Facilitating Interaction among Humans and Immobots about the Job

Debra Schreckenghost and Cheryl Martin

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

Abstract

To effectively support interaction with humans, embodied agents should be implemented using cognitively plausible models such as task models. Even so, these embodied systems should not bear the entire burden for making domain-specific interactions cognitively tractable for humans. Further, embodied systems need not model their users directly; only the necessary and relevant consequences of user state changes or actions should be modeled. The remaining gaps between an embodied system's implementation and the need to support human interaction should be addressed by human-centered software. We describe the use of *liaison* agents to bridge these gaps.

Introduction

We propose that effective interaction between humans and embodied agents working together to perform a job requires knowledge about both the tasks being performed (models of tasks) and knowledge about each other (agent and user models). Task models are cognitively tractable for humans because they represent actions as symbols that are the basis of decision-making. Although the embodied agent must represent and reason about task models to work effectively with the human, it does not need to represent and reason about user models directly. User models can be effectively utilized by a separate liaison agent responsible for managing the communication between the user and the embodied agent. This liaison agent also can help make domain-specific implementation models cognitively tractable for humans by translating between the user's mental models and the embodied agent's implementation models to support shared understanding (Schreckenghost, et al., 2002). Therefore, implementation models such as the embodied agent's physical description of its own parts and subsystems need not exactly match any particular user's corresponding mental model. However these implementation models should still be symbolic and support user interpretation, making the translation possible.

The approach of providing a separate liaison agent that represents an individual's interests has several advantages. It frees the embodied agent from the responsibility of maintaining knowledge about the user that has little bearing on performing the job and that can be computationally "distracting". It permits an existing embodied agent to be augmented post-implementation with communication capabilities in the liaison agent that support human understanding. Finally, this separation of the embodied agent's role from that of the liaison agent allows the user interface and support for interaction to evolve without the need to change the underlying operation of the embodied agent.

We have implemented our approach as the Distributed Collaboration and Interaction (DCI) environment for humans and embodied agents working together (Martin, et al., 2003). Within this environment each person has a personal liaison agent called an Ariel agent to assist its user in performing the duties associated with his or her job. An Ariel agent provides a configurable set of services to help its user that include such capabilities as event notification, daily plans with automatic schedule tracking and updating, command authorization for concurrent operations, and location tracking.

In this paper we first describe the Ariel agents and the DCI environment in which they operate. Next we describe our embodied agent, which is an immobot controlling a crew life support system for water recovery. We then provide detailed examples of interactions between humans and this immobot facilitated by the Ariel agents. These examples are based on our experience with crew life support systems at JSC, and show the effective use of the task models in the implementation of the immobot's control system as well as the effective separation of responsibilities between the immobot and the liaison agents for interacting with humans. We discuss what we have learned by deploying the software at JSC for a water recovery system. We close with how this approach can be applied to other types of embodied agents.

Agents that Facilitate Human-Immobot Interaction

The DCI System provides an environment for collaboration and interaction among a group of humans and highly automated embodied agents. Within this environment, each human in the group has an Ariel agent to assist him or her in working with other members of the group. To manage communication and coordinate activities within the group, each Ariel agent models its user's location, activities, group roles, and health. It uses this information to customize the services it provides its user. Additionally which services are used can be configured depending on the needs of the group. The following services are available for use by the Ariel agent

- *Notification Service*: filters and annotates incoming notices by comparing a set of notice specifications associated with the user's role to descriptors in the incoming notice. The annotation identifies how much latency can be tolerated in notifying the user and how much of the user's attention should be focused on the notice.
- *Location Service*: receives user logins/logouts to the DCI system and translates them to user location changes. As the elapsed time since the login/logout increases, the assessment of location becomes more general.
- *Task Status Service*: tracks completion status of the user's activities and transmits changes to the Conversion Assistant for Planning. The TSS also issues notices requesting the user to acknowledge the assignment of critical tasks.
- *Command and Authorization Service*: works with the Command and Authorization Manager to prevent concurrent activities from interfering with each other by authorizing human activities only if the requested activity does not conflict with ongoing activities. In conjunction with the Assistant for Commanding, it also reconfigures the automated control agents to ensure compliance with the authorization granted.
- *User Interface Service*: manages the presentation of all information to the user, including displays, paging, and the sending of emails.
- *State Management Service/State Data Service*: manages the user's state model, makes that state available to other services, and ensures that updates to that state are handled consistently.

See Figure 1 for a diagram of the DCI architecture.

In addition to the services internal to the agent, the DCI environment makes the following set of centralized capabilities available to all Ariel agents:

- *Activity Planner (AP):* builds and updates the daily duty schedules for all humans in group.
- *Conversion Assistant for Planning (CAP)*: manages plan monitoring and execution for the Activity Planner, including both environmental monitoring and information exchange between the Ariel agents and the Activity Planner. The CAP also monitors for events from the embodied agent indicating a need for replanning.
- *Simple Event Detection Assistant (SEDA):* recognizes simple warning signals from the immobot and translates these signals into human-understandable events such as Caution & Warning events and failure impacts using simple pattern-matching software.
- *Complex Event Recognition Architecture (CERA)*: summarizes operational situations consisting of sets of events that are hierarchical and temporal in nature. CERA was developed by I/Net (Fitzgerald, et al., 2004).
- *Situation Viewer*: provides a user interface for the situations produced by the CERA software. This viewer also retrieves data from parameter logs associated with the situation and shows it in a plot.
- *Command and Authorization Manager (CAM)*: accepts user requests for commanding through their Ariel agents. The CAM mediates between the Ariel agents and the Assistant for Commanding to determine if authorization can be granted and to carry out any reconfiguration of the immobot that is required for the requested commanding.
- *Assistant for Commanding (AFC):* interacts with the immobot to determine current configuration and to change configuration when needed. The AFC also can supplement the models provided by the immobot with models needed to interact with the humans (e.g., models of physical connectivity).

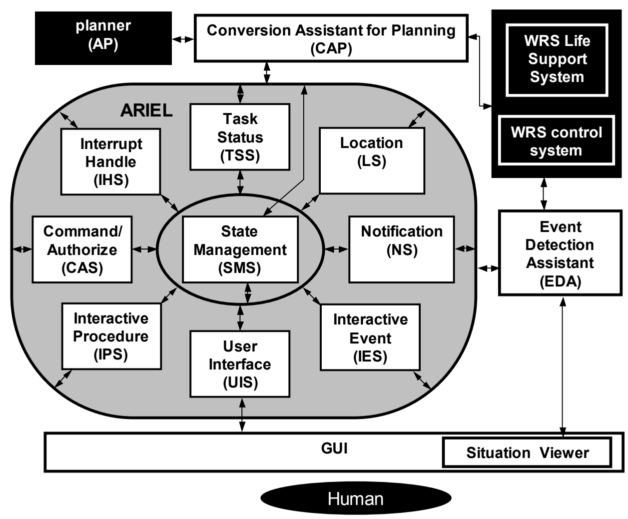

Figure 1. DCI Architecture

Human-Immobot Interaction for Crew Life Support

At Johnson Space Center we are investigating how Ariel agents assist humans interacting with autonomous crew life support systems. These life support systems are responsible to produce products such as oxygen or potable water with minimal human involvement. The tasks they perform to do this involve system reconfiguration and fault management. This combination of goal-driven autonomous control software with life support hardware can be considered an immobile robot or immobot (Williams and Nayak, 1996). We have developed life support immobots for water recovery (Bonasso, et al., 2003) and air revitalization (Schreckenghost, 1998).

The immobots typically utilize the following task models to perform the jobs of reconfiguration and fault management:

- Objectives: goals the mission should achieve
- Plans: tasks defining how to accomplish objectives
- Procedures: instructions defining how to accomplish plans
- Skills: low-level capabilities required to accomplish procedures

We use the 3T cognitive architecture (Bonasso, et al., 1997) to implement these task models. The use of these models in the control implementation itself facilitates communication about the job between the immobot and the human. These models can be used for communicating about shared goals, transitioning between manual and automated task execution, exchanging knowledge about needed skills or capabilities, and developing shared plans. As described below, however, task models alone are not sufficient to ensure effective interaction among humans and embodied agents. Specifically, information about who to notify when something important happens and how to notify them is needed. It also is necessary to provide information for coordinating concurrent activities to ensure they don't interfere.

We are currently evaluating human-immobot interaction at JSC with control engineers performing their duties associated with a multi-month ground test of a regenerative crew water recovery sub-system, the Post Processing System (PPS). The PPS is a water "polisher" used to remove the trace inorganic wastes and ammonium in recycled water using a series of ion exchange beds and to remove the trace organic carbons using a series of ultraviolet lamps. The control engineers are responsible to handle occasional anomalies in the PPS and to perform maintenance, such as sensor calibration. These tasks are assigned based on roles. The person assigned the role of PRIME is responsible to handle all problems in the life support system that the immobot cannot handle. The person assigned the role of BACKUP is called in when the Prime is unavailable. Both of these roles rotate weekly. The person assigned the role of COORDINATOR is knowledgeable about the life support domain. The COORDINATOR coordinates the duties of the team, monitors the overall operation of the PPS control system, and steps in to handle problems that are beyond the experience of people fulfilling the other roles. This role is assigned permanently to one person.

In the remainder of this section, we describe three cases where the liaison agent operating in the DCI environment facilitates interaction between the human and the immobot controlling a crew life support system. Each of these cases includes an example taken from the water recovery domain.

Event Notification

In order to work together effectively, humans need to maintain awareness about an embodied agent (awareness of anomaly warnings, important changes in the system's operating characteristics, etc.). Unfortunately, as studies of human cognitive function have shown, humans have limited attentional resources (Woods et al., 2002), and they are not naturally good at vigilant monitoring (Sumwalt et al., 2002). In addition, humans in our life support application must be free most of the time to do other things, unrelated to the immobot. Therefore, important requirements for an embodied agent/human interface include both (1) identification of situation changes and events related to the embodied agent that are important to the humans and (2) notification to the humans about these changes or events. In the DCI approach, the responsibility for meeting these requirements is shared between the embodied agent (immobot) and the DCI system. The majority of this responsibility rests appropriately, we believe, with the DCI liaison agents and associated support software.

For its part, the immobot must externalize important information about its operation (data values, state changes, warnings, etc.), such that this information is visible to the outside world, including the DCI system. Thus, the immobot is responsible to identify what information *may* be relevant or important to external entities, including humans, but is not responsible for any processing related to what final form that information should be in, who might be interested if anyone, or how to inform them.

The DCI system handles these remaining tasks, thus freeing the limited computing resources in the immobot for time-critical processing, as needed to perform its primary functions. Specifically, the DCI system handles the following aspects of event notification:

- One or more Event Detection Assistants (EDAs) process information exported by the immobot. The EDAs recognize which signals or patterns of data from the immobot are important to the humans. In this sense, the EDAs act as a filter protecting humans from information overload, which helps them better allocate their attentional resources. Further, the EDAs translate the information coming from the immobot (information tied to symbolic knowledge representations appropriate for the immobot's implementation) to information better matched to human mental models (mental representations of the immobot's processing or physical description). Upon recognition of human-relevant information from the embodied system, the EDAs generate their own events for the humans, which contain information suited to human understanding at appropriate levels of abstraction. We currently have two capabilities that operate in parallel for event detection, the Simple Event Detection Assistant and the Complex Event Recognition Architecture, as described above.

- The EDAs produce events for humans, in general, and each liaison agent associated with a particular human receives these events. Since a liaison agent best knows the characteristics of its individual user (his or her state, roles, etc.), it is the liaison agent's responsibility to determine if its user should be notified of that event. In this way, the liaison agent personalizes for its user the support DCI provides for managing attentional resources and information overload. The liaison agent uses information about its user as well as the user's own information preferences and the organization's policies about information distribution and situation-awareness requirements to determine if an incoming notice or event is of interest (Schreckenghost et al., 2002).
- Once the liaison agent determines that an event should be passed on to its user, it also personalizes *how* that notice is presented to the user. Based on the user's state model, organizational policies, and user preferences, the liaison agent tailors the latency, the demand for shifting the user's focus of attention, and the modality with which the notice is presented to the user (Schreckenghost et al., 2002). In the future, we hope to improve this customization of notice presentation to the user by, among other things, using cognitive models of the user's attentional processes to help minimize user interruption and distraction.

To support human interaction with the water recovery immobot, the DCI system currently monitors for failure signals such as loss of control communications as well as data patterns of interest such as end-of-test situations. Different humans are notified in different ways about these events. For example, the COORDINATOR receives more of these events with a low saliency (no focus-of-attention shift), the PRIME receives events related to anomalies with a high saliency (via pager if he is offline when the event occurs), and the BACKUP receives only archival copies of most events so that he is not distracted unnecessarily from performing duties in his other roles.

Anomaly Management

The immobot operates autonomously most of the time. Problems can occur, however, that require human intervention (Dorais, et al, 1999). For example, failed components must be repaired or replaced by a person and, in some cases, there may be components that can only be reset or adjusted by humans. When such problems occur, the human needs to take action to aid the immobot both to recover from the impacts of the problem as well as to fix the cause of the problem.

Because the person has other duties, however, these activities to assist the immobot must be interleaved with his or her other scheduled activities. The DCI system provides an automated planner to manage human activities. This planner uses a cognitively plausible model of tasks to build a centralized activity plan that allocates these tasks to individuals based on a simple model of skills. The DCI system augments traditional HTN planning with a plan monitoring and execution capability for humans, which is performed by the Conversion Assistant for Planning (CAP). The CAP performs a variety of functions for mediating between the immobot, the liaison agent modeling the human, and the Automated Planner building human activity plans. The CAP monitors events from the immobot to determine when situations occur that change the constraints or objectives under which the coordinating plan was built, requiring a replan. Such changes include problems that require human intervention. When such a problem is detected by the CAP, it adds a new goal for fixing the problem to the list of activity goals and initiates replanning. If the problem detected is urgent (i.e., time critical), the activity to achieve the new goal is scheduled as soon as possible. If the problem is not urgent, it may be scheduled for a later time.

Once a new activity plan has been built, it is passed to each liaison agent in the group. Each agent determines how to inform its user about the new plan. This includes whether to interrupt its user and, if so, how emphatically to interrupt her based on the importance of the changes in the schedule. The agent determines the importance of these changes based on a model of task criticality. If its user has been assigned a critical new task, such as an anomaly handling task, the liaison agent will issue a notice requesting the user to acknowledge the new task. If the user is logged into her agent, this urgent notice will be annunciated by a flashing icon on the DCI toolbar and a beep (see Figure 2).

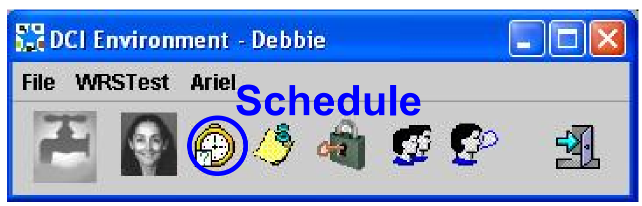

Figure 2. DCI Toolbar with Schedule Icon Highlighted

If the user is not logged into her agent, she will be paged. From either the agent interface or the pager, the user can accept or reject the task. If the user accepts the new task, the liaison agent marks the task as acknowledged and passes this status to the CAP. If the user rejects the new task, the CAP interprets this as an indication that the user is temporarily unavailable. The CAP informs the planner that the user is unavailable and initiates replanning to assign the task to another available person.

The liaison agent also monitors for evidence that the user has started and completed assigned activities. This evidence can include changes in user state such as the user moving to the location where the activity should take place. These changes in activity status are passed to the CAP, and the CAP translates them from values the human is comfortable with (i.e., matching the human's mental model) into values the automated planner can utilize (e.g., task *not-complete* becomes a goal *failed*). When activities are of low criticality, the liaison agent usually assumes they are complete when the estimated completion time passes. The human can correct errors the agent makes in marking task completion status. These corrections are sent to the CAP, and are reflected in the next plan update. For example, tasks marked *complete* will be removed from the list of goals and thus will not be in the next plan update. Similarly, tasks marked *not-complete* will be added back to the list of goals and should a part of the next plan update.

To illustrate how anomaly handling works, consider the following example from the Water Research Facility (WRF). In the PPS, the loss of RAPS communication (LORC) between the control software and the PPS hardware requires a human to be called in immediately to repair the problem. When a LORC occurs, a new activity to restore communication is added to the schedule of the control engineer assigned to handle problems that week (the PRIME). Because this failure isolates the hardware from the control system, it requires timely response from the human, and the liaison agent notifies the PRIME to acknowledge the new task and go to the WRF as soon as possible. If the PRIME accepts the task, his agent marks the task as acknowledged and begins to monitor for evidence that he has started the task (such as observing him relocate to the WRF). If the PRIME rejects the task or fails to respond within 15 minutes, the CAP informs the Activity Planner that the PRIME is not available for tasks and initiates replanning. The new plan reassigns the activity to restore communication to the BACKUP. The BACKUP's agent begins to monitor for evidence that he has started the task. Once the restore communication activity has initiated, the liaison agent begins to monitor for evidence that the task has been completed. The evidence for task completion is the restoration of RAPS communication (RORC) between the control software and the PPS hardware, indicated by a data signal called a RORC message. When the liaison observes a RORC message, it marks the restore communication activity complete and passes this to the CAP. The CAP translates this to a goal succeeded, and passes this to the Activity Planner. The anomaly is now resolved.

Command Authorization

Effective interaction between humans and an immobot often requires specialized support for coordination outside planned actions as well. In our life support application, multiple humans and the immobot's control software can take actions on the same underlying physical system (the immobot's hardware). The command and authorization services in DCI assist humans in taking these actions with a decreased risk of interfering with each other or with the immobot. In general, the benefits of commanding and authorization support include decreased risk of issuing conflicting commands, decreased risk of impeding the work of another agent (human or immobot), and decreased risk of the physical hardware system being put into a bad state (Martin et al., 2004a).

Coordination support for commanding and authorization is needed in our current application because (1) simultaneous action on the system is often required or desired (by multiple humans or by a human and the immobot), (2) commands may be issued remotely, so humans may otherwise have poor visibility into ongoing activities that are affecting the immobot, and (3) even though a great deal of the interaction between crew or ground test engineers with the immobot would be managed by a planner, unanticipated circumstances may give rise to unplanned actions, which must be coordinated outside the plan.

The command and authorization capabilities in DCI help users interact with the immobot at the task and procedure levels of abstraction. A procedure model is a process model that indicates the possible sequences of actions required to carry out a given task. Procedure models are cognitively tractable for humans and also map directly to implementation models in the immobot's 3T architecture. For both humans and the immobot, procedure models are identified with or indexed by a task name or identifier. Commands (actions taken on the immobot's physical system) are considered by both humans and the immobot in the context of these task and procedure models.

The DCI command and authorization capability assists users in performing commands and in reconfiguring the system so it is safe to perform a given procedure. It also provides command lock-outs by selectively granting authorization to act on the system to prevent possibly conflicting commands from different humans. These authorizations are granted for the duration of a task that the user requests to perform, and the procedure model for how that task is carried out indicates what the scope of the authorization should be (e.g. system or subsystem).

In the current DCI implementation, a user should request permission to perform a particular task related to the Water Recovery System (WRS) through his or her liaison agent. To grant commanding permission for a given task, the DCI system must first, if possible, grant authorization for the set of actions (including reconfiguration actions) required by the task's procedure, and then reconfigure the WRS hardware and the control software to the proper state required for the start of the procedure. To do this, the liaison agent requests authorization from a centralized Command & Authorization Manager (CAM), which checks that the requested task's procedure will not conflict with other ongoing procedures. If conflicts are found, authorization is denied, and this result is returned to the user along with a description of configuration conflicts. The human can override the authorization denial if he or she chooses. If authorization is granted, and the user wishes to continue, the CAM initiates any required reconfiguration on the WRS including orchestrating required manual actions by humans. The reconfiguration actions taken by the immobot are in turn triggered by a software module called the Augmentation for Commanding (AFC). The AFC shares static implementation models with the WRS immobot of both the physical WRS system and the procedures that can be performed on the system (including reconfiguration procedures).

Once the reconfiguration, if any, is complete, the CAM informs the user through his or her liaison agent that the WRS is ready to command. The user can then proceed with the actions required by the procedure for the requested task. When the user has completed the task, he or she requests the CAM to release commanding permission for the task. The CAM informs the AFC that the activity's configuration is no longer required (which may result in additional reconfiguration of the WRS by the AFC to “un-configure” for the activity) and then releases the authorization. For a detailed description of these command and authorization capabilities as well as specific examples of WRS tasks supported, see (Martin et al., 2004b).

Conclusion

Preliminary results from our evaluation of the DCI system to support control engineers interacting with the PPS in the WRF indicate that the liaison agents are an effective way to facilitate interaction between humans and life support systems. We have found that Ariel 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 the time to fix the problem from the previous approach of coordination by phone. We also have evidence that the interaction between the human and the immobot is evolving as a result of introducing the Ariel agent. For example, the protocols for notifying about events and assigning responsibility for handling anomalies continue to evolve as a result of use in the WRF.

We have described our use of liaison agents that model their users to facilitate interaction with automated control agents for crew life support, an immobile embodied agent. We believe that our use of facilitating agents will extend to other types of embodied agents that need to coordinate with humans to perform a job, including autonomous mobile robots and high-autonomy space vehicles like the NASA”s Crew Exploration Vehicle. For robots, we propose that the DCI system can improve joint human-robot task execution by modeling human activities and actively tracking their execution. We also propose that the DCI system can improve the allocation of tasks among humans and robots, including anomaly handling, by modeling human skills and actively updating these models based on observation of human performance. For highly autonomous vehicles, we propose that the DCI system represents an important component in the crew cockpit of the future. Similar to crew life support systems, DCI can provide event notification, anomaly management, and support for concurrent commanding by means of command authorization for autonomous vehicle systems. We propose that DCI can improve joint human-autonomy task execution by providing models of human state that are actively updated by observation. In the future we plan to investigate how a combination of a richer sensing environment and improved models to reason about this sensed data can increase the fidelity of our human modeling, thus improving the embodied agent’s ability to know the human’s state.

Acknowledgement

We acknowledge the support of Dr Michael Shafto, manager of Human-Centered Computing in NASA’s Intelligent Systems Program, who sponsored this work. We also recognize R. Peter Bonasso, Mary Beth Hudson, Tod Milam, and Carroll Thronesbery each of whom developed some of the software of the DCI environment and the Ariel agents described in this paper.

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.

Dorais, G., Bonasso, R. P., Kortenkamp, D., Pell, B., and Schreckenghost, D. Adjustable autonomy for human-centered autonomous systems. Proceedings of Adjustable Autonomy Workshop. International Joint Conference on Artificial Intelligence (IJCAI) Aug 1999.

Fitzgerald, W., R. J., Firby, A. Phillips., and J. Kairys. Complex event pattern recognition for long-term system monitoring. AAAI Spring Symposium. *Workshop on Human Interaction with Autonomous Systems in Complex Environments*. March 2003. pp 104-109.

Martin, C. E., Schreckenghost, D., and Bonasso, R. P. 2004a. Augmenting Automated Control Software to Interact with Multiple Humans. In Proceedings of AAAI Spring Symposium on Interaction Between Humans and Autonomous Systems Over Extended Operation, 141-146. Palo Alto, CA: AAAI Press.

Martin, C., D. Schreckenghost, P. Bonasso, D. Kortenkamp, T. Milam, and C. Thronesbery. Aiding Collaboration among Humans and Complex Software Agents. AAAI Spring Symposium. *Workshop on Human Interaction with Autonomous Systems in Complex Environments*. March 2003.

Martin, C. E., Schreckenghost, D., and Bonasso, R. P. 2004b. Command and Authorization Services for Multiple Agents Acting on an Advanced Life Support System. In Proceedings of AAMAS Workshop on Safety and Security in Multi-Agent Systems. New York, New York, USA

Sumwalt, R. L., III, Thomas, R. J., and Dismukes, R. K. 2002. Enhancing Flight-Crew Monitoring Skills Can Increase Flight Safety. In Proceedings of 55th International Air Safety Seminar, 175-206. Dublin, Ireland: Flight Safety Foundation.

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, Thronesbery, C., Bonasso, P., Kortenkamp, D., and Martin, C. 2002. Intelligent Control of Life Support for Space Missions. *IEEE Intelligent Systems* 17 (5): 24-31

Schreckenghost, D., Martin, C. E., and Thronesbery, C. 2002. Specifying Organizational Policies and Individual Preferences for Human-Software Interaction. In Proceedings of AAAI Fall Symposium on Etiquette for Human-Computer Work (Technical Report FS-02-02), 32-39. North Falmouth, MA: AAAI Press.

Williams, B, and Nayak, P. Immobile Robots: AI in the New Millenium. *AI Magazine.* Fall 1996.

Woods, D. D., Patterson, E. S., Roth, E. M., and Christoffersen, K. 2002. Can We Ever Escape From Data Overload?, Technical Report, Ohio State University.