Helping Humans: Agents for Distributed Space Operations

C. Martin, D. Schreckenghost, P. Bonasso, D. Kortenkamp, T. Milam, C. Thronesbery
NASA Johnson Space Center
TRACLabs
1012 Hercules, Houston, TX 77058, USA
cmartin@traclabs.com, ghost@ieee.org,

Keywords Agents, Autonomous Systems, Distributed Collaboration

Abstract

This paper describes the Distributed Collaboration and Interaction (DCI) environment, which supports interaction among humans and automated software systems. The DCI approach uses intermediate *liaison agents* associated with each human to provide an interfacing layer between the human and the automation. We have applied a DCI prototype to help human engineers interact with automated control software for the advanced Water Recovery System (WRS) at Johnson Space Center (JSC). This paper describes this application and the DCI design and implementation.

1. Introduction

Future manned space operations are expected to include a greater use of automation [6]. This automation will function without human intervention most of the time. However, humans will be required to supervise the automation, and they must be on-call to respond to anomalies or to perform related tasks that are not easily automated. In such an environment, humans perform other tasks most of the time, and their interaction with the automation may be remote and asynchronous. As automation becomes more prevalent, better support for such interaction is needed.

We are investigating the use of software agents to assist humans in this type of remote, distributed space operations. We have applied our approach to human interaction with control automation based on our experiences as control engineers for ground tests of advanced life support systems at Johnson Space Center (JSC) [13]. Control automation for advanced life support systems operates continuously to perform routine control operations such as vigilant monitoring and managing anticipated failures. Humans need to interact with this control automation for a variety of reasons including supervisory monitoring, modifying control parameters, maintaining or repairing underlying hardware or software, responding to anomalies, and taking advantage of opportunities.

To support these types of interaction, we have developed the Distributed Collaboration and Interaction (DCI) environment. The DCI approach uses intermediate *liaison agents* associated with each human to provide an interfacing layer between the human and the control automation. These liaison agents and the remainder of the DCI architecture, described in Section 3, provide a variety of services, which together support:

- human supervision of automated control systems,
- direct human control of processes such as crew life support,
- activity tracking and coordination among humans and automated systems interacting with the same process, and
- asynchronous information exchange among distributed, remote humans and automated systems.

This paper provides an overview of the DCI system and a description of an implemented software prototype. Section 2 describes our prototype application, and Section 3 provides a description of the DCI architecture. Section 4 presents a scenario demonstrating the use of DCI in the prototype application, and Section 5 concludes the paper.

2. DCI Prototype Application Domain

Since 1995, we have been developing intelligent control systems for advanced life support [3, 13]. These control systems have been realized using an architecture known as 3T [2], and were designed to run autonomously for months at a time. 3T is a layered control architecture whose top tier is a hierarchical task net (HTN) planner, the plans of which are executed through a reactive middle tier that in turn manages the sensors and actuators of the hardware via a low-level control tier.

One such life support system is the advanced Water Recovery System (WRS). Developed at Johnson Space Center (JSC), the WRS is comprised of four hardware subsystems that remove the organic and inorganic materials from waste water (hand wash, shower, urine and respiration condensate) to produce potable water. From January 2001 through April 2002, the 3T system controlled the WRS autonomously in a continuous 24/7 integrated test [3]. An early version of the DCI prototype was deployed with the WRS 3T system in April 2002, and we have continued to develop the prototype and apply it to this application using a simulation of the WRS 3T system.

During WRS operation, three human control engineers are responsible for monitoring and occasionally intervening in WRS operations while spending the majority of their time carrying out their daily tasks on unrelated projects. One person, the Prime engineer, has first responsibility for responding to problems in the WRS; a second person, the Backup engineer, has the job of taking over if the Prime is unable to respond to the problem. The Coordinator oversees the work of the other two people and also serves as the secondary Backup.

The DCI prototype aids remote monitoring and control of the life support system and coordinates the actions of the control team. It builds the engineers' daily schedules, tracks the completion status of tasks on their schedules, and updates the schedules when WRS problems arise that require new human tasks. The DCI liaison agents notify engineers of WRS events, including problems, based on the humans' currently assigned roles and availability. Assessing engineer availability and current task status are important DCI functions that rely partially on the underlying capability to track human location. The location tracking capability of DCI uses both sensed physical coordinates and computer access monitoring. The following section describes the DCI system in detail and shows how the current DCI prototype supports humans in interacting with the WRS control automation.

3. DCI Design and Implementation

Liaison agents are central to the DCI system because they enhance humans' ability to interact with other autonomous software and other humans. Using agents to support human interaction has also been explored by previous related work. The Electric Elves system is a very successful and innovative implementation of interaction between humans and software agents [5]. In this system, "proxy" agents for each person perform organizational tasks for their users such as monitoring the location of each user, keeping other users in the organization informed, and rescheduling meetings if a user is absent or unable to arrive on time. The MokSAF environment provides another example, in which "interface" agents assist humans with a military route-planning task [10]. These projects and other work in human-agent interaction [9, 11] have helped shape the design of the DCI system.

Figure 1 depicts representative elements of a DCI system. The liaison agents in DCI are called Attentive Remote Interaction and Execution Liaison (ARIEL) agents, in deference to Shakespeare's Tempest character. In addition to liaison agents, DCI provides *augmenting software*. This software includes stand-alone DCI tools or DCI tools associated with other existing software. In Figure 1, the Event Detection Assistant (EDA) and the Conversion Assistant for Planning (CAP) are representative pieces

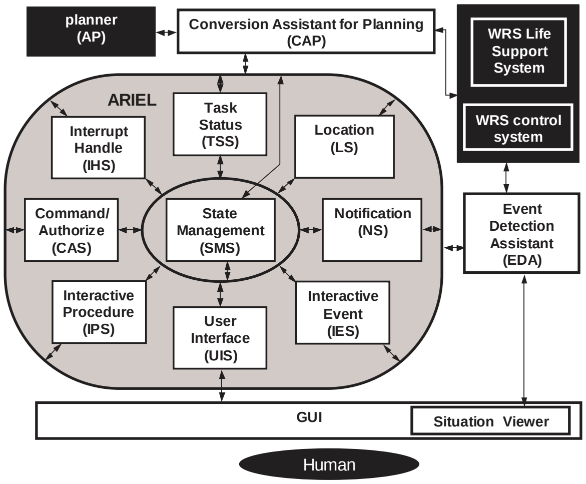

Figure 1. DCI Architecture

of augmenting software. The entities with black backgrounds in Figure 1 (the human, the WRS system and its control software, and the centralized planner) participate in, but are not part of, the DCI environment. This section describes each element of the DCI system shown in Figure 1.

The Event Detection Assistant (EDA) monitors data produced by the control automation and searches for patterns in this data that are of interest to humans supervising this system. The EDA is implemented using the Complex Event Recognition Architecture (CERA) [8]. As specified patterns are detected in the control data, the EDA generates and broadcasts its own events about these data patterns. This capability serves two purposes in the DCI prototype (1) to cause patterns of low-level WRS control data to trigger the generation of events with higher-level meaning that are represented using abstractions suited to human understanding and (2) to recognize patterns and trigger events that are needed by other software supporting humans, such as a centralized planner that is watching for events indicating that a human should be assigned a new task.

The DCI design off-loads this event detection functionality from the automated control system to a separate system with its own computational resources. Our approach avoids overburdening resource-limited control automation, whose processing may include time-critical tasks, with tasks that are not directly related to its primary objectives.

In the DCI system, the coordination of control automation and crew activities is based on a centralized high-level group activity plan. This approach is designed to prevent conflicting commands from humans and automation, to ensure activities are assigned to accomplish all operational goals, to avoid over-subscribed agents, and to assist handover between manual and automated tasking. Our planner is a hierarchical task net (HTN) planner, known as AP, that is capable of automatically monitoring and updating its plans [7]. The Conversion Assistant for Planning (CAP) augments the planner's ability to interface with humans. The CAP implementation is tightly coupled to, and shares models with, the automated planner. The CAP reacts to contingencies in the control situation that are relevant to human activities by monitoring events detected by EDA and automatically triggering replanning in AP as needed. The CAP can also trigger replanning upon a human's failure to complete a task in a timely manner. The CAP interacts with the ARIEL agent for each human to determine when human tasks in the plan are completed successfully or when they fail to complete.

ARIEL agents, as pictured in Figure 1, provide many services for and on-behalf-of their human users. These services are described in the following subsections.

3.1. State Management Service (SMS)

To assist a human user in the ways most appropriate for his or her current job context (including task, role, location, etc.), an ARIEL agent maintains a model of this current context. The SMS makes this model available to each of the other ARIEL services.

3.2. User Interface Service (UIS)

Each ARIEL agent is designed to provide support for a particular human, and therefore must have a rich user interface. The UIS manages all direct interaction between a user and his ARIEL agent. It invokes different modalities, such as display, pager, or email, to present information in the manner most appropriate to the user's current job context.

3.3. Notification Service (NS)

Communication protocols are used in manned space operations to ensure information is exchanged correctly, leading to timely decisions. Extending these protocols to support distributed collaboration must take into account the possibility of remote and asynchronous notification and ensure the proper routing of information to different humans based on the roles they currently hold. The NS supports both of these requirements by filtering notices and providing guidance about appropriate notice presentations for its user.

The NS uses information about (1) organizational policies for information distribution and situation-awareness requirements, (2) the user's own information preferences, and (3) the user's current roles and state (e.g., location) to determine if an incoming notice is of interest to a user, and if so, how to inform the user. The notices processed by the Notification Service include operational events (e.g., those generated by the EDA) and events about the activities of other members of the operational team, which are generated by other ARIEL agents. The NS implementation and a description of specifications that control the filtering and direct the presentation of notices in the NS are described in detail in [12].

In the DCI prototype application, the NS that is associated with the person who has the Prime role ensures that he is notified of important WRS anomaly events with high saliency. If his current state model indicates that he is remote and offline, then he is notified of important events via a pager message. If he is online, he is notified via a workstation display change. However, the Backup is allowed to continue her current task without distraction because her NS simply logs anomaly notices without demanding her attention.

3.4. Interactive Event Service (IES)

Occasionally, a human may need information, via event detection and notification, for which event detection definitions and notification specifications have not been previously established. The IES assists a user in interactively defining temporary or new operational events and controlling automated monitoring for these events (e.g., through the EDA). The IES interacts with the Notification Service (NS) to ensure that the user is notified of these newly defined events as they occur. The DCI prototype does not yet implement the IES. The current design for the IES would allow each engineer, in response to unusual control situations or operations, to define temporary monitors for operational changes in the WRS.

3.5. Task Status Service (TSS)

The activities of humans and software control systems can be coordinated to achieve overall mission objectives by using a centralized activity planning approach as described previously in this section. However, applying a planner to human activities poses challenges because, for example, humans do not usually provide as much feedback about their activities to the planner as do software systems. Humans may fail to acknowledge tasks before starting to execute them or fail to provide evidence that tasks have been completed. A planner’s ability to monitor for plan success or failure is greatly compromised under this type of open-loop operation. To assist in closing the loop for human activity planning, the TSS performs human activity tracking and provides feedback to the automated planner.

The TSS tracks the status of human tasks including assignment, acknowledgement, initiation, and completion as well as various failures that can occur (e.g., failure to complete a task within the allotted time or failure to acknowledge a task). The TSS receives updates from the planner about assigned tasks that are ready for the human to execute. It also interacts with the human user to obtain acknowledgement of assigned tasks that are modeled as time-critical tasks (i.e., tasks that must be performed within a given time period to avoid adverse effects). The TSS is designed to be as non-intrusive as possible, so, for tasks that are not time-critical, the TSS does not require explicit acknowledgement from the user. The TSS infers task initiation and completion using the best evidence it has available, including (in order of strongest to weakest evidence): (1) direct evidence obtained by monitoring the effects of actions in the data from the control system through the EDA, (2) direct evidence from monitoring computer-mediated manual tasks, (3) indirect evidence based on user location changes and the location where a task should be performed, and (4) indirect evidence based on the scheduled time of a task, the current time, and whether or not the user has previously viewed that task in her schedule. Although the weakest indirect evidence may be acceptable for non-critical tasks, stronger evidence is required to assess task status for critical tasks.

In the DCI prototype, WRS repair tasks are modeled as time-critical tasks, and routine daily tasks such as meetings and procedure authoring are non-time-critical. When a WRS repair task is assigned to the Prime due to a control failure, the TSS, in conjunction with the User Interface Service, requests an explicit acknowledgement that the Prime has received and will perform the assigned task. If the Prime does not acknowledge the task within a pre-defined time-out period, the TSS marks the task as failed (failure to acknowledge), and this triggers the automated planning system to reassign the task to the Backup.

3.6. Interruption Handling Service (IHS)

Distributing control operations leads to remote and asynchronous interactions among people and software.

In such an environment, the potential for interruption and distraction is very high. The IHS coordinates the actions of other services (the Notification Service and Task Status Service, for example) to minimize the impact of interruptions on the user's primary task. Support for interruption handling includes (1) determining when the user should be interrupted and how intrusive the interruption should be, (2) mapping the human concepts of task status at interruption (delayed, deferred, suspended) to the changes needed to update the plan from an automated planner (e.g., goal changes, task completion status changes), and (3) assisting the user in managing multiple, concurrent threads of activity. The IHS is not yet implemented in the DCI prototype. In the future, it will support operations such as assisting the Prime or Backup in making a smooth transition from a normal daily activity to a time-critical WRS repair task and then in resuming the previously interrupted activity.

3.7. Location Service (LS)

Location tracking is a fundamental capability supporting many of the other ARIEL services. The LS provides a human's location information for use in tracking the completion status of user activities, determining how to notify the user of events, and customizing presentation of information (e.g., handheld display versus workstation display). This service combines sensed user location from external sensors with computer login/logout events to determine the user's physical location and cyber location. The user's physical location and cyber location are combined to assess the user's *presence* (availability and accessibility).

Login and logout events help the LS to determine the human's *cyber location*. Information modeled about cyber location includes whether or not the user is currently online and which display platforms she is currently using. The human may use several network locations simultaneously. For every login event, the network address, host name, and platform type (e.g., workstation, handheld, laptop) are recorded. As long as there is one active login, the user is considered online. After the user logs out of her last session or after the last session has timed out, she will be considered offline.

A possible extension to this work will consider updating the timestamp of the cyber location whenever an input is received from a network login location. This will enable us to update the current cyber location more effectively and avoid timing out an active session.

Two sources of information are used to compute physical location. If the user logs into a platform that is not mobile (e.g., not a laptop or handheld) then the static network address and host name are mapped to a physical location using an ontology model. When more than one network login is active, the most recent login is used for physical location.

Physical location also can be monitored via sensed location coordinates. These coordinates are similarly mapped to the nearest physical location milestone using the location ontology. Over a configurable amount of time, the precision of the currently modeled location degrades to a more general milestone.

In the DCI prototype, examples of physical location milestones include *Water Lab*, *office*, and *NASA JSC*. As confidence in the value of sensed location information degrades over time, the precision with which an engineer's location is modeled may degrade from her *office* to *NASA JSC* and then to *Houston*.

The current prototype uses the Global Positioning Satellite system (GPS) to track a user's physical location outdoors. When she enters a building, the last known coordinates are used by LS as an approximate location (the nearest building). Using GPS for location sensing requires providing satellite signals to the ARIEL agents. Therefore, GPS cannot be used indoors due to the signal-shielding effect of buildings, and GPS cannot be used in space because GPS is a ground-directed system. For future implementations, we are investigating the use of RF-based tracking using 802.11 based hardware [1, 4, 14] for sensing physical location inside any enclosed area. RF-based tracking uses radio frequencies, as opposed to infrared, to transmit information. We plan to use low-cost hardware based on the 802.11b or 802.11g standards, which both use the 2.4 GHz range. We will be using at least 3 *access points* to provide RF signals to mobile computing platforms. An access point is a bridge between wireless and wired networks, so it also provides wireless network access for these mobile platforms.

In RF-based tracking, multiple access points are placed at known locations and a signal strength map is created. This map indicates the strength of the signals from each of the access points at discrete locations in the surrounding area. The density of these locations will help determine the accuracy of the tracking.

A mobile platform (e.g., laptop or handheld) equipped with a wireless card will be able to measure the signal strength from the access points and match these to the signal strength map to triangulate the current position. Several ways of performing the match have been researched. The Nearest Neighbor in Signal Space technique was presented in [1]. This technique was shown to approach 2-meter accuracy when using environmental profiling, which takes into account different signal strengths during different times of the day. A table lookup approach was used in [14], and [4] introduces a Bayesian network scheme for tracking location based upon not only signal strength, but probabilistic modeling as well. This last system has been implemented as Nibble, which we will evaluate to determine if it will fit in with our existing system.

In the DCI prototype, physical location information is used primarily for activity tracking and group awareness. For example, when the Prime (or Backup) enters the water lab during a time that he (or she) is assigned a WRS-related task, the Task Status Service can infer that he has initiated that task. When a human's physical location changes, his ARIEL agent informs the other humans by generating an event and sending it to all other ARIEL agents. Cyber location information is used primarily to determine the most appropriate notification method and to present information to the user in the most effective way for his current location.

3.8. Command and Authorization Service (CAS)

When situations arise in an automated control system that fully autonomous operations cannot address, it is necessary to support some level of human intervention into the hardware system being controlled. *Human commanding* refers to a human's action of issuing directives to the underlying physical system (e.g., turning on a pump). When possible, such commanding should be mediated through the control automation. DCI supports such mediation, which allows the control automation to maintain a current and consistent model of the control state, even when asynchronous intervention occurs.

The CAS will assist a human in commanding by reconfiguring automation for manual actions and by providing access to computer-based interfaces for the execution of control procedures. The CAS will coordinate with the control automation to ensure that remote users are authenticated and authorized to command, and it will resolve conflicting commands from users at different locations.

The CAS is not yet implemented in the DCI prototype. We are currently investigating a design for automatically reconfiguring the control automation when users perform standard manual procedures. In the future, the CAS will help WRS engineers adjust the control automation to suspend appropriate automated actions for the duration of manual actions (e.g. suspend automatic shutdown during maintenance activities such as sensor calibration).

3.9. Interactive Procedure Service (IPS)

By mediating human commanding through an existing software control system, DCI (1) allows the control system to operate continuously and maintain a consistent state and (2) provides opportunities to track user activity by monitoring software interaction. However, the standard procedures used by a control system do not often support all actions a user needs to take. Computer mediated commanding capabilities can be greatly enhanced by allowing a user to modify control procedures that are normally executed by control automation. This *procedure modification* capability allows a user to specify new operations by modifying an automated procedure and triggering the control automation to perform this procedure. The IPS assists the user in temporarily modifying standard operating procedures executed by the automated control software.

Because procedures and their interactions with other control system operations can be complex, the IPS is designed to guide a user through structured modification of procedures including (1) selecting a procedure from a presented library of available procedures, (2) changing steady state operating parameters, (3) changing to alternative or backup sensors, (4) adding probes to export information about the execution of control tasks, (5) disabling or enabling selected automated operations or responses, or (6) deleting an existing control procedure. Once a user has specified a modification, the IPS loads the modified procedure into the automated control software. The IPS can also later reverse modifications that are deemed temporary.

A standalone prototype of the IPS has been implemented, but not yet integrated into the ARIEL agents in the DCI system. This prototype assists the user in deleting a control procedure temporarily, and in later restoring the unmodified version of the control procedure. This standalone prototype has been exercised on one subsystem of the WRS.

Overall, the current DCI implementation uses Java 1.4 and Allegro Lisp 6.0 with CORBA for interprocess communication. All ARIEL services work together to support an individual human's interaction with automation software and other humans. The following section describes a scenario demonstrated by the DCI prototype, which shows all implemented services working together.

4. Evaluation

We have shown the potential benefits of the DCI system and ARIEL agents by demonstrating the following typical scenario involving the WRS and control engineers.

- A loss of controls communication in the WRS control software requires a human to reinitialize the software.
- The Prime engineer's ARIEL notifies him about the problem and the assignment of a new WRS repair task.
- The Prime engineer is offline and doesn't respond to the pager notification in a timely manner.
- The Prime engineer's ARIEL re-issues a task assignment acknowledgement request with increased urgency.
- The Prime engineer still does not respond, and his ARIEL indicates to the planner that the task has failed.
- The planner reassigns the WRS repair task to the Backup engineer.
- The Backup engineer's ARIEL notifies her about the assignment of a new WRS repair task.
- The Backup engineer is online, and responds to a display notification to acknowledge the change in her schedule.
- The Backup engineer is located remotely from the WRS control system, and must travel to the Water Lab, where the WRS is located, to fix the problem.
- In the Water Lab, the Backup engineer reviews a summary of the anomaly situation via a notice previously logged by her ARIEL.
- Based on that review, she determines how to respond.
- Once the problem is fixed, the Backup engineer's ARIEL notifies her when the water system has returned to normal and she leaves the Lab.
- When the Prime engineer logs into the DCI environment later, he reviews his ARIEL's notifications about how the control team coordinated to resolve the situation and what impacts the problem had on his schedule (if any).
- Throughout this interchange, the Coordinator relies on his ARIEL to inform him of events in the WRS and the response of the control engineers on his team.

As we extend the capabilities of the ARIEL agent, we will be able to support increasingly complex scenarios. In October 2003, we will participate in an integrated demonstration of Agents for Distributed Team Operations (ADTO). This demonstration will bring together several projects from Johnson Space Center (JSC) investigating agents to support both crew and ground controllers. The ADTO scenario will showcase ARIEL support for space station crew and ground-based mission support personnel as multiple anomalies occur and interact to create a novel situation.

5. Conclusion

We have demonstrated that the DCI environment has the potential to provide unprecedented support for distributed collaboration among humans and software systems in the context of future manned space operations. Our previous experiences in building and using complex software automation at JSC indicate that additional support software must be provided to ensure effective human interaction with automation. We have designed the DCI system to address these issues and have successfully demonstrated a prototype implementation. DCI represents a significant step toward providing a productive environment for distributed collaboration in future space operations.

Acknowledgements

We want to acknowledge the support of Dr. Michael Shafto, the manager of Human-Centered Computing in NASA's Intelligent Systems Program, under which this work was done.

References

[1] Bahl, P. and V. N. Padmanabhan, “RADAR: An RF-Based In-Building User Location and Tracking System,” presented at IEEE INFOCOM, 2000.

[2] Bonasso, R. P., J. R. Firby, E. Gat, D. Kortenkamp, D. P. Miller, and M. G. Slack, “Experiences with an Architecture for Intelligent, Reactive Agents,” *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 9, pp. 237-256, 1997.

[3] Bonasso, R. P., D. Kortenkamp, and C. Thronesbery, “Intelligent Control of A Water Recovery System: Three years in the Trenches,” in *AI Magazine*, vol. 24, 2002.

[4] Castro, P., P. Chiu, T. Kremenek, and R. Muntz, “A Probabilistic Room Location Service for Wireless Networked Environments,” presented at Ubiquitous Computing, Atlanta, GA, 2001.

[5] Chalupsky, H., Y. Gil, C. A. Knoblock, K. Lerman, J. Oh, D. V. Pynadath, T. A. Russ, and M. Tambe, “Electric Elves: Applying Agent Technology to Support Human Organizations,” presented at Innovative Applications of Artificial Intelligence, Seattle, WA, 2001.

[6] Cooke, D. and B. Hine, “NASA's New Era in Space Exploration,” in *IEEE Intelligent Systems*, vol. 17, 2002.

[7] Elsaesser, C. and J. Sanborn, “An Architecture for Adversarial Planning,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 20, pp. 186-194, 1990.

[8] Fitzgerald, W., J. R. Firby, and M. Hannemann, “Multimodal Event Parsing for Intelligent User Interfaces,” presented at Intelligent User Interfaces, Orlando, FL, 2003.

[9] Lewis, M., “Designing for Human-Agent Interaction,” in *AI Magazine*, vol. 19, 1998, pp. 67-78.

[10] Payne, T. R., K. Sycara, and M. Lewis, “Varying the User Interaction within Multi-Agent Systems,” presented at Fourth International Conference on Autonomous Agents, Barcelona, Catalonia, Spain, 2000.

[11] Rich, C. and C. L. Sidner, “COLLAGEN: A Collaboration Manager for Software Interface Agents,” *User Modeling and User-Adapted Interaction*, vol. 8, pp. 315-350, 1998.

[12] Schreckenghost, D., C. E. Martin, and C. Thronesbery, “Specifying Organizational Policies and Individual Preferences for Human-Software Interaction,” presented at AAAI Fall Symposium on Etiquette for Human-Computer Work (Technical Report FS-02-02), North Falmouth, MA, 2002.

[13] Schreckenghost, D., C. Thronesbery, R. P. Bonasso, D. Kortenkamp, and C. E. Martin, “Intelligent Control of Life Support for Space Missions,” in *IEEE Intelligent Systems*, vol. 17, 2002, pp. 24-31.

[14] Small, J., A. Smailagic, and D. P. Siewiorek, “Determining User Location for Context Aware Computing Through the Use of a Wireless LAN Infrastructure,” Institute for Complex Engineered Systems, Carnegie Mellon University, Pittsburgh, PA http://www.cs.cmu.edu/~aura/docdir/small00.pdf.