

Evolving from Planning and Scheduling to Real-Time Operations Support: Design Challenges

Jessica J. Marquez^{*}, Melissa Ludowise[†], Michael McCurdy[‡]
NASA Ames Research Center, Moffett Field, CA, 94035

and

Jack Li[§]
San Jose State University Research Foundation, San Jose, CA, 94035

Versions of Scheduling and Planning Interface for Exploration (SPIFe) have supported a variety of mission operations across NASA. This software tool has evolved and matured over several years, assisting planners who develop intricate schedules. While initially conceived for surface Mars missions, SPIFe has been deployed in other domains, where people rather than robotic explorers, execute plans. As a result, a diverse set of end-users has compelled growth in a new direction: supporting real-time operations. This paper describes the new needs and challenges that accompany this development. Among the key features that have been built for SPIFe are current time indicators integrated into the interface and timeline, as well as other plan attributes that enable execution of scheduled activities. Field tests include mission support for the Lunar CRater Observation and Sensing Satellite (LCROSS), NASA Extreme Environment Mission Operations (NEEMO) and Desert Research and Technology Studies (DRATS) campaigns.

Nomenclature

<i>CE</i>	=	Constraint Editor
<i>DRATS</i>	=	Desert Research and Technology Studies
<i>ISS</i>	=	International Space Station
<i>LCROSS</i>	=	Lunar CRater Observation and Sensing Satellite
<i>LER</i>	=	Lunar Exploration Rover
<i>MAPGEN</i>	=	Mixed-Initiative Activity Plan Generator
<i>MER</i>	=	Mars Exploration Rover
<i>MSL</i>	=	Mars Science Laboratory
<i>MSLICE</i>	=	MSL Interface, instantiation of SPIFe for Mars Science Laboratory mission
<i>NEEMO</i>	=	NASA Extreme Environment Mission Operations
<i>PSI</i>	=	Phoenix Science Interface, instantiation of SPIFe for Phoenix Mars Lander mission
<i>Score</i>	=	Instantiation of SPIFe for ISS mission operations
<i>SPIFe</i>	=	Scheduling and Planning Interface for Exploration

I. Introduction

Over the course of seven years, the NASA Ames Research Center has been actively involved in researching and advancing the state-of-the-art in planning and scheduling tools for NASA mission operations. Central to this is the planning toolkit named Scheduling and Planning Interface for Exploration (SPIFe), which integrates a variety of essential elements such as timelines, automated planning, and resources modeling. The design and development of

^{*} Human Systems Integration Division, NASA Ames Research Center, M/S 262-2, Moffett Field, CA 94035, and AIAA Member.

[†] Human Systems Integration Division, NASA Ames Research Center, M/S 262-4, Moffett Field, CA 94035.

[‡] Human Systems Integration Division, NASA Ames Research Center, M/S 262-4, Moffett Field, CA 94035.

[§] Human Systems Integration Division, NASA Ames Research Center, M/S 262-4, Moffett Field, CA 94035.

SPIFe is driven by human-computer interaction principles, enabling work in a rich prototyping environment where design solutions can be tested for an evolving operational tool while still supporting multiple flight projects across the agency. New SPIFe features are developed and integrated as this software tool is introduced to different domains and to a diverse set of users. Recently, users have requested real-time operations support. This presents a unique design challenge, as SPIFe has primarily been used for planning and scheduling, a task that normally takes place before execution. This paper describes ongoing efforts to understand different user needs for real-time operational support and how best to leverage the existing foundation of planning tools.

II. Background

SPIFe (pronounced “spiffy”) was originally designed in the context of Mars tactical planning, where users scheduled a set of predetermined activities. Versions of SPIFe have supported Mars robotic missions since 2003, including the Mars Exploration Rovers (MER) and Phoenix Mars Lander¹. An instantiation of SPIFe is tailored to each planetary mission. The MER mission used earlier tools such as the Mixed-Initiative Activity Plan Generator (MAPGEN) and the Constraint Editor (CE), the fundamental concepts of which gave rise to the SPIFe toolkit. The Phoenix Mars Lander was the first mission to use a dedicated adaptation of SPIFe called the Phoenix Science Interface (PSI). SPIFe-based planning tools are also baseline for use on the future robotic mission Mars Science Laboratory (MSL). For these Mars missions, each toolkit has supported robotic planning and scheduling of the day-to-day operational and scientific activities as well as validation of complex robotic plans.

Tactical planning for the Mars domain involves a planner creating a daily schedule that accommodates all the mission constraints and a variety of competing goals, from engineers and scientists, within the time frame that is allowed to generate a plan for the following mission day. A schedule is a collection of activities planned to occur at predetermined times (Figure 1). Within SPIFe, each activity has associated constraints and utilizes resources, which are modeled within the system. Let’s take, for instance, the simple activity of a robotic rover taking a picture. There are many parameters associated with this activity, such as the type of camera to be used and the filters needed for that picture. In order to take the picture, the rover needs to be stationary and the image must be taken at a particular time of day. These would be the constraints associated with the activity. Taking a picture also requires the consumption of some resources. For this activity, the utilized resources are time, power consumed and data stored (how long, how much power, and how much data it takes to capture that image). The planner has to integrate many different types of activities and constraints without violating any of the flight rules, like exceeding power or data limits. SPIFe supports this complex task of creating intricate, mission-critical schedules.

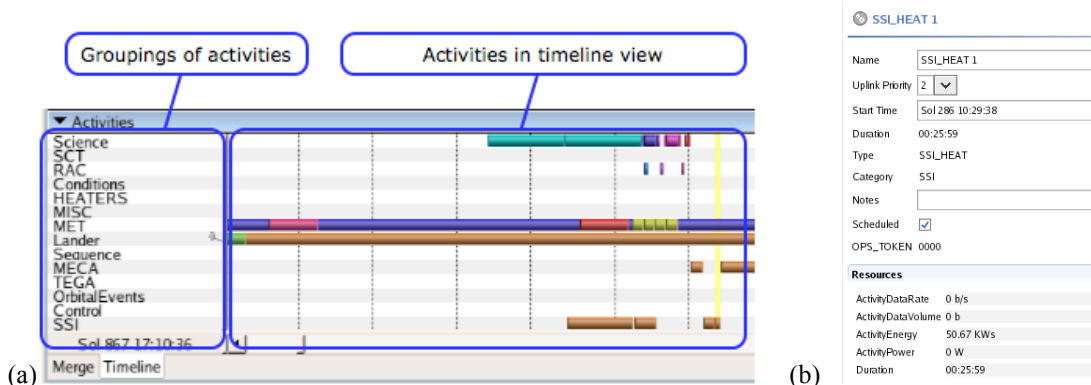


Figure 1. (a) Basic schedule and (b) activity detail from Phoenix Science Interface.

Subsequent instantiations of SPIFe have been used for other domains, including those where humans were the primary executors of the scheduled activities. The Flight Analogs Project Bedrest Study used SPIFe to help schedule research participants. Plans covered about two weeks of operations for the General Clinical Research Center facility conducting the microgravity analog experiments. Meals, tests, and countermeasure activities had to be scheduled for a maximum of ten participants, with corresponding personnel support and available equipment². More recently, planning for the NASA Extreme Environment Mission Operations (NEEMO) mission has used SPIFe to plan their two-week long underwater mission. The schedule for six people, four main crewmembers and two habitat technicians, includes performing experiments, underwater dives (analogous extravehicular activities), and daily

activities, such as sleep and meals³. Currently, NASA Ames Research Center and NASA Johnson Space Center are collaborating to develop the next generation planning and scheduling toolkit to support human spaceflight mission operations. For International Space Station (ISS), operations planners have to plan weekly schedules for up to six crewmembers. This instantiation of SPIFe is named Score. Astronauts have to accomplish a wider variety of activities, from ISS maintenance tasks to science experiments to public media & education events. Furthermore, planners have to integrate the needs of other space agencies: the Russian Federal Space Agency, the European Space Agency, and the Japanese Aerospace Exploration Agency.

Regardless if planning for robots, spacecrafts, or people, the task of planning is difficult and intricate. SPIFe does connect to a robust planning engine, where a complex network of planning constraints can be defined and automated. However, SPIFe is not a highly automated planning system, i.e., providing an automatically generated schedule. One key finding from the Mars experience was that human-machine interaction issues play a significant role in determining the utility of planning and scheduling software. Thus, SPIFe users leverage automated assistance to facilitate in the task of planning. For instance, automated assistance helps planner satisfy constraints and flight rules, yet the planner still has the flexibility to adapt to changing circumstances that are not automated.

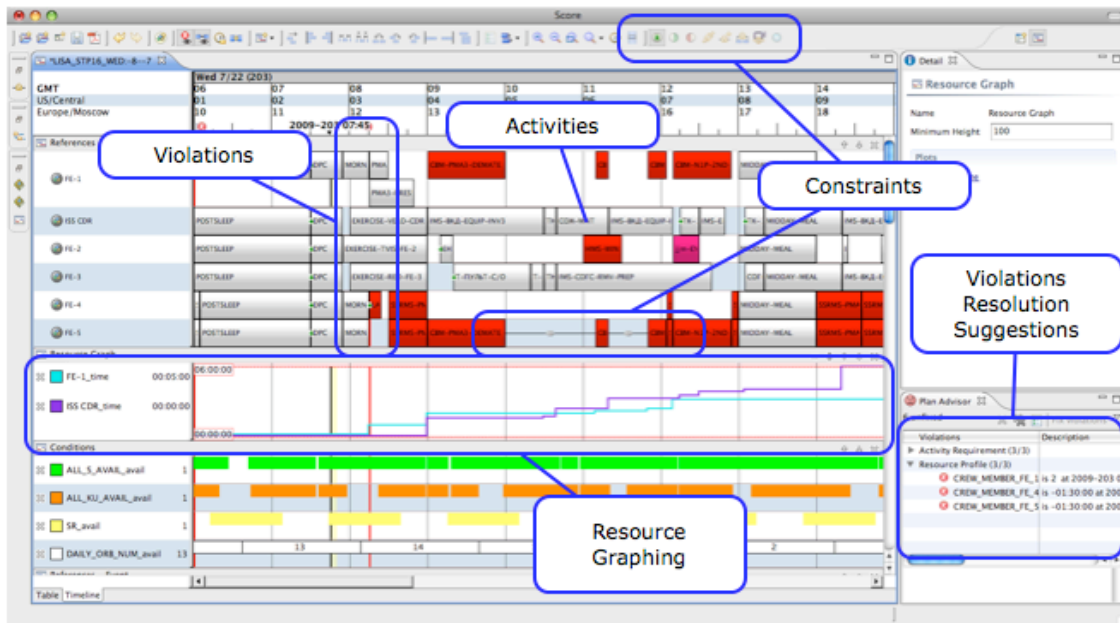


Figure 2. Screenshot of SPIFe (Score instantiation) with some key features highlighted.

While there are differences between planning domains, scheduling activities with SPIFe remains essentially the same. Planners still need to create a set of feasible activities, organize them in a timeline, satisfy the constraints and flight rules of the domain, and integrate and balance competing needs from a variety of stakeholders. For this reason, many of the planning features that SPIFe maintains, remain fundamental components of the toolkit and successfully support the task of complex planning and scheduling across domains. Users have identified these features, through usage and feedback, as key and desirable SPIFe elements. Some of these are listed below and illustrated in Figure 2:

- *Multiple perspectives to view information:* activity information can be viewed individually or collectively in either a tabular or timeline manner. The timeline perspective allows for many types of groupings for the same set of activities.
- *Resource modeling and graphing:* modeled resources being utilized by activities are calculated and can be displayed in a graphical manner alongside activities. Presenting an integrated view of both allows users to deduce relationships between activities and resources.
- *Ease of constraint modeling:* users can easily insert common timing constraints, such as sequential ordering of activities, time critical events, and required time gaps. This increases the users' capability of introducing complex relationships between activities that may not have otherwise been predetermined.
- *Automatic constraint violation checking and resolution:* plan violations, be it those that are user defined or modeled beforehand, are displayed, alerting the user to their existence. Suggested resolutions and suspension of constraints assist users in creating error-free plans.

- *Plan manipulation shortcuts*: many manipulation features, like undo & redo, copy cut, & paste, are standard. Other more specific features, like eliminating time gaps between activities, swapping activities, or align activities, facilitates planners' job.

III. Real-Time Operations Support Challenges

Over time, SPIFe has matured within the domain of tactical planning and scheduling, i.e., planning activities for the near future. However, with the inclusion of new domains, users have requested in addition, leveraging SPIFe during plan execution. Hence, recent design efforts have focused on the development of new features that better support real-time operations. The challenge with this has been two-fold. First, how do we adapt a tool that has primarily been designed and used for *planning and scheduling* to support real-time operations? Previous domains allowed planners anywhere from hours to days to appropriately schedule all the activities and verify an error-free plan. Even when re-planning is necessary, the changes are relatively small and can be quickly handled. Many of the SPIFe features, such as modeling of activities, constraints, and resources, are used for planning. Planners were training to use SPIFe, including many hours of experience with the interface to understand the vast range of capabilities of the tool. The needs for real-time operations are different. In the context of real-time execution, a plan is being followed, not created nor modified. The end users are more diverse, no longer just the trained planners, but people that are either monitoring or executing off the plan. The interface must communicate which activities are happening at the current time and the rationale behind the schedule. Additionally, the interface needs to help the user understand the future state of events and how changes now will affect the future. New SPIFe features have been incorporated in order to start addressing these differences between planning and real-time operations support.

The second challenge is how to go about clearly identifying *all* the needs for real-time operations support. Our approach to better understanding the needs for real-time operations support is based on the same human-computer interaction principles that have always guided the design of SPIFe. Principally, SPIFe is gradually being introduced to another phase of operations, and to distinct new set of users. Through contextual inquiry techniques⁴ and immersion in the domain, the user needs will emerge and new capabilities will be implemented that support real-time operations. By field-testing these features across various domains, much like we have done with planning features, they will evolve, mature and become integral components of SPIFe.

The first real-time use of SPIFe on a flight project was during the Lunar CRater Observation and Sensing Satellite (LCROSS, 2009) mission. SPIFe will also be used for 2010 NEEMO and Desert Research and Technology Studies (DRATS) campaigns. For LCROSS, SPIFe was not used for planning and scheduling directly but rather as a tool to show the dispersed ground controllers the timeline, providing shared situation awareness of current and upcoming events. LCROSS was a unique opportunity to participate in a spacecraft mission with a relatively short mission lifetime⁵. The mission goal was to confirm the presence or lack of water ice on the Moon by impacting a shadowed crater. The plans were relative stable (once a trajectory was fixed, no major re-planning was necessary), imported into SPIFe, and displayed in the timeline on a large screen.

For NEEMO, a detailed two-week plan has been prepared for the aquanauts (underwater crew). They need to follow the plan in their underwater habitat, leveraging the interface to track completion of activities and linking to procedures. During part of the mission, expected to occur in May 2010, the crew will simulate having degraded communications with ground control, and they will be responsible for managing their own activities and scheduling additional events. In September 2010, DRATS will use Score to both plan and view the schedule. Currently, plans are being created for the Earth-analog mission. Like LCROSS, the end users are striving to achieve common situation awareness between the crew in the Lunar Exploration Rover (LER) and mission controllers. While the finalized operational concept is still being refined, it is expected that the schedule will be visible within the LER, which serves as a habitat for the simulated planetary exploration mission.

IV. Real-Time Operations Support Features

This section details the real-time operations features that have been implement for SPIFe over the course of two years to support the variety of missions previously delineated. Most of these features have been used in an operational setting (though NEEMO and DRATS field tests have yet to occur at the time this went to press).

A. Current Time Indicators

One of the first identified needs for supporting real-time operations is the ability for the user to quickly see current time, allowing a variety of end users to monitor the schedule and the corresponding, current activities. To this end, SPIFe has a Marcus Baines line, a red, vertical line in the timeline view that shows the user the current time (Figure 3). Additionally, the user can toggle “follow along”, which keeps the Marcus Baines line visible and the

current activities are selected. Another current time indicator implemented is called “Elapsed Table” (Figure 4). The table lists all activities, highlighting current activities, labeling them with “Now”. This table also shows the time remaining until future activities, and the time past since the previous activity. The start and end times for activities are also listed alongside. The “Elapsed Table” aims to assist users in not only following activities concurrently happening, but also to project and prepare for upcoming activities.

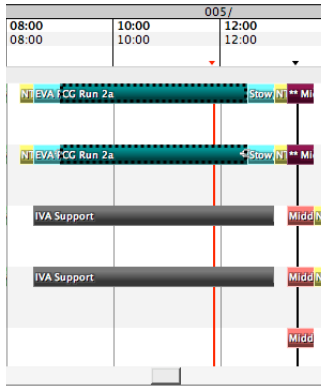


Figure 3. Current time shown as vertical, red line.

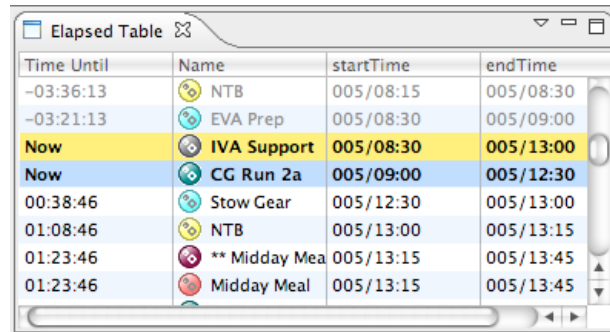


Figure 4. Elapsed Table, highlighting current activity and time until next activities.

Finally, a simple addition to SPIFe was to create a clock view. The clock shows current or simulated time in several time zones including mission-specific elapsed time. Having a large, visible display of current time adjacent to the plan helps support shared situation awareness.

These current time indicator features were well received by end-users. In particular, these supported LCROSS mission operations from start to finish. It appears that these items are desirable, perhaps even essential, for future use in real-time operations support.

B. Enabling Execution

Supporting real-time operations includes helping people work “from” the plan, using it as a guide for when to perform which activities. Two initial features have been included in SPIFe to better enable users to leverage the plan. Activities have two new imbedded parameters: a link to procedures and an execution status (Figure 5). After selecting an activity, the user can through the details view, open a link to the associated procedures, directly connecting how to execute planned activities with the activity itself. Additionally, any execution (from the planner) and operations (from the executioner) notes are embedded with each activity.

The user also can change the status of activities through the execution parameter. The activity’s status can be set to pending, active (i.e., currently executing), paused, deferred, and completed. By clicking the “play” button, the user enables a timer, which concludes when the “stop” button is pressed. In this manner, how long that activity took to execute is saved with its associated event. Collecting execution durations for all activities may be valuable information for future planning, resulting in more efficient plans.

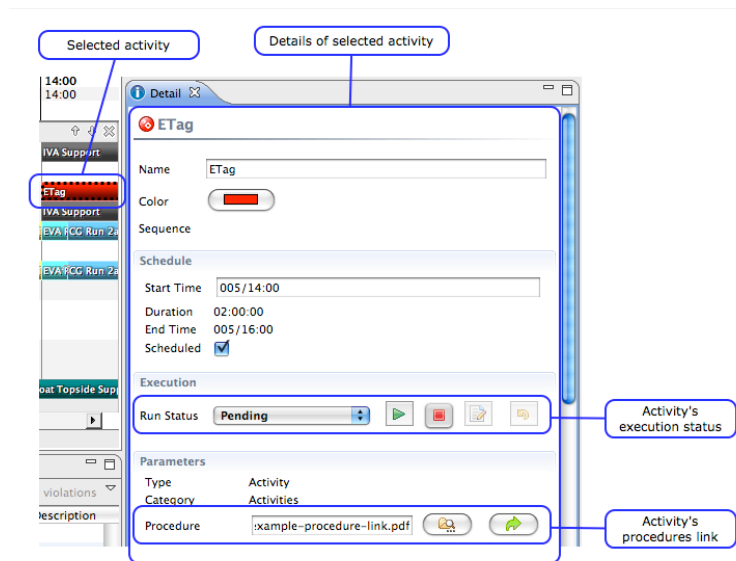


Figure 5. Activity selected with links to procedures & execution status.

C. Planning Features Supporting Real-Time

While the previous descriptions elaborated mostly on new SPIFe features, it is worth noting that there are planning features that can support real-time operations. For instance, activities with user-defined time constraints are called “pinned”, i.e., the activity is “pinned” to a particular time. This time constraint is displayed in the interface with an icon annotating the activity, communicating to the user that this activity must be executed at a specific time. Leveraging the user-defined constraint tools and the modeled constraints, the planner can have a very rich model of constraints imbedded into the schedule. Viewing this information is helpful during last minute re-planning under time pressure (though care must be taken to not overwhelm the user). Activity manipulations, e.g., reschedule or deletion of activity, that create plan violations are automatically flagged, providing instantaneous feedback that an invalid change has occurred. Furthermore, quick suggestions to resolve violations are accessible through tooltips and the plan advisor (Figure 6). Another feature that could be used during these unexpected re-planning events is “constrained move.” When this mode is enabled, the tool prevents activity manipulations that would create plan violations. This means that the user could reschedule an activity *only* if it still falls within the modeled constraints. For example, an activity could not be placed in an invalid order or start too early nor late *if* the appropriate constraints have been modeled and incorporated into the schedule.

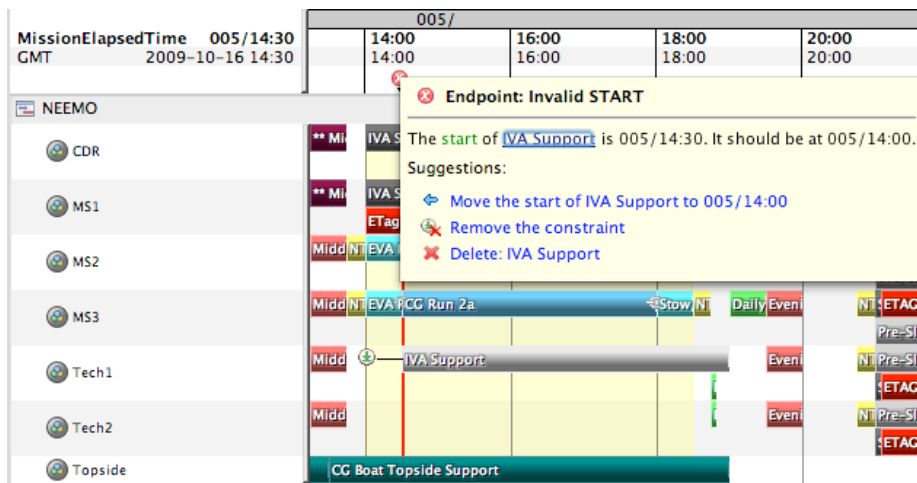


Figure 6. SPIFe timeline showing an activity with a time constraint, a violation and corresponding suggested resolutions.

V. Conclusion

By gradually increasing SPIFe’s real-time operations features through user experiences in different domains, this interface, originally designed for planning and scheduling, will better support execution of missions. Essential user feedback will be gained from assisting in analog missions. One key feedback will be which planning features are not practical during real-time operations. After NEEMO 14 (May 2010) and DRATS (September 2010), future work includes continuing development of Score with the ISS mission operations planners. Their planning and scheduling necessitate a great variety of software tools, like On-board Short Term Plan Viewer (OSTPV), from which we can learn successful best practices. One relevant aspect of these users’ needs is that they have to plan schedules that range from six months (strategic planning) to day-of (tactical planning). However, software tools that support this have competing requirements, and hence, planning flexibility that supports all time scales, including real-time operations, will further drive development of new SPIFe features. How this flexibility is incorporated will remain as our future design challenge, to both provide users with a powerful planning system without overwhelming the various end-users.

NASA Ames Research Center remains on the forefront of state-of-the-art planning systems by supporting a wide-range of space mission operations. This not only contributes to NASA’s goals, but also is of interest to the planning community, which ranges from air traffic control to resource deployment. Developing software systems for planning and scheduling continues to be a research topic because this task is so complex and requires a careful balance between people and automation. Integrating with real-time operations in recent missions has resulted in the design and integration of new features that support this aspect of mission operations. Positive feedback has been received on the current time indicators and users appear to approve of the planning features that communicate

violations. By addressing these diverse needs of our end users, SPIFe is becoming a more flexible planning aid for current and future space exploration missions.

Acknowledgments

Authors would like to thank the developers that supported this particular work, including Arash Aghevli, Andrew Bachmann, and Eugene Turkov. Authors would also like to acknowledge the support and feedback provided by the ISS Operations Planners group within NASA JSC Mission Operations Directorate (MOD), and the staff and participants of NEEMO 14, DRATS 2010, and LCROSS. This work was sponsored by the NASA Behavioral Health & Performance, Space Human Factors Engineering, and MOD Next Generation Planning System projects.

References

¹Aghevli, A.; Bachmann, A.; Bresina, J.; Greene, K.; Kanefsky, B.; Kurien, J.; McCurdy, M.; Morris, P.; Pyrzak, G.; Ratterman, C.; Vera, A.; and Wragg, S. "Planning Applications for Three Mars Missions with Ensemble." *International Workshop on Planning and Scheduling for Space*. Baltimore, MD, 2006.

²McCurdy, M., Li, J. and Ludowise, M. "Planning and Scheduling Software for Bed Rest Operations: Lessons Learned Toward a Crew Scheduling System for Exploration." Poster at Human Research Program Workshop, Houston, TX, January 2008.

³Marquez, J.J., McCurdy, M., Li, J., and Ludowise, M. "Planning and Scheduling Tool for NEEMO 14 - A Simulated Space Environment: Developing Real-Time Operations Support through Crew Scheduling Systems." Poster at Human Research Program Workshop, Houston, TX, January 2010.

⁴Beyer, H. and Holtzblatt, K. "Contextual Design: Defining Customer-Centered Systems." San Francisco, CA: Morgan Kaufmann Publishers, 1998.

⁵Lunar CRater Observations and Sensing Satellite (LCROSS) Project Site. [online] <http://lcross.arc.nasa.gov> [accessed February 2010]