Distributed Operations among Human-Agent Teams for Testing Crew Water Recovery Systems

D. Schreckenghost, C. Martin, P. Bonasso, M. B. Hudson, T. Milam, and C. Thronesbery

NASA Johnson Space Center, 2101 Nasa Road One, Houston, TX, 77058
ghost@ieee.org, cmartin@traclabs.com

Abstract

Control automation can reduce human workload by automating routine operations such as system reconfiguration and anomaly handling. Humans need assistance in interacting with these automated control agents. We have developed the Distributed Collaboration and Interaction (DCI) system, an environment in which humans and automated control agents work together. Within this environment, we provide personal liaison agents, called Ariel agents, for each human in a group to assist them in performing the duties associated with the roles they perform for that group. We have deployed the DCI system at NASA's Johnson Space Center (JSC) to assist control engineers in performing duties associated with crew water recovery systems. In this paper we describe the water recovery system in the Water Research Facility (WRF) at JSC, the DCI system we developed, and our experiences in deploying DCI for use in the WRF.

Introduction

In manned space operations, control automation can reduce human workload by automating routine operations such as system reconfiguration and anomaly handling. The use of such automation in space operations requires that humans be cognizant of automation performance and be available to handle problems outside the capabilities of the automation. These human tasks are infrequent, typically require little time, and often can be performed remote from the system being controlled. To accomplish these tasks, humans need support for distributed control operations with automated control agents.

We have developed the Distributed Collaboration and Interaction (DCI) system, an environment in which humans and automated control agents work together. Within this environment, we provide personal liaison agents, called Ariel agents, for each human in a group to assist them in performing the duties associated with the roles they perform for that group (Schreckenghost, et al., 2002). We have deployed the DCI system at NASA's Johnson Space Center (JSC) to assist control engineers in performing duties associated with crew water recovery systems. In this paper we describe the water recovery system in the Water Research Facility (WRF) at JSC, the DCI system we developed, and our experiences in deploying DCI for use in the WRF.

Water Research Facility at JSC

The Crew and Thermal Systems Division (CTSD) performs ground-based testing of regenerative water recovery systems in the Water Research Facility at JSC. It consists of a waste water collection facility, an analysis laboratory, and a post processing hardware subsystem. Currently the Post Processing System (PPS) is being evaluated in extensive ground testing. The PPS is a water "polisher" used to bring relatively clean water to within potable limits. It removes the trace inorganic wastes and ammonium in recycled water using a series of ion exchange beds and removes the trace organic carbons using a series of ultraviolet lamps.

The PPS is controlled by the 3T automated control software (Bonasso, et al., 1997). The 3T architecture consists of three tiers of parallel control processing:

- Deliberative Planner. hierarchical task net planner to manage activities with resources or temporal constraints, or requiring multi-agent coordination,
- Reactive Sequencer. reactive planner to encode operational procedures that can be dynamically constructed based on situational context, and
- Skill Manager. layer for closed loop controllers.

This approach is designed to handle the uncertainty inherent in complex domains. Control commands flow down through the hierarchy and feedback flows back up through the hierarchy to close the control loop. If a command fails at any level, it can initiate a repair action (e.g., replanning at the deliberative level, selection of an alternative sequence at the reactive level). Each layer operates at a different time constant, allowing high speed controllers at the lower level of the architecture to operate in parallel with the slower, deliberative algorithms at the high level. The 3T architecture has been used extensively during ground tests of life support systems for regenerative water recovery (Bonasso, et al, 2003) and air revitalization (Schreckenghost, et al, 1998).

DCI System

The DCI System consists of an *Ariel agent* for each human in the group, an *Activity Planner* for planning and coordinating human activities, an *Event Detection Assistant* (EDA) for detecting simple domain events, a *Complex Event Recognition Architecture* (CERA) for capturing complex domain events, and the DCI infrastructure required for these components to communicate using CORBA. Each Ariel agent provides services to assist its user in performing the duties associated with his or her job. The DCI System has been implemented using Java 1.4.2, Allegro Lisp 6.1, and CORBA ORB implementations JacORB 1.4.1 and ILU 2.0b1. In this section we describe the services and capabilities implemented for the control engineers in the WRF.

Notification and Alerting. The Ariel Notification Service encodes communication protocols to notify distributed team members when important events or operations occur. Protocols must be adapted to account for the fact that an individual may hold multiple roles, that roles can be reassigned dynamically, and that changes in role alter the information requirements of the human. Notices of user state changes (e.g., location changes) are exchanged among the liaison agents to support group coordination. The EDA detects simple events (e.g., thresholds, triggers) and publishes them to the liaison agents. CERA from I/Net (Fitzgerald, et al., 2003) captures complex event sequences associated with operational situations and anomalies. CERA applies language recognition principles to detect event patterns with complex temporal and hierarchical relationships among them. These event sequences also are published to the Ariel agents as situation summaries. The Notification Service within the Ariel agent is responsible to filter all events received by the agent. The User Interface Service then determines how to present notices of interest to the user. These services use protocols encoded as rules describing the conditions for identifying the notices of interest to the agent's user and directives for how to present the notice to the user. These rules include the use of domain ontologies to describe conditions and directives. The user attention is shifted to important notices by changing the notices icon in the DCI toolbar (Figure 1). To view the new notices, the user selects the notice icon in the toolbar and the notice viewer will be displayed (Figure 2). Notices can also be sent via a pager or via email.

Figure 1. DCI Toolbar

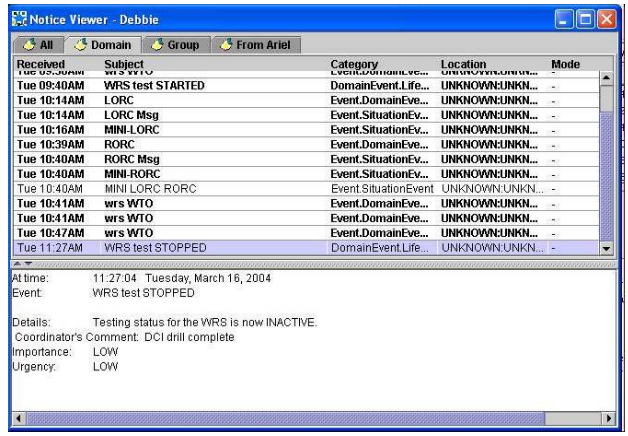

Figure 2. DCI Notice Viewer

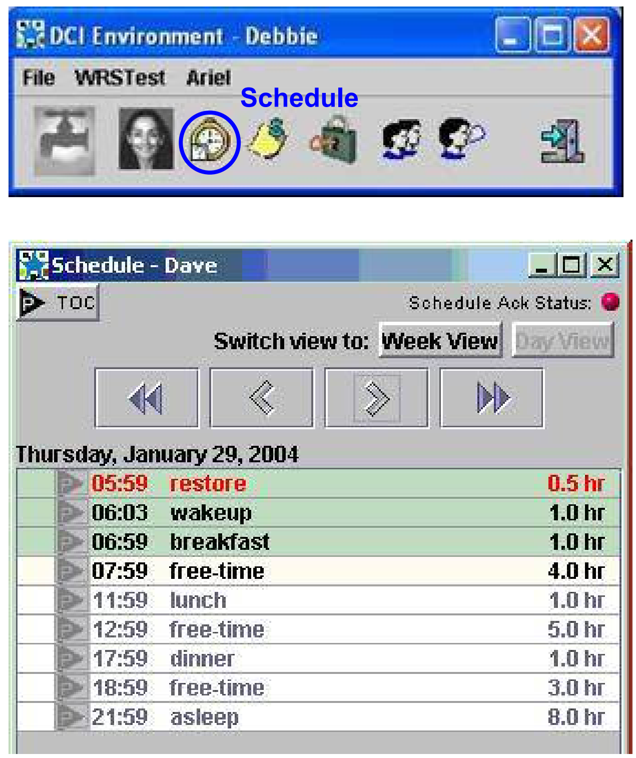

Figure 3. DCI Schedule

Activity Coordination using Shared Plans. We use a centralized plan built by a hierarchical task net planner to coordinate the actions of humans performing space operations. We chose to use a centralized planning approach to ensure activities are assigned to accomplish all operational goals, to help prevent humans from issuing conflicting commands, and to coordinate the handover between manual and automated tasking. The Activity Planner assigns roles, builds the activity plan, and tracks progress in completing the plan. When activities are not completed successfully, the Activity Planner builds another plan to accomplish the operational goals, including reassigning tasks. The Task Status Service for each Ariel agent determines its user's progress in completing the assigned activities, informs the Activity Planner about task progress, and notifies its user when her schedule changes. The Ariel agent determines task completion either using direct evidence obtained through computer-mediated manual tasks or indirect evidence based on the user's location from the Location Service. In the WRF, the planner is used primarily to manage anomaly handling activities. As shown in Figure 3 above, the schedule of the person responsible to fix the problem is updated automatically with the anomaly handling task ("restore" task in red text).

Figure 3. DCI Schedule

Location Tracking. The Location Service of the Ariel agent determines its user's location information for use in tracking the completion status of crew activities, in determining how to notify the crew of events, and in maintaining awareness of a distributed group via a group awareness display. This service interprets sensed user location from computer login/logout events to determine the user's physical location and whether the user is online or offline. Computer locations are mapped to the closest physical location milestone (e.g., lab, office) using a location ontology. The user is considered online when a login event is received. The user is considered offline when a logout event is received indicating the user has no active session, or when the last active session times out. The user's physical and cyber-location are combined to assess availability and accessibility (i.e., user presence). A group awareness display can be access from the DCI toolbar that provides a view of the location, role, and schedule of all users in the operational group (Figure 4 above). In this example, Carroll was recently logged into his Ariel system in his Building 32 office. As indicated by his grayed out picture, however, he is no longer "online" with his agent.

Related Work

The Ariel agents of the DCI system resemble the Friday agents in the Electric Elves project (Scerri, et al., 2002) in that both types of agents aid humans interacting in groups. The Friday agent were intended to assist professors and students at the University of Southern California in conducting their daily business. These Friday agents scheduled meetings, arranged for speakers at those meetings, scheduled rooms for the meetings, and made reservations for luncheons. The Ariel agents differ from the Friday agents, however, in that the Ariel agents aid interaction among humans and automated control agents while the Friday agents only aid interaction among humans.

Within NASA, the Mobile Agent Project (Sierhuis, et al., 2003) has developed both proxy agents and personal agents for use in a Mars analog project. In this project, the mobile agents assist humans in collecting and organizing geological and biological data during a planetary traverse. As with the Friday agents from the Electric Elves Project, a fundamental difference between the Mobile Agents and the Ariel agents is that the Ariel Agents are designed to assist interaction among teams of humans and automated control agents, while Mobile Agents are intended to aid human interaction.

Also within NASA, the Personal Satellite Assistant (PSA; Bradshaw, et al., 2000), is a robotic device designed to assist astronauts with communications and remote monitoring. The PSA has been simulated in a testbed at the Ames Research Center. Like the Ariel agent, the PSA is intended to free the astronauts from routine tasks and increase the level of crew autonomy. The focus of the PSA, however, is on technology for embodied agents while the focus of the Ariel agent is on software agents.

Deployment of DCI in the Water Research Facility

In this section we discuss our experiences in deploying the DCI system for the control engineers in the WRF. We describe how the operational protocols have been changed by using the DCI system, how the DCI system supports distributing operations, how DCI has evolved to provide more stable, efficient operations, and an example of how we have evaluated the DCI system.

Operational Protocols

The control engineers in the water lab fulfill three roles: Prime, Backup, and Coordinator. The *Prime* is the person responsible to handle anomalies that cannot be handled by the control automation. This requires going to the water lab where both the control software and the life support equipment reside. The *Backup* is responsible to handle problems when the Prime cannot. The *Coordinator* is the most knowledgeable about the PPS system and is responsible to ensure the overall health of the PPS. Each week the roles of Prime and Backup shift among the control engineers. The role of Coordinator, however, is assigned to a single person and does not change.

For previous water tests, the Prime would bring up data monitoring displays when in his office to watch for problems. When out of the office, he would try to check these displays every four hours throughout the day. He was on-call overnight, if someone noticed a problem. But there was no automatic way of detecting and notifying him of a problem. When a problem occurred, the phone was used to coordinate group members.

The availability of the DCI system has changed these protocols. Now the Prime brings up his DCI system when in the office. His Ariel agent monitors for events and notifies him when something important happens. The agent also captures the important situation summaries for his inspection after the fact. If the person is offline, his agent will page him when an important event occurs. His agent can assign him tasks when problems occur, and does this with knowledge of his availability based on his schedule. If he cannot respond in time, the DCI Activity Planner will reassign the responsibility to handle the situation to the Backup automatically and will notify both the Backup and Prime of the task reassignment. The phone is still used for coordination in cases where a group member cannot access his DCI system.

Once we deployed DCI in the WRF, we revised the new protocols in the following ways:

- DCI Shutdown: we added notices sent to all users of the DCI system to let them know when the DCI system is about to be shutdown and restarted. This ensured that users who were interacting with their agents were given warning that the system was about to become temporarily unavailable.
- Notify critical anomalies on multiple channels, such as email and paging, since we are not sure which are accessible to the person when offline.
- Provide both computer-based and other ways to confirm or deny acceptance of tasks assigned for critical anomaly handling (e.g., reply via pager).
- Do not eliminate existing coordination techniques that work well, such as the telephone. For example, we saw an over-reliance on automatic paging which prevented directly calling engineers.

Changes for Distributed Operations

The DCI system improves the ability to support distributed control operations, permitting remote access to the PPS from offsite offices and home. In this section we discuss the changes we made to the DCI system during deployment to better support distributed operations.

Notification. As the operations group is distributed and potentially unavailable at times, it becomes important to support *asynchronous notification*. For our group, we use pagers for high saliency, asynchronous notification. We initially tried to use a single pager for the entire group, which was passed with the assignment of the Prime role. Our first PPS anomaly (Loss of Communication) emphasized the need for pagers for all roles. During that anomaly, the battery failed in the single pager we had. The anomaly occurred around 5:30 AM. Had the Backup had a pager, he would have received a notice well before going to work. With only a single pager, however, the notification was not received by the Prime or the Backup until they got to work and logged into the DCI system.

Activity Planning. Another aspect of distributed operations is building plans for *remotely assigned tasks*. As mentioned above, the DCI system manages human tasks using an automated planner. During the course of using the DCI system we learned that a significant portion of the task duration is travel time to the water lab (up to 50% in some cases). We also learned that when notifying a person about a new task assignment it was helpful to clearly identify the task name along with some background information about why the task is needed (e.g., the anomaly to be handled or an indication that Prime did not respond). In anomaly cases, the replanning that results in this new task assignment happens automatically because the assigned person does not respond in a timely manner. In some cases, however, the Coordinator may need to manually activate replanning from a remote location. During deployment we developed a simple *user interface to remotely activate replanning*. From this interface the Coordinator can either create a new plan (invoke a reset and re-plan) or update an existing plan by performing a re-plan. This was primarily used to build a new plan when user roles were changed each week.

Location Tracking. The DCI system tracks the location of group members to aid in group awareness as well as to determine whether members are online or not for notification. For coordination in the water lab, we provide a *group awareness display* that shows the locations and activities of all control engineers. The location tracking is based on where the engineer last logged into the DCI system. The last login is mapped to a physical location ontology. Physical location is combined with whether the user is logged in or not to determine user presence. During deployment we *simplified the user presence ontology* to represent three concepts: (1) online (2) local, and (3) remote. This representation is readily understandable by users and easy to use by other services. The DCI system permits the user to log into her Ariel agent from more than one location. The location tracking system registers her location to be at the last location she logged into. During deployment, however, we discovered that when she logged out, the user was relocated erroneously to her previous login. We have since revised DCI so a user is logged out from all DCI interfaces when she logs out of any interface.

Changes for Stable, Efficient Operations

Prior to deployment in the WRF, the DCI system had only operated continuously for periods of hours at a time. Our targeted operating time for the water lab is 24/7 support for at least weeks, and possibly months at a time. To date, we have operated up to a week continuously. We have had PPS tests ranging from 8 hours to 72 hours. Each test evaluates a different configuration of the PPS hardware (e.g., mix of resins in the beds filtering the water). In this section we discuss changes we have made that address system stability as well as efficient extended operation.

Configuration. Most of the DCI system is encoded using Java. Prior to deployment, we distributed the system across multiple computing platforms in multiple virtual machines to load balance and to simplify debugging. For deployment in the water lab, we had limited computing resources. We restructured the DCI system to run *most Java processes in a single virtual machine* on a computing platform (the Lisp portions of the system ran in their own process as well). This was much more resource efficient, since we no longer had multiple virtual machines on a computer. It required, however, rewriting software to ensure that class members that were not intended to be shared across different instances of Ariel agents were not implemented as static class members. We also revised the DCI system to permit reconfiguring (i.e., shutting down and restarting) portions of the system without completely taking down the entire system. *Reconfiguring the agents* requires that each service within the agent supports startup and shutdown state transitions. It also requires that a user interface be provided for controlling these state changes. Finally, to ensure coordinated communication via the same time stamp, we have a *network time server providing system time* for the machines used by the DCI system and the control system

Cleanup. When testing the capability to shutdown and restart services within an Ariel agent (i.e., reconfiguring the agent), we discovered that some objects were not being cleanly removed from memory when the associated service was shutdown. This resulted in spurious service behavior (e.g., old services seem to "reappear"). We revised the agent reconfiguration to ensure that objects are removed from memory when the associated service is shutdown. We also discovered that a failure of our data server from the control automation would take down all processes running in the Java virtual machine where this interface was running. We revised the interface to the control automation to handle these server failures as an exception that does not take down other processes running in the same virtual machine.

Logging. In preparation for deployment, we added the ability to log all events exchanged within the DCI environment. These events are logged by a process that listens to the all the existing event distribution channels in the system. We log these events to a new file each day, archive these files in a common directory, and backup these directories to tape daily. We developed an event log viewer that loads these files for viewing and permits filtering events by date and time. We also log parameter data for display when anomalies occur. To avoid data files getting large, we have implemented a strategy whereby we reduce the logging rate when circumstances are quiescent.

Buffer Size. When operating for hours at a time, one must consider constraining the size of buffers. For the DCI system, we have found it necessary to constrain the growth of the agent process size by limiting the number of notices that are buffered for viewing in an Ariel agent's display. We also limit the history of user state changes kept for review in the group awareness display in the Ariel agent. In both of these cases, this information can be reviewed if needed by loading the logged event files from previous days.

Workaround versus Fix. For operational systems, it is important to minimize the downtime of the system. When possible, we fix problems as soon as they occur. In some cases, however, implementing a fix can take awhile and it is important to keep the system operational while the fix is being implemented. One strategy for reducing the downtime in this case is the *workaround*. A workaround is a code change or an operational change that minimizes the effect of the problem while a fix is being worked out. For example, we have had a persistent planning problem that causes duplicate tasks to be placed temporarily in our schedules. This problem only occurs at the plan update on the day after a loss of communication anomaly. And it will "fix" itself at a subsequent plan update. We have implemented a procedural workaround where the Coordinator activates a manual plan update the morning after a loss of communication error until a fix is installed.

Evaluation during Operations

We began operating the DCI system in the WRF on Jan. 28, 2004. We have supported 7 PPS tests to date. The number of tests have dropped during April due to some difficulties encountered by the PPS hardware engineers in developing reliable models from the data collected during previous tests. The DCI software operates continuously, even when the PPS is not in test, although the urgency of response is less when out of test. Initially we restarted the DCI system 3-4 times weekly to upgrade the software. Currently we are restarting the system about once a week.

Incidents during Test. The most critical anomaly that we currently detect is a loss of communication in the control system. This anomaly suspends the control software and requires human intervention to reset the control software. We have observed three loss of communication anomalies since the end of January, 2004, when we began operating in the water lab. The first incident, on January 28, occurred at 5:30 AM. At that time our protocols required only the Prime to have a pager. Because his pager batteries had died, the anomaly notice was not received until Prime got into work at 9:00, even though our agent system had attempted to notify him earlier. Personnel who had arrived at the water lab prior to the Prime chose not to call him once they realized there was a problem because they assumed he had already been paged. From this experience we modified protocols in two ways: (1) got pagers for all control engineers, and (2) reminded personnel to call even if pages have been sent (i.e., do not assume paging always works). The remaining two incidents were false alarms that occurred on February 25 and April 7.

False Alarms. We have found the DCI system to be effective at distinguishing false alarms from real alarm situations. For example, we have found that the Loss of Communication anomaly can be induced erroneously by the Coordinator when he is working with the control system in the WRF. As a result, the Prime has learned to first check his group awareness view to see if the Coordinator is working in the WRF before responding to the anomaly. If he is, the Prime calls to discuss the situation with the Coordinator and determine what, if anything, the Prime should do. If the situation is a false alarm, the Coordinator will reset the control system. If the situation is an actual anomaly, the Coordinator usually will handle it since he is already in the WRF.

Simulated Anomalies. We have conducted tests with simulated anomalies in the water lab to evaluate both our agent software and our operational protocols. We do this by simulating an anomaly in the system and letting the control software, the agent system, and the control engineers respond. We conducted simulated anomaly tests on March 2 and March 4. On March 2 we conducted a test where control engineers were notified prior to the test exactly when the anomaly would occur. On March 4 we conducted a test where control engineers did not know when the anomaly would occur. We chose a time when it would be difficult for both Prime and Backup to respond – Prime was in an important meeting at JSC and Backup was in a meeting off site. For the March 2 test we learned that our estimated duration for restoring the loss of communication was too short. We needed to add more time to allow sufficient time to travel from offsite offices to the water lab. For the March 4 test we learned that our pager messages were not very informative when taken out of context of the other information provided in the DCI system. We also learned that even in this worst case scenario where both Prime and Backup were unavailable it was possible to resolve the problem in around an hour using the support provided by the DCI system.

Conclusions

Deploying the DCI system for use in the WRF is proving to be an effective way to mature tools and protocols for distributed space operations. We have found that Ariel personal agents can reliably notify control engineers about important events based on the roles they fulfill. We have anecdotal evidence that our approach of using an automated planner to assign anomaly handling responsibility can reduce time to recover from anomalies from the previous approach of coordination by phone. The protocols for notifying about events and assigning responsibility for handling anomalies continue to evolve as a result of use in the WRF. Currently we are considering revised protocols where we allocate task responsibility based on the availability of engineers at the time the anomaly occurs instead of using a fixed role allocation. We are also implementing protocols that permit engineers to accept or deny a task assignment via a pager for situations where they cannot get online quickly (e.g., a meeting, in a car). We expect to continue our use of DCI in the WRF through Fall, 2004.

Acknowledgements

We acknowledge the support of Dr Michael Shafto, manager of Human-Centered Computing in NASA's Intelligent Systems Program, who sponsored this work.

References

Bonasso, P., Firby, J., Gat, E., Kortenkamp, D., Miller, D, and Slack, M. 1997. Experiences with an Architecture for Intelligent, Reactive Agents. *Journal of Experimental Theory of Artificial Intelligence*. 9: 237-256.

Bonasso, R. P., Kortenkamp, D., and Thronesbery, C. 2003. Intelligent Control of A Water Recovery System: Three years in the Trenches. *AI Magazine* 24 (1): 19-44.

Bradshaw, J. M., Sierhuis, M., Gawdiak, Y., Thomas, H., Greaves, M. and Clancey, W. J. (2000). Human-Centered Design for the Personal Satellite Assistant, In International Conference on Human-Computer Interaction in Aeronautics 2000, Toulouse, France.

Fitzgerald, Will; R. James Firby; & Michael Hanneman. Multimodal Event Parsing for Intelligent User Interfaces. *Intelligent User Interfaces*. Orlando: ACM, 2003.

Scerri, P., David V. Pynadath, Milind Tambe: Why the elf acted autonomously: towards a theory of adjustable autonomy. AAMAS 2002: 857-864

Schreckenghost D.; Ryan, D.; Thronesbery, C.; Bonasso, P.; and Poirot, D. Intelligent control of life support systems for space habitats. *AAAI IAAI* 1998. Madison, WI. July 1998.

Schreckenghost, D., Martin, C., Bonasso, P., Kortenkamp, D., Milam, T., and Thronesbery, C. Supporting group interaction among humans and autonomous agents. *Connection Science.* 14(4) 2002. pp. 361-369.

Sierhuis, M.; Bradshaw, J.M.; Acquisti, A.; Hoof, R.v.; Jeffers, R.; and Uszok, A. Human-Agent Teamwork and Adjustable Autonomy in Practice, in Proceedings of The 7th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS), Nara, Japan, 2003.