Human Supervision of Robotic Site Surveys

Debra Schreckenghost[1], Terrence Fong[2], Tod Milam[3]

[1]*TRACLabs, 1012 Hercules, Houston, TX 77058*
[2]*NASA Ames Research Center, Moffett Field, CA 94035*
[3]*S&K Aerospace, 201 Flint Ridge Plaza, Ste 102, Houston, TX 77598*

Abstract. Ground operators will interact remotely with robots on the lunar surface to support site preparation and survey. Astronauts will interact with robots to support outpost buildup and maintenance, as well as mission operations. One mode of interaction required for such operations is the ability to supervise robots performing routine autonomous tasks. Supervision of autonomous robotic activities requires monitoring the robot's performance of tasks with minimal human effort. This includes understanding its progress on tasks, awareness when important milestones are achieved or problems impede tasks, and reconstructing situations after the fact by relating task events to recorded data. We are developing a software framework to support such interaction among distributed human teams and robots. We are evaluating our framework for human supervision of mobile robots performing routine site survey operations. We are prototyping a system that (1) monitors data from the K10 robot performing surveys to determine the depth of permafrost at the Haughton Crater on Devon Island, (2) computes performance measures about how well the survey is going, (3) builds summaries of these performance measures, and (4) notifies to appropriate personnel when milestones are achieved or performance indicates a problem. We will evaluate our prototype using data collected during Operational Readiness Tests for the Haughton Crater field test to be conducted in July 2007. In this paper we describe our approach for human supervision of robotic activities and report the results of our evaluation with the K10 robot.

Keywords: human-robot interaction, supervision of autonomous robots, agent-based systems
PACS: 07.05.Mh, 07.05.Wr

INTRODUCTION

Robots are expected to fulfill an important role in manned exploration operations. They will perform uncrewed missions to pre-position resources for manned missions. They will assist astronauts in site preparation and buildup for a lunar outpost. They will support outpost operations and maintenance by minimizing required crew EVA and improving crew productivity on routine tasks. Interaction with robots for NASA space missions today is performed in two main ways – tele-operations for near-Earth missions and carefully scripted task sequencing for deep-space missions (Bresina, et al., 2005; Mishkin, et al., 2007). As NASA prepares for Exploration missions, other forms of human-robot interaction are needed. Task-level commanding and predictive interaction are promising for tasks where time delay and limited bandwidth make tele-operations difficult (Hambuchen, et al., 2006). Supervised autonomy (Sheridan, 1992) has potential to make better use of human resources by reducing the mission preparation time for routine tasks.

But these new types of operations require human operators to support new types of tasks, including maintaining awareness of robotic activities not directly controlled by humans, and detecting situations and incidents requiring human intervention. Health and performance monitoring is needed to assess robot reliability as well as robot task performance. This includes the ability to continuously monitor a robot, compute performance metrics, detect performance milestones and anomalies, summarize performance, and notify responsible personnel of significant events. Support for task coordination is needed to ensure the safety of both astronauts and robots. This is particularly important when controlling robots remotely over time delay. Effective coordination requires that the operational team maintain awareness of the activities, location, and status of distributed robots.

We are developing software aids for humans interacting with robots that operate at different levels of autonomy. Our approach is to provide an agent-based software framework for humans to use when supervising robots. This software framework monitors the health and performance of robots, notifies humans of important events or problems, and assists task coordination by improving situation awareness of distributed robots. In this paper we describe our software framework for human-robot interaction and present the results of a recent experiment with the NASA Ames K10 rovers during the Haughton Crater field test.

APPROACH FOR HUMAN-ROBOT INTERACTION

We support human interaction with robots by providing a software agent for each person in the team. This software agent monitors the health and performance of robots supervised by the people on the team, detects events of interest, and notifies the appropriate personnel based on their assigned roles and accessibility. This agent-based software framework is based on the Distributed Collaboration and Interaction (DCI) software developed to support distributed engineering teams when managing long duration tests of life support equipment at JSC (Schreckenghost, et al, 2002). The DCI system provides a set of configurable services that can be customized for the jobs of the operational personnel. In this application, our software framework supports human supervision of distributed robots. We have selected the following services to support such supervision:

- Event Detection: monitors robot data to determine health and performance, as well as to detect task-related events indicating mission progress.
- Situation Summarization: collects and reports related events useful in understanding progress on task or mission, or needed to reconstruct situations for problem solving.
- Notification: focuses human attention on the most important events as situations change, especially when the human-robot team is distributed.
- Task Coordination: alerts supervisor about ongoing activities and potential interactions between robots and humans to ensure safe operations when astronauts are working near robots controlled from Earth.

In this section we describe how our software framework provides these services.

Event Detection monitors data from the robot to discern when data values indicate a meaningful or significant change in system state or health has occurred. We designed an approach for Event Detection that can be applied with different robots and environments, and supports user definition of events. In this approach, the user describes (1) computations to be performed on incoming data, including combinations of computations, (2) conditions defining when to make computation, and (3) conditions triggering the sending of notices. These descriptions of computations and conditions are stored using the eXtensible Markup Language (XML). A data dictionary is used to associate data items from the robot to these computations and conditions. Using this XML, data computations are linked to libraries of algorithms available for the domain and conditions are loaded into pattern-matching software. When executing event detection, incoming data values are passed to the associated computations and condition monitors. When conditions are met, the associated computations are made and triggered notices are sent to the software agents of interested users. The results of computations can be stored for use by other computations and conditions. Using this approach, an event will occur when conditions are manifested in data or computations.

Situation Summarization collects events and computations from Event Detection and formats them into summaries and reports for presentation to a user. We designed an approach for Situation Summarization that supports user definition of summaries. First the contents of the summary are defined as a set of computed values and events, as described in Event Detection. Next the conditions are defined that determine when a summary should be produced. When these conditions are observed, the computed values and events are stored in a XML data structure. This XML data structure is then formatted for the user using a XML Stylesheet Language (XSL). The user is notified of this summary using the Notification Service described previously. The summary can be viewed from an email or from the archive using a web browser.

Notification brings notices to the user’s attention. These notices are passed from Event Detection or Situation Summarization to users via the DCI software agents. Notices are encoded in XML and communicated over Corba event channels. The software framework filters and routes notices to users based on the roles they fulfill. It uses notification specifications to associate message content with user roles and presentation modalities (Schreckenghost, Thronesbery, and Hudson, 2005). Content used for notice filtering and routing is based on application ontologies, including domain categories (e.g., Robots.Rovers.K10), event categories (e.g., Event.DomainEvent.HaughtonData), and notification categories (e.g., CautionandWarning.Alarm). The notification specification used for notice filtering and routing consists of three parts:

- Context: associates user roles with the specification (e.g., K10 CockpitSupervisor),
- Conditions: describe patterns in the content of an event that must match for the event to be passed to the user (e.g., notification category = Alarm), and
- Directive: identifies the media used to present the notice (e.g., email, computer display) and the urgency and emphasis with which it is presented.

Notice specifications are built for each user role in a domain when deploying the software framework. The specification is represented in XML using ontologies defined for the domain.

Task Coordination is achieved by combining Event Detection and Notification. Event detection is used to monitor for evidence of progress on tasks and to detect anomalies affecting task performance. Notification is used to selectively inform personnel affected by these coordination events. Both people and robots need to be informed about actions and conditions that can impact their ability to safely perform assigned tasks. Robots should be notified when humans are in their workspace. Supervisors should be notified about robot task performance, limitations in its battery power, and human proximity to active robots. Additionally, team members should be notified about events and situation summaries related to their assigned roles. For example, the EVA astronaut is interested in events about robots that are nearby (i.e., robots in whose workspace the astronaut has entered). A remote Supervisor is interested in information about the robot it is supervising, and any humans or robots that are near it. When responding to a problem, a person should be informed of the current situation and the situation leading up to the problem.

The DCI software framework is implemented in Java 1.5. It uses Saxon 8.9 to provide XML stylesheet language translation. It uses JacORB CORBA middleware for interprocess communication. It operates under Windows XP, a variety of Linux platforms, and Mac OS.

We have described our approach for Event Detection, Notification, and Situation Summarization. Taken together, they provide the user with the ability to extract significant events from raw data, collect these events into summaries, and inform the user using a variety of presentation media.

EVALUATION

In this section we describe the site survey conducted by the K10 rovers at Haughton Crater in 2007, the Ground Operations performed at JSC to support this survey, and the use of our software framework to support human supervisors of the K10 rovers during the 2007 Haughton Crater field test.

K10 Site Survey

In July 2007, the Intelligent Robotics Group (IRG) at NASA Ames Research Center conducted a field test of a robotic survey system at Haughton Crater (Devon Island, Canada). Two NASA Ames K10 rovers (see Figure 1) were used to perform a simulated lunar robotic survey of several sites, including a 700m x 700m region called “Drill Hill.” The rovers carried two, non-contact, survey instruments: (1) a Ground Penetrating Radar (GPR) to characterize subsurface structure (such as water layering) via parallel-line transects and (2) a 3D Lidar for topographic mapping via panoramic scans at multiple locations (Fong, et al., 2007).

IRG’s robotic site survey system involves three phases: preparation, execution, and analysis (Fong, et al., 2006). In the preparation phase, 3D terrain modeling is performed using overhead (aerial and/or satellite) imagery of the survey zone, which has resolution (20-50 cm) similar to what will be acquired by the Lunar Reconnaissance Orbiter. A traversability analysis is then performed to distinguish safe terrain (regions the survey rover can traverse) from hazardous terrain (regions the survey rover must avoid). During the second phase, the traversability map is processed by a survey coverage planner, which calculates survey points. A global task executive is employed to command a robot to drive to each survey point and acquire data. The resulting data is then stored in a database for post-survey analysis.

Two third-generation K10 rovers equipped with survey instruments were used during this field test. “K10 Black” was configured with GPR and “K10 Red” was configured with 3D Lidar. K10 is designed to operate in a wide range of environments, from high-friction indoor to moderate natural outdoor (30 deg slope, hard-pack dirt), at human walking speeds (up to 90 cm/s) and over 20 cm step obstacles. The third-generation K10 measures 1.3 m (height) x 0.9 m (width) x 1.0 m (length) and weighs 125 kg (including 50kg payload). The rover is equipped with a variety of sensors for navigation including: carrier-phase differential GPS, electronic compass, stereo vision, and 2D laser scanner. On-board avionics include power conditioning and management, wireless communications, and a Linux based controller (running on a dual-core Pentium laptop). K10 is powered by Li-Ion batteries, which can be “hot swapped” to enable continuous, long-duration operation.

FIGURE 1. NASA Ames K10 Rovers Conducting Systematic Site Survey at Haughton Crater.

Two, non-contact survey instruments were used for the Drill Hill survey: (1) the CRUX ground penetrating radar (to characterize subsurface structure) and (2) an Optech ILRIS-3D scanning lidar (for high-resolution topographic mapping). Developed for the lunar "Construction and Resource Utilization Explorer" (CRUX) project by NASA JPL, the CRUX GPR is optimized for lunar prospecting: it has relatively shallow penetration (~5 m depth) and high resolution (15 cm). The GPR is a short-pulse type system operating at 800-MHz (center frequency), which responds to interfaces between materials of differing dielectric permittivity. Optech's Intelligent Laser Ranging and Imaging System (ILRIS-3D) is a laser-based imaging and digitizing system. The ILRIS-3D Lidar is approximately the size of a survey "total station" and has a large dynamic range: from 3 m to more than 1,500m. Prior tests at Haughton Crater have found that lidar can be helpful for characterization and analysis of remote geological formations.

Ground Operations

One objective of the Simulated Lunar Robotic Site Survey at Haughton Crater was to remotely monitor the K10 rovers from JSC. To achieve this objective, IRG personnel set up a simulated "ground control" facility at the JSC Cockpit (see Figure 2) in the Dexterous Robotics Lab (DRL) of the Automation, Robotics, and Simulation Division (Code ER). Remote monitoring and basic vehicle tele-operation was performed from this ground control facility. A priori information about the field site was combined with in-situ rover telemetry and sensor data to support these remote operations. High fidelity monitoring and control of rover activities was performed with approximate lunar communications delays. Tools used from the remote site included visualizing a priori and in situ data, , processing LIDAR and GPR data, and visualizing raw and derived products. Tasks supported ranged from remote driving from JSC cockpit to summarizing robot performance.

FIGURE 2. Haughton Crater Ground Control in the JSC Cockpit.

Summaries of Robot Performance for Ground Operations

We developed a prototype of the software framework for human-robot interaction to support ground operations of the K10 rovers during the Haughton Crater field test. We prototyped a system that (1) monitors data from the K10 rovers performing surveys to determine the depth of permafrost at the Haughton Crater on Devon Island, (2) computes performance measures about how well the survey is going, (3) builds summaries of these performance measures, and (4) notifies appropriate personnel when summaries are complete. We successfully evaluated our prototype using data collected during the Operational Readiness Tests (ORTs) conducted in May and June 2007, and data collected during the first week of the Haughton Crater field test in July 2007. See Figure 3 for an overview of the software architecture for the site survey prototype.

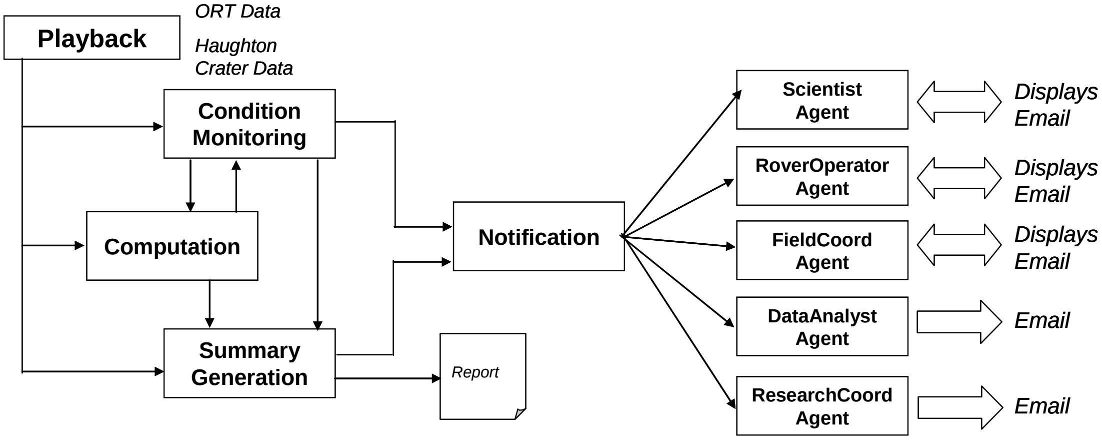


FIGURE 3. Prototype for Haughton Crater Site Survey.

We defined a daily summary of the site survey conducted near Haughton Crater. Two summaries were defined, one for K10 Red and one K10 Black. We identified the computations needed to build a summary for each type of survey and the conditions that should activate these computations. Computations needed to compute performance include distance traveled by the rover, samples collected by the rover, run-time of instrument payloads on the rover, drive-time of the rover, and run-time of the rover. Events affecting the autonomous survey include data dropout and restarts of the rover's onboard software. The summary provides information in the following five areas:

- Overview – identifies the robot, the survey location, and time period,
- Performance with respect to Plan – compares distance traveled and samples taken to the planned distance and number of points,
- Instrument Performance – summarizes number of samples, the instrument run-time, and where possible identifies problems (e.g., bad scans, restarts),
- Robot Performance – summarizes robot's daily and mission performance in terms of distance traveled, run-time, and drive-time, and
- Event Log – details specific events occurring during the survey; events can be nominal or off-nominal.

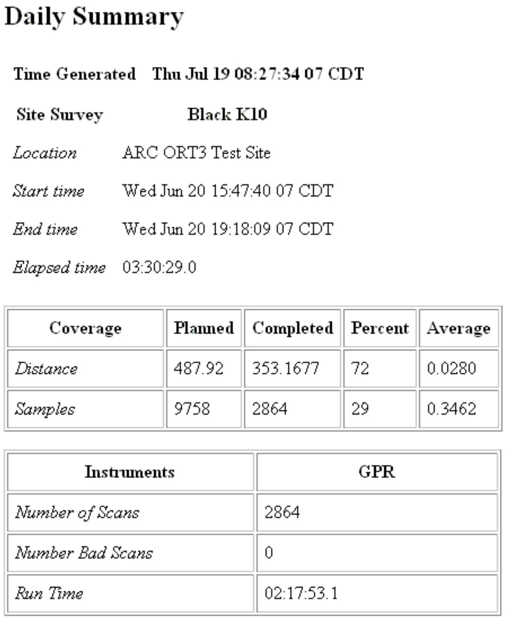
For each computed item in the summary, we defined the required computations, associated them with data, and specified the conditions under which a computation should be activated. For each event in the summary, we identified patterns of data indicating the event had occurred. We encoded libraries of functions to perform these computations and pattern matching. We built XSL stylesheets that describe how to format the XML-formatted computations and events into a HTML summary. One report was defined for each of K10 Black and K10 Red. We used to the open source XSLT software to build these summaries using the summary stylesheets.

Once a summary was generated, it was archived and embedded in a notice. The notice was passed to the software agents associated with the site survey team. We identified the following user roles for the site survey: (1) Research Coordinator, (2) Field Test Coordinator, (3) Data Analyst, (4) Instrument Lead, (5) Rover Engineer, and (6) Rover Operator. In this simulated mission, the Research Coordinator and Data Analyst are ground-based roles subject to bandwidth-constrained, time-delayed satellite communications. The remaining roles are space-based roles, with personnel located in a nearby habitat/vehicle (Intra-Vehicular Activity) with high-bandwidth, low-latency communications. The DCI role-based notification was used to route these summaries to appropriate personnel, based on their roles, and to select the best way to present this information to personnel. For this prototype, we focused on two presentation modalities, computer display and email. We defined notification specs to route the daily summaries based on user roles. For example, the Research Coordinator and Data Analyst receive summaries in email. The Field Coordinator and Rover Operator receive summaries through a computer display if logged into the software framework (i.e., online), and through email if not logged in (i.e., offline). Because of the limited scope of the notices in this evaluation, we chose not to model the roles of Instrument Lead or Rover Engineer. Adding such roles in the future should be straightforward.

We executed this software by playing back data recorded during the ORTs conducted at NASA Ames in May and June, and data recorded during the Haughton Crater Field Test. Overall, we evaluated data from four days of site survey, two from the ORT in June and two from the field test. For each day of the ORT, we build summary files for both K10 Black and K10 Red. For Haughton Crater, we had data from one survey by K10 Black and one survey by K10 Red. Figure 4 shows two tables from a daily summary for K10 Black on Day 1 of the ORT.

RESULTS

During the Haughton Crater field test, we demonstrated the end-to-end capability to monitor rover data, build summaries of rover performance on site survey, and notify operators about them. We identified an initial set of performance measures for autonomous survey and demonstrated the ability to derive them from data exported by the rovers during the field test. These measures include performance with respect to a plan (i.e., distance traveled and samples collected) and percentage of time spent on tasks (i.e., time taking samples compared to total run time). We also identified an initial set of events affecting autonomous survey and demonstrated the ability to detect them. These events included data dropouts and restarts of the robot's onboard software. We demonstrated the feasibility of our planned approach to use XML specifications for associating data items with computations and conditions, and for translating the resulting derivations into customized reports. This capability is central to our approach of providing tools that allow users to define performance measures and configure how they will be informed of them. Finally we demonstrated the ability to integrate this event detection and summarization approach with our existing capability to notify personnel based on their roles. Reports were routed via email to interested users based on an XML specification of their information requirements to fulfill a role.



Daily Summary

Time Generated	**Thu Jul 19 08:27:34 07 CDT**
Site Survey	**Black K10**
Location	ARC ORT3 Test Site
Start time	Wed Jun 20 15:47:40 07 CDT
End time	Wed Jun 20 19:18:09 07 CDT
Elapsed time	03:30:29.0

Coverage	Planned	Completed	Percent	Average
Distance	487.92	353.1677	72	0.0280
Samples	9758	2864	29	0.3462

Instruments	GPR
Number of Scans	2864
Number Bad Scans	0
Run Time	02:17:53.1

FIGURE 4. Example of K10Black Daily Summary.

We also learned that the performance summaries we built can assist data analysis as well as robot supervision. The events and computations that summarize the rover's daily performance also can provide useful summaries of the contents of a data log. For example, these reports can be used to find the data set that contains a particular incident.

Summary events can be used as entry points back into the gigabytes of data requiring offline data analysis. The parameters indicating the event, the time interval during which the event occurred, and the data or images collected are linked to the event. Using this information, it is possible to retrieve the relevant information from archives. These queries can form the basis for further investigation of the event and the situation around it. For such an application, custom summary formats could be defined that monitor salient features of a data set.

RELATED WORK

Similar to the DCI System, the Electric Elves system (Chalupsky, et al., 2001) was designed to support distributed human groups perform their duties. Specifically, it assists professors and students in scheduling meetings, designating speakers, and identifying where to meet for lunch. Also like the DCI System, the Electric Elves was deployed for extended use at the University of Southern California. A key difference, however, is that the agents of the Electric Elves are designed to aid human-human interaction, while the agents of the DCI System assist human interaction with automation.

Our use of software agents that mediate interaction among humans and automated systems is most similar to work by Scerri (2003) and Bradshaw (2001). Scerri has successfully extended the results of the Electric Elves project to aid teams of humans, robots, and agents by dynamically adjusting control responsibility. Bradshaw has investigated human-agent teamwork for NASA’s Personal Satellite Assistant, a robotic assistant designed to aid crew when operating inside a space vehicle. Neither of these approaches, however, provides mediating software agents to aid interaction with autonomous agents, nor do they address such issues as health and performance monitoring that are essential for such interaction with autonomous space systems.

The DCI notification service has developed adaptive strategies for filtering and routing notices to a distributed user group based on the users’ roles in that group (Schreckenghost, Thronesbery, and Hudson 2005). This differs from other notification research that addresses the needs of individuals without considering their group memberships (Horvitz, Jacobs, and Hovel, 1999; Schmandt, et al., 2000). NASA’s BEACON software (Wyatt, et al., 1997) monitors data for events and notifies ground personnel when events are observed. Like our software framework, it uses pattern-matching to detect events. BEACON’s approach to notification, however, does not route and filter notices based on user role. Additionally, BEACON does not adjust presentation of notices depending upon the accessibility of a user and the urgency of the notice. The DCI approach using notice specification most closely resembles Bradshaw’s use of ontology-based notification policies in the KaOS system (Bradshaw, et al., 2003). Commercial notification software such as Stirling Systems Group JobMon and Bear Mountain Software’s Topper address remote notification via paging using static routing specifications. DCI provides for dynamic notice routing by using it’s knowledge of user roles and location to adjust what notices are sent to a user and what notification mechanism is used (e.g., email). Our work on quantifying and summarizing human-robot performance is based on metrics for human-robot interaction (Steinfeld, et al., 2006).

CONCLUSIONS

The evaluation of our software framework for human-robot interaction during the Haughton Crater Field Test proved the feasibility of our approach to health and performance monitoring. The core services of event detection, situation summarization, and notification combine to provide the user with timely information about progress on the survey and operational events impacting this progress. We plan to implement a fully functional version of the core services we prototyped for Haughton Crater. Using this software we will compute performance measures for rovers like K10, and evaluate the usefulness of these measures in support human supervision of robots.

We expect the software framework described in this paper to have general application for remote supervision of robots. In addition to the daily summaries of site surveys created for the K10 rovers, we have used this technology to create summaries for operators remotely supervising the Centaur rover. We constructed incident reports for Centaur using data recorded during the Desert Research and Technology Study (RATS) field test.

We plan to develop tools for users of our software framework. This includes aids for defining new events and summaries in response to changing situations and missions. Additionally we plan to develop visualization software to aid understanding of performance trends, and relate these trends to operational events and the underlying data upon which these events are based.

ACKNOWLEDGMENTS

We acknowledge Mark Allan, Matt Deans, Lorenzo Fluckiger, and Hans Utz for their assistance in integrating the software framework with the K10 rover and providing information about the summaries needed for the site survey. This work was funded under NASA SBIR contract #NNJ07JB31C.

REFERENCES

Bradshaw, J., Sierhuis, M, Gawdiak, Y, Jeffers, R., Suri, N., and Greaves, M. "Adjustable Autonomy and Teamwork for the Personal Satellite Assistant," in *Proc. IJCAI-2001 Workshop on Autonomy, Delegation, and Control: Interacting with Autonomous Agents*, Seattle, WA, 2001, pp. 20-26.

Bradshaw, J., Uszok, A., Jeffers, R., Suri, N., Hayes, P., Burstein, M., Acquisti, A., Benyo, B., Breedy, M., Carvalho, M., Diller, D., Johnson, M., Kulkarni, S., Lott, J., Sierhuis, M., and Van Hoof, R. "Representation and Reasoning for DAML-based Policy and Domain Services in KAoS and Nomads," *AAMAS 2003*. Melbourne, Australia, ACM Press. 2003.

Bresina, J., Jónsson, A., Morris, P., and Rajan, K. "Mixed-Initiative Planning in MAPGEN: Capabilities and Shortcomings," *International Conference on Automated Planning and Scheduling*. Monterey, CA. June 2005.

Chalupsky, H., Gil, Y., Knoblock, C., Lerman, K., Oh, J., Pynadath, D., Russ, T., and Tambe, M. "Electric Elves: Applying Agent Technology to Support Human Organizations," *Proc. Innovative Applications of Artificial Intelligence*, Seattle, WA, 2001, pp. 51-58.

Fong, T., Bualat, M., Edwards, L., Flueckiger, L., Kunz, C., Lee, S. Y., Park, E., To, V.,Utz, H., Ackner, N., Armstrong-Crews, N., and Gannon, J. "Human-Robot Site Survey and Sampling for Space Exploration," *AIAA-2006-7425*. Proc. AIAA Space 2006, San Jose, CA. 2006.

Fong, T., Deans, M., Lee, P., and Bualat, M., "Simulated Lunar Robotic Survey at Terrestrial Analog Sites," *Proc. 38th Lunar and Planetary Science Conference*, Abstract 1487, Houston, TX. 2007.

Hambuchen, K., Bluethmann, W., Goza, M., Ambrose, R., Rabe, K., and Allan, M. "Supervising Remote Humanoids Across Intermediate Time Delay," *2006 6th IEEE-RAS International Conference on Humanoid Robots*. Dec. 2006. pp. 246-251

Horvitz E., Jacobs A., and Hovel D. 1999. "Attention-sensitive Alerting," *Proceedings of UAI '99*, Stockholm, Sweden. Morgan Kaufmann, 305-313 Jul 1999.

Mishkin, A., Lee, Y., Korth, D., and LeBlanc, T. "Integrated Human-Robotic Missions to the Moon and Mars: Mission Operations Design Implications," *Proc. IEEE Aerospace*. 2007.

Scerri, P., Pynadath, D., Johnson, L., Rosenbloom, P., Si, M., Schurr, N., and Tambe, M. "A Prototype Infrastructure for Distributed Robot-Agent-Person Teams," in *Proc. 2nd International Conference on Autonomous Agents and Multi-Agent Systems*, Melbourne, Australia, 2003, pp. 433-440.

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. publisher Taylor and Francis. pp. 361-369.

Schreckenghost, D., Thronesbery, C., and Hudson, M. B. "Situation Awareness of Onboard System Autonomy," *The 8th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Munich, Germany, September, 2005.

Schmandt C., Marmasse N., Marti, S., Sawhney, N., and Wheeler, S. "Everywhere Messaging," *IBM Systems Journal* 39(3&4) published by IBM Corp., 660-677 2000.

Sheridan, T., *Telerobotics, Automation, and Human Supervisory Control*, published Cambridge, MA: MIT Press, 1992.

Steinfeld, A., Fong, T., Kaber, D., Lewis, M., Scholtz, J., Schultz, A., and Goodrich, M. "Common Metrics for Human-Robot Interaction," *Proceedings of ACM Conference on Human-Robot Interaction*, Salt Lake City, UT. 2006.

Wyatt, E. J., Hotz, H., Sherwood, R., Szijjarto, J., and Sue, M. "An Overview of the Beacon Monitor Operations Technology," *International Symposium on Artificial Intelligence, Robotics, and Automation in Space*, Tokyo, Japan. 1997.