet. Also gestures can provide critical information.

- Physical interaction: Robots can interact physically with the user and the environment. These interactions require extra consideration for coordinating activities, possibly including models of physical processes.

Our initial implementation of the architecture has not addressed these robot interaction issues, but we expect to do so in the future.

3. Components of the Architecture

We have developed a distributed interaction architecture design based on the operations concepts in the previous section and and a literature review (see Section 5). Components of this architecture that assist the user in robot interaction include a crew proxy for each user. A user interface mediates the interaction with these components. Each crew proxy utilizes the following *collaboration services*:

- A task status service that provides activity tracking and plan management capabilities for use by both the user and the robots.
- A location service that uses a variety of technologies to determine whether the user is online and to locate both user and robot relative to important features in environment (e.g., near the tank).
- A notification service that determines which user to notify of an event and how.
- A command and authorization service that determines which user has the authority to command a particular robot.

The services are self-explanatory, so we will concentrate on some other central components of the architecture next – the crew proxy and the user interface. Figure 2 shows all of these components.

3.1 Crew Proxy

Central to our approach for assisting the user in interacting with autonomous robots is providing crew proxies who represent and take action for the user. Each user has a proxy to represent his or her interests and concerns. From one perspective, this proxy stands in for the human agent in the interaction architecture. As a stand-in, it can interpret information from the robot or other agents, respond to requests from other agents and make decisions without bothering the human. From another perspective, this proxy augments human capabilities by enhancing human perception and actuation. As a capability augmentation, it can provide improved insight and enhanced ability to respond to robot situations.

The proxy provides functionality for a single user by coordinating collaboration services based on logical dependencies among these services. All user utilization of these services is mediated by the proxy software. The proxy also provides uniform access to information about its user that supports collaboration with other agents (robots or other users) in the architecture. It utilizes standardized knowledge models to delineate and represent collaborative information. Although our initial architecture only develops proxies for the crew, we believe that the proxy concept can be useful to a robot as well. This extension to the architecture permits crew proxies and robot proxies to collaborate without adversely affecting robot control.

3.2 User Interface

The bottom layer in Figure 2 contains components of a user interface. These components mediate the interaction between the user and the proxy, as well as providing a situation awareness interface that displays robot parameters (e.g., location, fuel, health, etc.). Teleoperation of the robot is achieved through a more direct connection with the robot control software, because there may be robot-specific requirements. Our goal is to provide a customizable interface for human-robot interaction.

4. An Example Scenario

An example scenario is taken from a space domain and has two mobile monitors – one external to the spacecraft and one internal to the spacecraft and two crew members (users). In the scenario there is also an autonomous life support system. At the start of the scenario the two mobile monitors are engaged in routine inspection activities, with the status of those activities being monitored by the crew members. Event detection software for the life support system detects a possible gas leak. The crew members are alerted through their proxies in the way most appropriate for their current location and activity. While they work to reconfigure the system to bypass the leak, they dispatch (via their task planners) the mobile monitors to try to detect the leak – from both the inside and outside. The mobile monitors interface to the life support autonomous control system to narrow down their search. They both converge on the leak and notify the crew via the notification service and the proxies. A crew member uses its proxy (which ensures correct authorization) to take command (via teleoperation) of each mobile monitor in turn and fly in for closer looks. After rerouting gas around the leaks, the crew turns control of the mobile monitors back to their autonomous control systems and asks that they continue to monitor the leak site.

5. Literature Review

Very little previous research has focused explicitly on multi-user interaction with multiple robotic agents. However, we can apply lessons from existing research on coordination and distributed collaboration among humans and software agents. We examined a number of implemented systems that helped inform the initial architecture design.

System characteristics, algorithms and interaction models that support different types of basic collaboration capabilities have been developed by previous research addressing human-agent interaction and mixed-initiative planning (i.e. COLLAGEN Rich and Sidner, 1998 and TRIPS Ferguson and Allen, 1998) as well as overall coordination in distributed multi-agent systems Jennings, 1996; Lesser, 1998. In particular, the proxy model of interaction between humans and software agents has been successfully demonstrated by the Electric Elves system Chalupsky et al., 2001. In this system, proxy agents for each person in an organization perform organizational tasks for their users (e.g., monitoring the location of each user, keeping other users in the organization informed, and rescheduling meetings if one or more users is absent or unable to arrive on time). Previous research has also addressed other collaboration needs including the development of "advisable" agents that incorporate users' preferences about when to ask for permission or consultation for given behaviors Myers and Morley, 2001.

Other previous research has developed integration infrastructure for multi-agent systems: CoABS Grid (http://coabs.globalinfotek.com/), COUGAAR (http://www.cougaar.org), JADE Bellifemine et al., 1999, KAoS Bradshaw et al., 1997, RETSINA Sycara et al., 2002, and the Open Agent Architecture (http://www.ai.sri.com/ oaa/whitepaper.html).

Knowledge representations and knowledge models supporting collaboration are critical to implementing a distributed interaction architecture. Various knowledge models have been employed by previous research to support collaboration and coordination. These models include task and activity models Clancey et al., 1998, team and shared plan models Kumar et al., 2000, resource and capability models Chalupsky et al., 2001, user preference models Myers and Morley, 2001, and roles, authority and organizational models Bradshaw et al., 1997. These knowledge models and their representations together with the infrastructure support provided by multi-agent development platforms and distributed computing technologies provide a foundation for the implementation of the architecture. By leveraging this previous work concerning software agents, we can make rapid progress toward building a system to support multi-user/multi-robot interaction.

Acknowledgments

This work is supported by several NASA research grants. Including a grant from the NASA Intelligent Systems Program, Human Centered

Computing and a grant from the NASA Engineering Complex Systems program. Discussions with Carroll Thronesbery (NASA JSC/SKE Incorporated), Pete Bonasso (NASA JSC/Metrica Inc.) and Tod Milam (NASA JSC/Metrica Inc.) contributed to the architecture.

References

Barber, S., Goel, A., and Martin, C. (2000). Dynamic adaptive autonomy in multi-agent systems. *Journal of Experimental and Theoretical Artificial Intelligence*, 12(2).

Bellifemine, F., Poggi, A., and Rimassa, G. (1999). JADE - a FIPA-Compliant agent framework. In *Proceedings of the Fourth International Conference and Exhibition on the Practical Application of Intelligent Agents and Multi-Agents (PAAM'99)*.

Bradshaw, J. M., Dutfield, S., Benoit, P., and Woolley, J. (1997). KAoS: Toward an industrial-strength generic agent architecture. In Bradshaw, J. M., editor, *Software Agents*. AAAI/MIT Press, Cambridge MA.

Chalupsky, H., Gil, Y., Knoblock, C. A., Lerman, K., Oh, J., Pyandath, D. V., Russ, T. A., and Tambe, M. (2001). Electric elves: Applying agent technology to support human organizations. In *Proceedings of the Innovative Applications of Artificial Intelligence*.

Clancey, W. J., Sachs, P., Sierhuis, M., and van Hoof, R. (1998). Brahms: Simulating practice for work systems design. *International Journal of Human-Computer Studies*, 49.

Ferguson, G. and Allen, J. (1998). TRIPS: an integrated intelligent problem-solving assistant. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*.

Jennings, N. R. (1996). Coordination techniques for distributed artificial intelligence. In O'Hare, G. M. P. and Jennings, N. R., editors, *Foundations of Distributed Artificial Intelligence, Sixth-Generation Computer Technology Series,*. John Wiley and Sons, New York.

Kortenkamp, D., Keirn-Schreckenghost, D., and Bonasso, R. P. (2000). Adjustable control autonomy for manned space flight. In *IEEE Aerospace Conference*.

Kumar, S., Cohen, P. R., and Levesque, H. J. (2000). The adaptive agent architecture: Achieving fault-tolerance using persistent broker teams. In *Proceedings of the International Conference on Multi-Agent Systems*.

Lesser, V. R. (1998). Reflections on the nature of multi-agent coordination and its implications for an agent architecture. In *Autonomous Agents and Multi-Agent Systems (AAMAS-98)*.

Myers, K. L. and Morley, D. N. (2001). Directing agent communities: An initial framework. In *Proceedings of the IJCAI-2001 Workshop on Autonomy, Delegation, and Control: Interacting with Autonomous Agents*.

Rich, C. and Sidner, C. L. (1998). COLLAGEN: a collaboration manager for software interface agents. *User Modeling and User-Adapted Interaction*, 8(3-4).

Sycara, K., Paolucci, M., van Velsen, M., and Giampapa, J. (2002). The RETSINA MAS infrastructure. In *Autonomous Agents and Multi-Agent Systems*.