Leveraging Multiple Artificial Intelligence Techniques to Improve the Responsiveness in Operations Planning: ASPEN for Orbital Express

Russell Knight, Caroline Chouinard, Grailing Jones, Daniel Tran

■ *The challenging timeline for DARPA's Orbital Express mission demanded a flexible, responsive, and (above all) safe approach to mission planning. Mission planning for space is challenging because of the mixture of goals and constraints. Every space mission tries to squeeze all of the capacity possible out of the spacecraft. For Orbital Express, this means performing as many experiments as possible, while still keeping the spacecraft safe. Keeping the spacecraft safe can be very challenging because we need to maintain the correct thermal environment (or batteries might freeze), we need to avoid pointing cameras and sensitive sensors at the sun, we need to keep the spacecraft batteries charged, and we need to keep the two spacecraft from colliding ... made more difficult as only one of the spacecraft had thrusters.*

Because the mission was a technology demonstration, pertinent planning information was learned during actual mission execution. For example, we didn't know for certain how long it would take to transfer propellant from one spacecraft to the other, although this was a primary mission goal. The only way to find out was to perform the task and monitor how long it actually took. This information led to amendments to procedures, which led to changes in the mission plan. In general, we used the ASPEN planner scheduler to generate and validate the mission plans. ASPEN is a planning system that allows us to enter all of the spacecraft constraints, the resources, the communications windows, and our objectives. ASPEN then could automatically plan our day. We enhanced ASPEN to enable it to reason about uncertainty. We also developed a model generator that would read the text of a procedure and translate it into an ASPEN model. Note that a model is the input to ASPEN that describes constraints, resources, and activities. These technologies had a significant impact on the success of the Orbital Express mission. Finally, we formulated a technique for converting procedural information to declarative information by transforming procedures into models of hierarchical task networks (HTNs). The impact of this effort on the mission was a significant reduction in (1) the execution time of the mission, (2) the daily staff required to produce plans, and (3) planning errors. Not a single misconfigured command was sent during operations.

ISSN 0738-4602

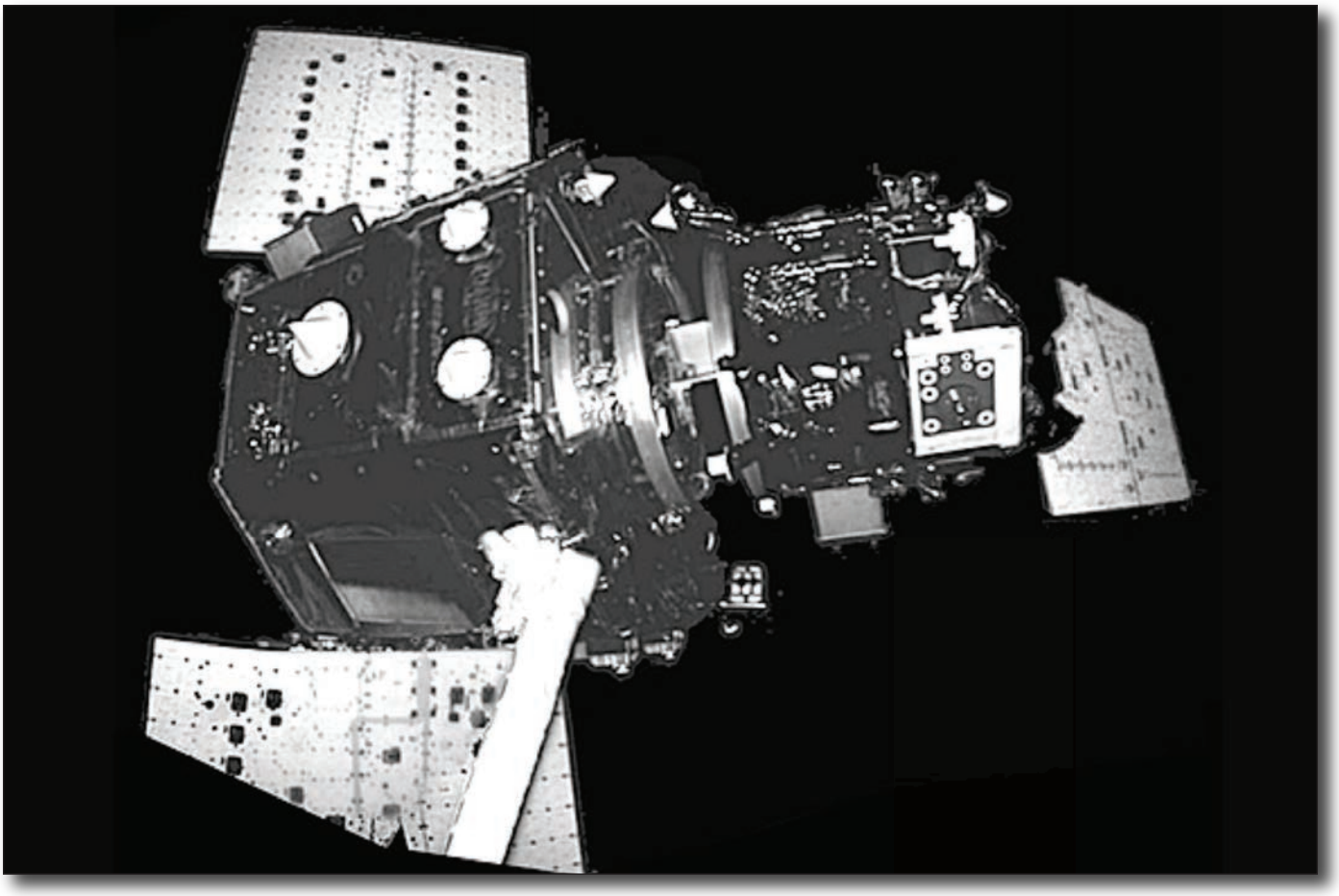

Figure 1. The ASTRO and NextSat Spacecraft, on Orbit Without the Separation Ring.

It is often the case that new technology for space needs to be taken out of the lab and proven in space. The primary goal of a technology mission is to prove the capabilities of newly developed space technology. Most technology missions have the challenge of discovering the limits and capabilities of new systems, and the Defense Advanced Research Projects Agency's (DARPA) Orbital Express (OE) mission was no exception. Orbital Express launched March 8, 2007, and decommissioned on July 22, 2007. The Orbital Express mission demonstrated on-orbit servicing of spacecraft, including rendezvous, transfer of battery and CPU modules, and transfer of propellant, the actual duration of each being approximated but not known to high enough fidelity to commit to a communications and operations plan.

This introduced challenges because pertinent information needed for planning was not available until the various technology experiments were performed, but to perform these experiments we needed to submit plans to reserve communications resources, configure execution scripts, and monitor execution. We note that it is often the case that bounds of performance are known a priori. Our automated planning system leveraged this information to help address the inherent uncertainty in experiment planning.

The Orbital Express planning cycle consisted of a long-term plan (planned four weeks in advance) and a short-term plan (planned the day before operations). The long-term plan was produced before any of the experimental/unknown information was learned. The role of the long-term plan was to ensure that enough resources are reserved ahead of time. Planning at this point required accommodation of the bounds of possible execution. Traditionally, this level of planning is relatively conservative. For example, if we thought it likely that a propellant transfer would require 20 minutes, but could take up to an hour, we would plan for the whole hour. We also couldn't count on getting all of the communications passes we asked for, so we would plan for 20 percent of them to be dropped, and thus ask for more than we needed.

The short-term plan was produced immediately before execution, and some of the experimental or

Figure 2. The Ejected Separation Ring.

NextSat (top) is at the end of the robotic arm (upper left).

unknown information is known and should be integrated. Also, we knew with certainty which communications passes we would be granted. This allowed us to free resources and helped us to reduce cost or reduce the risk of other missions using the shared resources.

Mission planning had many challenges. First, there was a large degree of uncertainty in the execution of our tasks, such as, pumping propellant; second, the procedures for operations were changing within hours of the time that we had to deliver our plans for execution; third, we had to predict how the on-board execution system would behave so that we could accommodate its behavior, even though we were not planning for it explicitly; and fourth, the problem of building a plan even for single day was monumental: hundreds of constraints needed to be checked for thousands of actions, and all needed to be coordinated with procedures that were being developed: building a plan by hand was simply infeasible, regardless of staffing.

Technologies that address each of these challenges were leveraged in developing the Orbital Express ground-planning system are schema-level uncertainty reasoning (for long-term planning), procedure parsing for model generation (for short-term planning), procedural to declarative model translation, and, most importantly, automated planning and scheduling.

DARPA's Orbital Express mission demonstrated on-orbit servicing of spacecraft. Servicing spacecraft has a great potential to increase the lifespan of these exceedingly expensive assets, but the complexity involved in servicing a spacecraft on orbit had been overwhelming. Consider that all spacecraft in low Earth orbit will fall to Earth unless they expend propellant to stay in orbit. If we could pump more propellant into these, we would be giving them new life. Add to this the potential of replacing modules, such as batteries or central processing units (CPUs), then the value of on-orbit servicing becomes clear. Two spacecraft were flown: Boeing's Autonomous Space Transport Robotic Operations (ASTRO) spacecraft (see figure 1), whose role was that of a doctor. ASTRO had a robotic arm, replacement modules (battery and CPU), a propellant pumping system, and a capture device (used to grasp other spacecraft and lock in the propellant pumping mechanism). Ball Aerospace's Next Generation Serviceable Satellite (NextSat) spacecraft (see figures 1 and 2) had the role of a patient. NextSat could receive propellant and modules, but it couldn't maneuver because it had no thrusters. It was up to ASTRO to perform all of the maneuvering. Experiments included rendezvous and capture, fluid propellant transfer, and on-orbit repair.

The mission planning team was divided into two units, the rendezvous planners who concerned themselves primarily with computing the locations and visibilities of the spacecraft, and the scenario resource planners (SRPs) who were concerned with assignment of communications windows, monitoring of resources, and sending commands to the ASTRO spacecraft. The SRP position was staffed by Jet Propulsion Laboratory (JPL) personnel who used the Activity Scheduling and Planning Environment (ASPEN) planner-scheduler.

We discuss the Orbital Express domain in the context of ASPEN, the technologies added to the planner to accommodate the mission objectives, and the ground operations of long-range and daily planning for the mission.

Mission Operations Planning

The OE mission domain required fast turnaround of heterogeneous, dynamic plans. Every day was a different scenario, where communication passes could be lost within hours of the daily planning delivery deadline. Procedures could change within hours of the delivery deadline because the impacts of on-orbit experiments on spacecraft resources (energy and memory) were to a significant extent unknown. Limited available communications existed using primarily the high-bandwidth ground-based Air Force Satellite Control Network (AFSCN) sites, while the relatively low-bandwidth GEO-Synchronous spaceborne tracking and data relay satellite system (TDRSS) communications could potentially vary by the hour. The main difference between AFSCN and TDRSS is

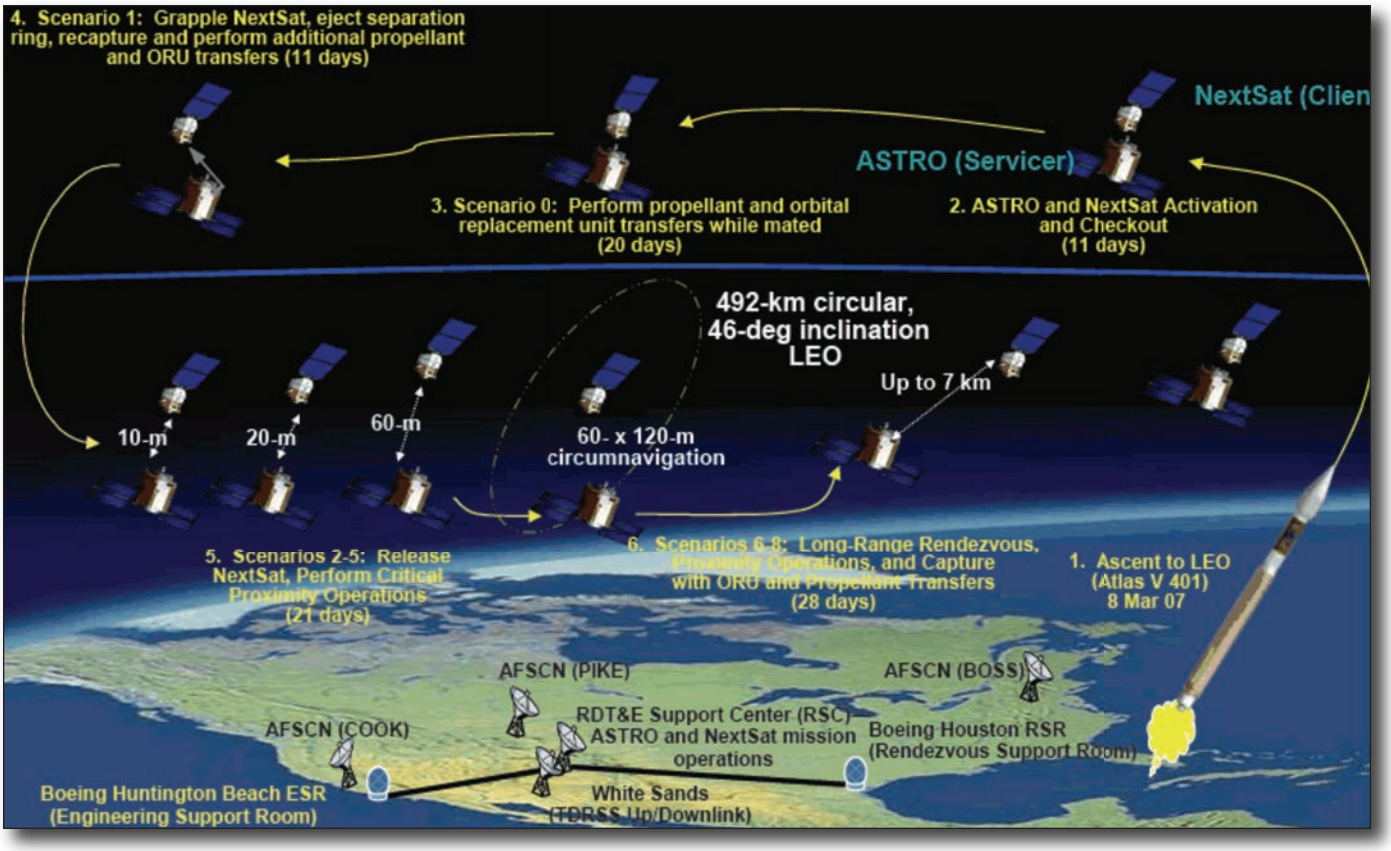

Figure 3. The Orbital Express Scenarios: Increasing Autonomy and Complexity.

that AFSCN sites are ground-based and are only in view for about 5 minutes (when in low Earth orbit, as ASTRO and NextSat were), but TDRSS is satellite-based and is pretty much always in view. The challenge is that these resources are shared across many missions, and each mission has to reserve time in advance. But events can cause resources to become available or to drop out, so it is often the case that we need to be able to respond to these changes quickly.

A scenario (see figure 3) typically consisted of a series of procedures, each of which was written by the operations personnel in Microsoft Word table format. Each procedure listed its steps and associated durations and represented the need for a contact and the type of contact desired or required. Several procedures had other embedded procedures and some spanned more than one day. As an example, the unmated scenario required an initial setup procedure, then the unmated procedure would be kicked off; the de-mate, hold, and re-mate would execute, and then a postrendezvous and capture transfer procedure would be planned. See figure 4 for images of the unmated scenario midexecution in the de-mated configuration and in the final stages of berthing to the mated state.

The schedule of each scenario was dependent on what had been accomplished to date, as the goal of each scenario was to become increasingly more autonomous. The planning schedule was also dependent on the state of the flight system, the amount of preparation time needed before execution, and resources available on future dates. Calendar planning was done by a combination of inputs from flight directors, mission managers, project management, and DARPA.

Procedures were delivered to the SRP and copied to Excel. An ASPEN model-generation script was then used to create ASPEN Modeling Language (AML) representations of the procedures. Once the AML model existed for a procedure, the ASPEN tool read the AML description of the procedure and could be used to add any number of different procedures in a plan required to make up the scenario. See the data flow diagram and the final daily plan in figure 5.

Roles of Mission Planning

Mission planning had two primary roles for Orbital Express: (1) evaluate scenarios for feasibility early in the design of the mission, and (2) provide responsive communications and commanding planning and scheduling during the mission. To serve both roles,

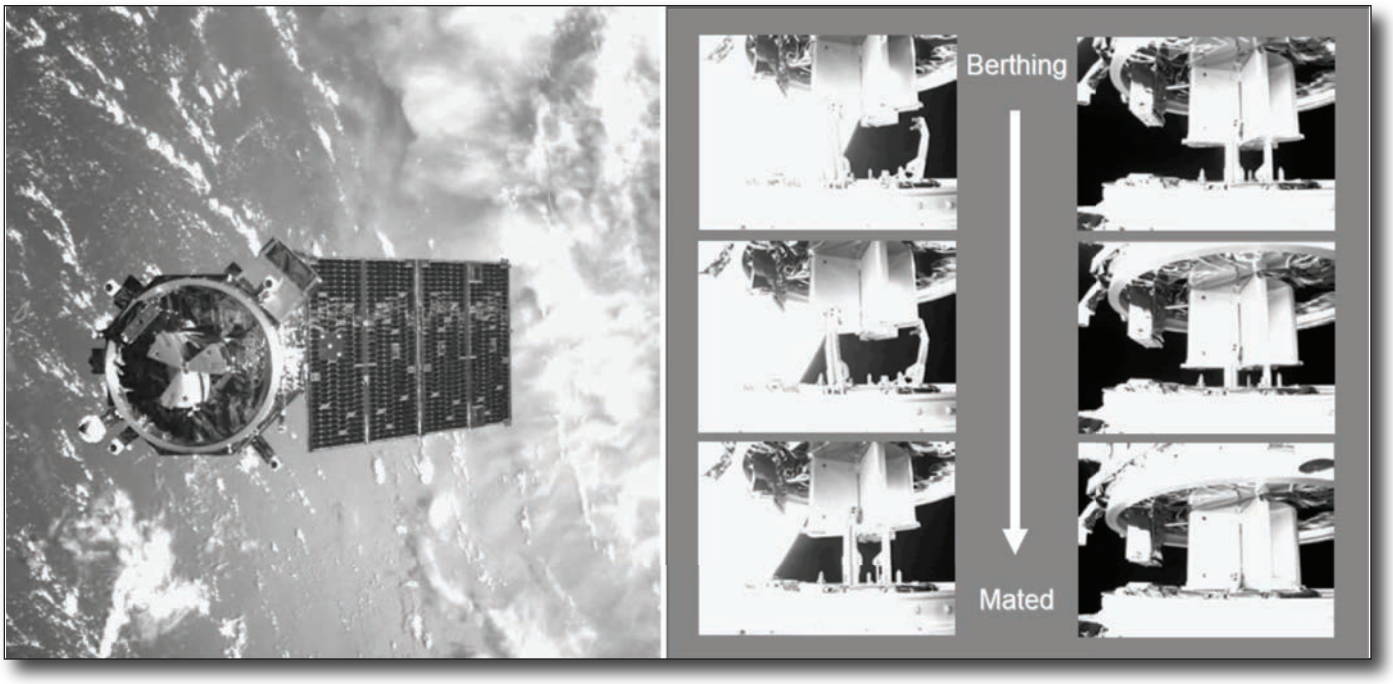

Figure 4. A Demate/Mate Scenario.

NextSat is 14m away during a departure, then progressive side view configurations of the "Berthing" to "Mated" states are shown.

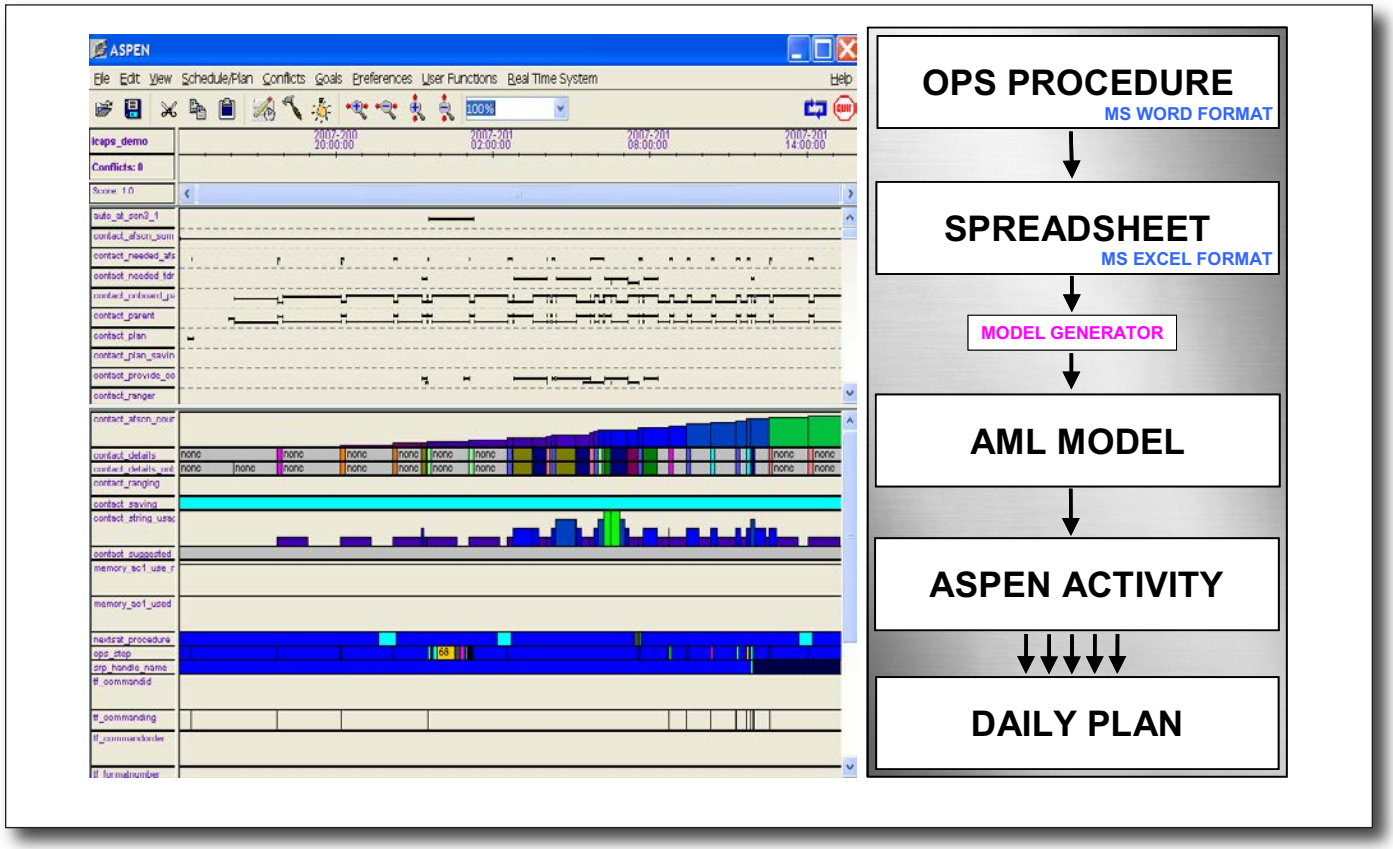

Figure 5. The ASPEN Planner-Scheduler Displays a Daily Plan.

we modeled the mission scenarios using the ASPEN (Rabideau et al. 1999) planning system. OE required evaluation of many alternatives, so ASPEN was modified to accommodate reasoning about schema-level uncertainty. Rehearsals for operations indicated that the SRP needed to be very responsive to changes in the procedures. To accommodate this, we implemented a system for reading the procedures and interpreting these into ASPEN models.

Technologies

The technologies we leveraged were (1) schema-level uncertainty reasoning, (2) procedure parsing for model generation, (3) procedural to declarative model translation, and (4) automated planning and scheduling. Schema-level uncertainty reasoning has at its core the basic assumption that certain variables are uncertain but not independent. Once any are known, then the others become known. This is important where a variable is uncertain for an action and many actions of the same schema (or type) exist in the plan. For example, the number of retries to purge the pump lines were unknown (but bounded), and each attempt required a subplan. Once we knew the correct number of attempts required for a purge, it would likely be the same for all subsequent purges. To accommodate changing scenario procedures, we ingested the procedures into a tabular format in temporal order, and used a simple natural language parser to read each step and derive the impact of that step on memory, power, and communications. We then produced an ASPEN model based on this analysis. That model was tested and further changed by hand, if necessary, to reflect the actual procedure. This resulted in a great savings in time used for modeling procedures.

Schema-Level Uncertainty Reasoning

To accommodate schema-level uncertainty reasoning in ASPEN, we modified the ASPEN Modeling Language to include a new reserved word — uncertain. Any parameter of any activity type that was unknown (but bounded) would be labeled using this reserved word, such as,

uncertain int retries = [1, 2]

or

uncertain string mode = (idle, transmitting, off).

Then, when an instance of ASPEN was started with uncertain variables, the cross-product of the instantiations of uncertain variables was used to produce unique instances of plans. Each of these instances is called an alternative. Note that this is the cross-product of the schema-level instantiations, not the actual activity-level instantiations. If we take our previous example, we would instantiate six alternatives:

retries = 1, mode = “idle”

retries = 1, mode = “transmitting”

retries = 1, mode = “off”

retries = 2, mode = “idle”

retries = 2, mode = “transmitting”

retries = 2, mode = “off”

Now, every activity in each alternative would have the same value, so it wouldn’t matter how many activities we had. This differs greatly from activity-level uncertainty. In this case, we would need to generate an alternative for each possible activity assignment. This means that we would have exponentially many alternatives with increasing activities. Since the uncertain parameters are those that we expect to learn (and to not vary), then we can expect that if a parameter has a value earlier in the day, it will have the same value later in the day.

Also, operations staff was loathe to trust a more analytical and compressed form of uncertainty reasoning. It was a very compelling case to see all possible executions, and when they needed to justify a certain resource allocation they found it simple and intuitive to use the set of alternatives.

To perform planning, we plan each alternative as if it was a separate schedule, and then perform a merge of the schedules, resulting in what operations people consider to be “odd” schedules, where we might ask for resource allocations that are impossible for a single spacecraft but still must be accommodated if we are to accommodate all possible alternatives. If we are not granted an allocation, we can go to each alternative and either try to replan it or simply carry it as a risk.

In practice, uncertain labels were used judiciously, not only to reduce the size of the set of problems to solve, but also to keep the solutions presented within the space of what humans could inspect and certify. The largest cross-product of schemas produced at most 32 separate plans.

Procedure Parsing for Model Generation

To accommodate late changes in procedures we implemented software that read procedures and produced ASPEN models. At first, this seemed like a daunting problem: we are in essence reading English text for content and producing a declarative activity/time line–based model of the procedure. One key observation we made is that the language of procedures is nearly as strict as a programming language, so we did not need to produce a parser capable of complete natural language processing; we just needed to accommodate stylistic differences between authors. Of course, some free-form text does appear in the procedures, and the model needed to be annotated such that the ASPEN model parser would complain in a meaningful way and the human modeler would address the text that was not understood.

This highly circumscribed form of natural language arose from the fact that these procedures were to interleave human actions on the ground and

machine actions in space. This is in stark contrast to other procedures (such as International Space Station [ISS] procedures) that might leave much to the interpretation of the reader and require training to be able to understand and perform, although currently efforts are under way to make ISS procedures more formally structured (Kortenkamp, Bonasso, and Schreckenghost 2008).

The procedures consisted of an introduction of human readable text, followed by a table of steps. They were authored using Microsoft Word. We found that most of the information needed to generate the procedure model was available in the table, so we would copy and paste the table into a Microsoft Excel document. Our parser was written in Visual Basic and embedded in the Microsoft Excel document.

Each step of the procedure had a number, the position or role responsible for carrying out the step, the action that was taking place, the response or verification to the action, and the expected duration. By parsing the action, we could determine whether the step included loops, if statements, or commands.

Loops in the procedures were accommodated using recursive decompositions. In ASPEN, it is often convenient to model activities and subactivities in trees known as hierarchical task networks. This representation is handy, but does not accommodate dynamic structures in the hierarchy. But it does allow for disjunctions, for example, an activity heater_parent can decompose into either a heater_child or a dummy activity. If we allow loops in the hierarchy, we can represent dynamic structures. The problem introduced by this is that the hierarchy appears to be infinitely deep. Therefore, we need to ensure that there are termination criteria; that is, at some point the loop breaks out to a subbranch that has no loops.

If statements were modeled using disjunctive decompositions. Both loops and ifs were candidates for uncertain variables.

The table also had commands that were to be sent to the spacecraft at execution time. Some of these commands were simple in that no further information was needed. In this case, the command activity was included as part of a decomposition. But, some of the commands required information to be input or computed. In this case, a human modeler needed to decide on the information source. To keep this from accidentally generating a working model, we would assign a known nonexistent variable the string value of the text describing the command argument. To ensure that command arguments and mnemonics were correct, we produced an ASPEN model from the command dictionary stored in a Microsoft SQL database. This was a piece of SQL code written by Boeing personnel. This included the legal bounds for each argument.

If any procedure had poorly formed commands, the ASPEN parser would catch them, and the procedure would be corrected. This was a relatively free value-added effect that resulted in the correction of many procedures.

Procedural to Declarative Model Translation

We have hinted at the necessity of converting procedural modeling information into declarative models. ASPEN is by design a declarative system. The procedures that were parsed were by nature procedural … meaning that each consists of a series of steps that include blocks, decision points (if statements), and loops. We not only needed to model the ground procedures but also had to model the on-board sequencer's (Timeliner) behavior. These were a collection of scripts that were executed for each command. The necessity of modeling the on-board execution comes from the requirement to model power use and to accommodate communication windows. These scripts were translated into ASPEN Modeling Language similarly to the previously mentioned procedures.

1. Each step in time is modeled as an individual activity, with the variables of interest being parameters in the activity.

2. Each series of steps was modeled as a single block, with a parameter network being constructed that represented the execution of the series of steps, as one might model a program execution in several rows in Excel.

3. Each if statement was modeled as a hierarchical activity with a disjunctive decomposition (a this-or-that decomposition). Constraints were imposed that forced the selection of the correct decomposition according the expression being evaluated by the if statement.

4. Each loop was modeled as a hierarchical activity with a disjunctive decomposition that included a recursion (that is, an activity lower in the instantiated hierarchy could be of the same type or schema as an activity higher in the hierarchy.) Note that termination criteria should default to terminating or the decomposition would expand endlessly on instantiation.

Ad Hoc Adjustments

Of course, the models needed to be maintained. As the mission progressed, unknown variables were adjusted to best estimate reality; for example, a hydrazine propellant transfer became more predictable after several were successfully demonstrated. Any model representing a procedure could need updating over time. There were cases in which the values simply changed; for example, the rate of a fuel transfer needed updating. However, there were also cases in which steps in the procedure needed to be removed or, more difficult from a planning perspective, added. To remove a step, the duration of the step could be set very low, or in the worst case, the procedure model could simply be regenerated. To

add steps, we almost always simply regenerated the model from the procedure using the translator.

Automated Planning

Underlying all was the ASPEN planner-scheduler. No adaptation-specific code was used for Orbital Express … all of the adaptation used the out-of-the-box capabilities of ASPEN. To plan any given day the following steps were followed:

1. Load all of the known opportunities (AFSCN visibilities, TDRSS visibilities, and procedures that have already been preplanned by hand or from previous runs of ASPEN).

2. Instantiate as many alternative plans as necessary to accommodate the uncertain parameters. In practice, long-term plans had several alternatives but short-term plans had only one.

3. Schedule all activities using an earliest-deadline-first ordering, also known as forward dispatch scheduling. This results in realigning activities to the known availabilities and initial expansion into the supporting subactivities.

4. Iteratively repair the plan to address any plan flaws (for example, unexpanded decompositions, open temporal associations between activities) and constraint violations (such as resource oversubscription, state violations, and timing violations).

Each of these steps was automatically performed by the ASPEN system under the direction of the SRP.

Long-Range Planning

The planning process for any given day began weeks in advance. A plan was built from knowledge of the existing contacts available and an ASPEN-generated and edited model of what the procedure was to do and how the contacts should lay out across time (figure 6)

The AFSCN contacts were reserved up to a limit and occasionally with elevated priorities specifically for the unmated scenarios. TDRSS support was originally also scheduled in the long-range planning time frame for all scenarios; however, cost constraints and changes to the plan in the short term dictated the need for a policy change.

It was determined more efficient to schedule TDRSS at the daily planning time, except in the case of unmated scenarios, where the timing and the more definite guarantee of contacts was crucial.

Although the essential replanning generally occurred at the daily planning time, variations on the long-range planning occurred from several factors. First, our launch delay created the need to replan all existing long-range plans to shift both AFSCN and TDRSS requests. Second, changes to models occurred often during the long-range process, due to many factors, including updated knowledge of timing, procedure step removals and additions, and general modifications to procedure step times or requirements. Third, occasionally, maintenance requirements or site operating hours were learned postdelivery of the long-range planning products and a replan was necessary. Finally, other factors that required replanning the long-range products were often late enough in the plan time line that a new "midrange" plan was created. This usually was done a few days outside of the daily planning. Figure 6 depicts the planning flow.

Daily Planning

In the morning of daily planning, the SRP would receive the list of contacts lost to other spacecraft and any suggested additions to replace these losses, and he or she would also receive the most up-to-date list of TDRSS availabilities. The contact losses would need to be evaluated against the procedure objectives of the day to determine whether they could still be met. The ASPEN model of the procedure could be adjusted as needed to reflect any operations updates, and the ASPEN activity could be moved around throughout the day to accommodate the contact requirements.

In the nominal case, the planning process would call for the use of the long-range plan and simply update necessary timing information to create the daily plan. However, daily planning was based on many variable factors culminating in a need for both simple updating of the plan and completely replanning the long-range plan: (1) The visibilities of contacts with the position of the spacecraft drifts slightly per day and must be updated in the short term to make most efficient use of the AFSCN communication times. Even one minute of contact coverage loss was, at times, considered valuable. (2) The daily deconfliction process can mean a loss of several contacts based on any number of reasons (site-specific issues, other satellite conflicts). Losses may require a shift of the procedure to perform the requested objectives. Also, losses are often accompanied by gains, and replanning can be based on such new additions to the plan. (3) Scoping of the day's long-range plan may change due to both anomalies and new direction from operations. Updating the existing plan at the daily planning time was often required for previously unknown amounts of needed coverage or for real-time failures of contacts pushing into the next day. (4) TDRSS support was originally requested in advance for all long-range planning, but as cost became an issue for unused contacts, the requests for TDRSS became part of the daily planning process. This was a major addition to the update of the long-range plan. (5) Dealing with the sometimes unpredictable conditions of space and limited mission time, a number of unforeseen events could cause the need to update the long-range plan.

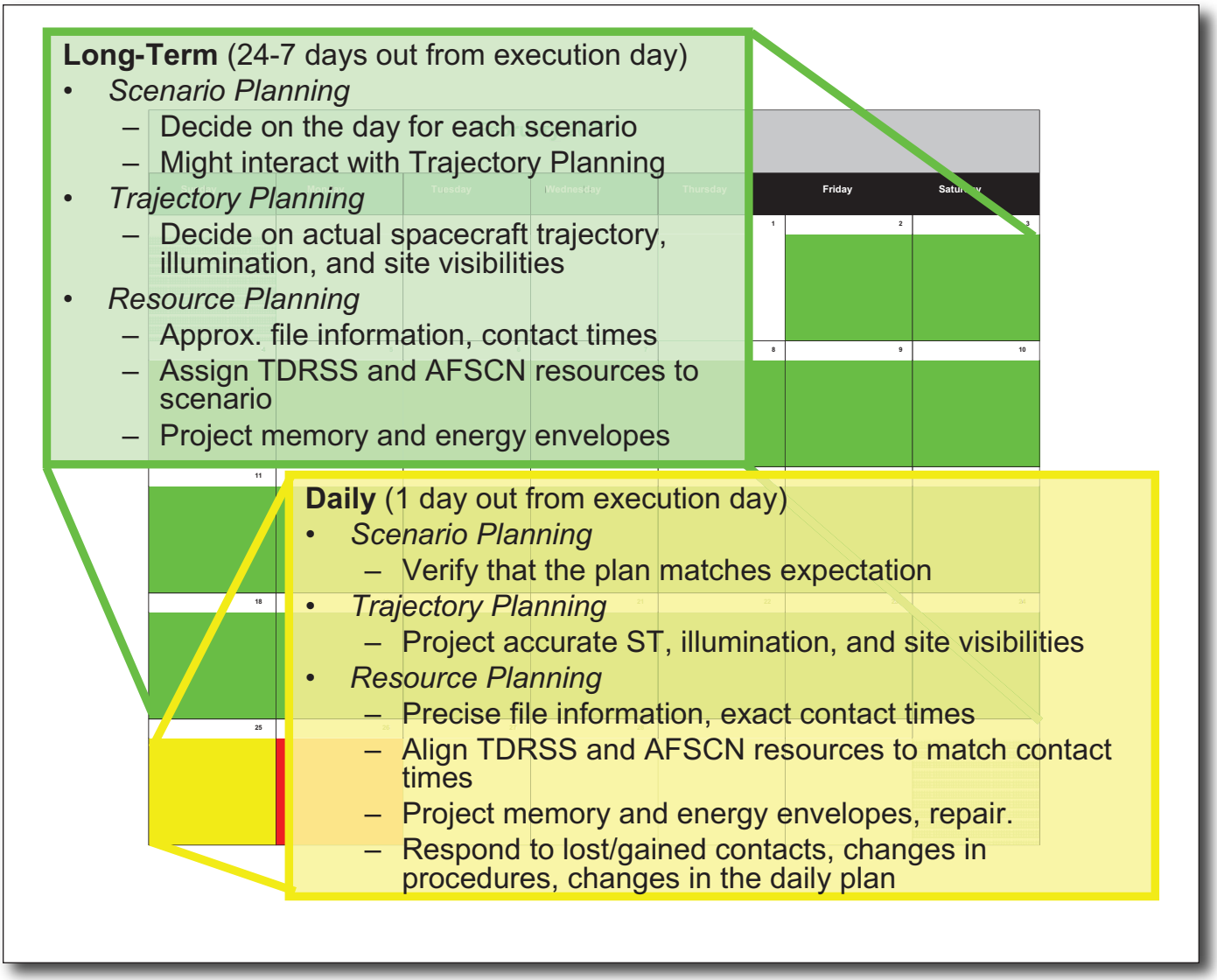

Figure 6. The Planning Flow.

Long Range and Daily Planning.

Impact

We were able to produce several alternatives for long-term planning so that enough communications resources were available at the time of execution. (For each alternative, we needed enough resources to handle all communications. Each alternative was based on differing execution paths.) We also were able to deliver operations plans daily, even in the face of changing procedures and changing resource availability. Together this contributed to the success of the mission.

The overall affect of using ASPEN has been approximated by the flight director as a 26 percent reduction in the execution time of the mission, a 50 percent reduction in the daily staff required to produce plans, and a 35 percent reduction in planning errors. Note that by planning error, we mean errors made by staff in terms of what plans should be executed that day, not errors in the plans themselves. This reduction was due to the high visibility of each plan and ready inspectability by expert staff. In fact, not a single misconfigured command was sent during operations.

All of these savings resulted in an estimated overall cost reduction of $10.4 million, mostly due to the reduction in execution time. Keeping in mind that the total cost of development and operations for the automated ground planning system was less than $1 million, this becomes a clear winner. Note that this does not include any economies of scale (which are normally associated with claims of automation reducing cost) as each major experiment was performed only once.

Lessons Learned

With a 100 percent mission criteria success rate, the Orbital Express project proved that spacecraft servicing is a reality for space operations. The goals for the JPL ASPEN team were to model the procedures and constraints of the mission and plan the long-term and daily operations. Using ASPEN and AML for Orbital Express modeling and planning, the planning team was able to represent mission constraints and procedures. The planning tool was flexible and adaptable to changing parameters.

In the long-term plan time frame, the plan for the execution day often changed or had several alternatives in one day (the nominal plan versus a backup plan). ASPEN's internal activity structure and repair algorithm allowed procedures to be shifted easily from one contact to another and to be deleted and replaced by new procedures without major reworking of the plan. Daily planning was adaptable to changes in which procedures and their associated ASPEN models were updated by operations the day of planning. The autogeneration of models allowed the planning team to share new procedure information and process it quickly for use on orbit. It also allowed the decision of using a procedure on a given day to be made at the very last minute without affecting the mission schedule. Models could be generated during the daily planning process and used the same day to plan that day's contacts.

Originally, NextSat contacts were not going to be scheduled using ASPEN; however, the simplicity of adding objectives to contacts with the planning tool, NextSat's low-maintenance strategy, and how easy it was for the SRPs to add the activities in ASPEN allowed SRPs to plan for multiple satellites and account for many real-world factors in planning operations.

The operational success of ASPEN's OE model can be largely attributed to the general benefits of automated planning and scheduling in which reusable activity models allow for faster human planning and decrease the need for redundant verification steps in the operations process; high levels of model parameter control allow quick adjustments to be made to both activities and the initial and/or ongoing state of the spacecraft and its domain; further, automated scheduling helps the plan operator or user view the "conflicts" that may or may not exist in a plan. The basic planning constructs of the ASPEN Modeling Language along with more complex capabilities introduced for OE (schema-level uncertainty and recursive decompositions) as well as the method in which the ASPEN core can invoke specialized functions for any existing model, particularly contributed to the success of this application deployment.

From an operational standpoint, long-term planning time could also have been reduced by requesting all visible contacts, instead of creating expected scenarios with their associated contacts; however, the activities of the scenarios would not have been validated to the extent they were long before execution.

Related Work

In June 1997, a docking of a Progress supply ship at the Mir space station was attempted but did not succeed. The Air Force Research Laboratory (AFRL) launched XSS-10 in 2003 and XSS-11 in 2005 with the objectives of advancing autonomous navigation and maneuvering technologies. Orbital Express was the first successful demonstrator of autonomous ORU (Orbital Replacement Unit) transfers in the world and of autonomous refueling in the United States. While several other missions over the past decade have approached the idea of autonomous satellite servicing with rendezvous and other robotic maneuvers, including NASA's Demonstration of Autonomous Rendezvous Technology (DART) satellite and Japan's National Space Development Agency (NASDA) Engineering Test Satellite 7, OE was the first successful demonstrator of autonomous rendezvous and docking (Dornheim 2006).

Planning operations for the Mars Exploration Rover (MER) mission is aided by the NASA Ames Research Center software tool Mixed Initiative Activity Plan Generator (MAPGEN) (AI-Chang et al. 2003), which is similar to ASPEN as an automated planner through the use of activities and temporal constraints. The nature of search for MAPGEN does not allow it to search the infeasible space for plan solutions; that is, when a constraint violation arises, the planner backtracks. ASPEN admits search in the infeasible space (in fact, threats and constraint violations are rolled up into a single generic entity called a conflict) allowing for faster response to off-nominal execution (Chien et al. 2000; Chien et al 2005). In fact, ASPEN has been demonstrated for in situ rover technology (Castano et al. 2007; Estlin et al. 1999; Gaines et al. 2008)

P. Maldague, A. Ko, D. Page, and T. Starbird (Maldague et al. 1997) have developed a schedule generation framework (APGEN) that automatically generates and validates plans used for commanding spacecraft but does not perform general planning and scheduling.

For an even more expansive survey of time line–based planning systems, see the paper by Chien et al. (2012). Note that much of the effort for Orbital Express was not in planning technology per se (we assumed it existed and worked), but in agile planning model generation.

Acknowledgements

Portions of this work were performed by the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration. Portions of this work were performed by the Jet Propulsion Laboratory,

California Institute of Technology, and were sponsored by the Defense Advanced Research Projects Agency. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not constitute or imply its endorsement by the United States government, or the Defense Advanced Research Projects Agency, or the Jet Propulsion Laboratory, California Institute of Technology.

References

AI-Chang, M.; Bresina, J.; Charest, L.; Jonsson, A.; Hsu, J.; Kanefsky, B.; Maldague, P.; Morris, P.; Rajan, K.; and Yglesias, J. 2003. MAPGEN: Mixed Initiative Planning and Scheduling for the Mars 03 Mer Mission. Paper presented at the 7th International Symposium on Artificial Intelligence, Robotics, and Automation in Space (i-SAIRAS-03), Nara, Japan, 19–23 May.

Castano, R.; Estlin, T.; Anderson, R.; Gaines, D.; Castano, A.; Bornstein, B.; Chouinard, C.; Judd, M. 2007. OASIS: Onboard Autonomous Science Investigation System for Opportunistic Rover Science. *Journal of Field Robotics* 24(5): 379–397.

Chien, S.; Johnston, M.; Frank, J.; Giuliano, M.; Kavelaars, A.; Lenzen, C.; Policella, N. 2012. A Generalized Timeline Representation, Services, and Interface for Automating Space Mission Operations. Paper presented at the 12th International Conference on Space Operations (SpaceOps 2012), Stockholm, Sweden, 11–15 June.

Chien, S.; Rabideau, G.; Knight, R.; Sherwood, R.; Engelhardt, B.; Mutz, D.; Estlin, T.; Smith, B.; Fisher, F.; Barrett, T.; Stebbins, G.; Tran, D. 2000. ASPEN – Automating Space Mission Operations Using Automated Planning and Scheduling. Paper presented at the 1st International Conference on Space Operations (SpaceOps 2000), Toulouse, France, 19-23 June.

Chien, S.; Sherwood, R.; Tran, D.; Cichy, B.; Rabideau, G.; Castano, R.; Davies, A.; Mandel, D.; Frye, S.; Trout, B.; Shulman, S.; Boyer, D. 2005. Using Autonomy Flight Software to Improve Science Return on Earth Observing One. *Journal of Aerospace Computing, Information, and Communication* 2(4): 196–216.

Dornheim, M. A. Orbital Express to Test Full Autonomy for On-Orbit Service, 2006. *Aviation Week and Space Technology* 164(23) (June 4): 46–50.

Estlin, T.; Mann, T.; Gray, A.; Rabideau, G.; Castano, R.; Chien, S.; and Mjolsness, E. 1999. An Integrated System for Multi-Rover Scientific Exploration. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*. Menlo Park, CA: AAAI Press.

Gaines, D. M.; Estlin, T.; and Chouinard, C. 2008. Spatial Coverage Planning and Optimization for Planetary Exploration. Paper presented at the 9th International Symposium on Artificial Intelligence, Robotics, and Automation in Space (i-SAIRAS 2008), Los Angeles, 26–29 February.

Kortenkamp, D.; Bonasso, R. P.; and Schreckenghost, D. 2008. A Procedure Representation Language for Human Spaceflight Operations. 2008. Paper presented at the 9th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS-08), Los Angeles, 26–29 February.

Maldague, P.; Ko, A.; Page, D.; and Starbird, T. 1997. APGEN: A Multi-Mission Semi-Automated Planning Tool. Paper presented at the 1st International Workshop on Planning and Scheduling for Space. Oxnard, CA, October.

Rabideau, R.; Knight, R.; Chien, S.; Fukunaga, A.; Govindjee, A. 1999. Iterative Repair Planning for Spacecraft Operations in the ASPEN System. Paper presented at the International Symposium on Artificial Intelligence, Robotics, and Automation in Space, Noordwijk, The Netherlands, 1–3 June.

Russell Knight is a member of the technical staff, Artificial Intelligence Group, Jet Propulsion Laboratory, California Institute of Technology. He is currently working on the automated scheduing and planning environment (ASPEN) planning system for mission operations, the Continuous Activity Scheduling, Planning, Execution, and Replanning (CASPER) real-time planning system, and the Mission Data System (MDS) (specifically the scheduler). Additionally, he supports research efforts in Antarctica. His research interests are in planning and scheduling, heuristic search, operations research, and space exploration automation.

Caroline Chouinard joined Red Canyon Software in 2011 during the prelaunch software verification stage of the Juno mission. She is currently working on the next-generation astronaut transport capsule known as MPCV (multipurpose crew vehicle). Chouinard came to Red Canyon Software from the Jet Propulsion Laboratory where she was most recently leading the Sequence Virtual Team for Cassini. She also held multiple roles on the Mars Exploration Rovers mission in software development and both strategic and tactical mission operations and was awarded for her work on the DARPA-funded technology demonstration Orbital Express.

Grailing Jones, Jr., is a senior technical staff member in the Planning and Execution Systems Group at the Jet Propulsion Laboratory. He has developed and operated mission planning and execution systems for a variety of NASA and DOD projects. He earned his B.S. in astronautics from the United States Air Force Academy, master's in aerospace engineering from the University of Colorado at Boulder, and master's in systems architecture and engineering from the University of Southern California.

Daniel Tran is a member of the technical staff in the Artificial Intelligence Group at the Jet Propulsion Laboratory, California Institute of Technology, where he is working on developing automated planning and scheduling systems for on-board spacecraft commanding. Tran attended the University of Washington and received a B.S. in computer engineering. He is currently the software lead for the Autonomous Sciencecraft Experiment, and he is cowinner of the 2005 NASA Software of the Year award.