ASPEN – Automated Planning and Scheduling for Space Mission Operations

S. Chien, G. Rabideau, R. Knight, R. Sherwood, B. Engelhardt, D. Mutz, T. Estlin, B. Smith, F. Fisher, T. Barrett, G. Stebbins, D. Tran

Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Drive, M/S 126-347, Pasadena, CA 91109-8099
email: {firstname.lastname}@jpl.nasa.gov
http://planning.jpl.nasa.gov

Abstract

This paper describes the ASPEN system for automation of planning and scheduling for space mission operations. ASPEN contains a number of innovations including: an expressive but easy to use modeling language, multiple search (inference) engines, iterative repair suited for mixed-initiative human in loop operations, real-time replanning and response (in the CASPER system), and plan optimization. ASPEN is being used for the Citizen Explorer (CX-1) (August 2000 launch) and the 2nd Antarctic Mapping Missions (AMM-2) (September 2000). ASPEN has also been used to automate ground communications stations – automating generation of tracking plans for the Deep Space Terminal (DS-T). ASPEN has been used to demonstrate automated command generation and onboard planning for rovers and is currently being evaluated for operational use for the Mars-01 Marie Curie rover mission. CASPER, the soft real-time versions of ASPEN, has been demonstrated with the Jet Propulsion Laboratory (JPL) Mission Data Systems (MDS) Control Architecture prototypes.[·]

Overview

Planning and scheduling spacecraft operations involves generating a sequence of low-level spacecraft commands from a set of high-level science and engineering goals. ASPEN (Automated Scheduling and Planning ENvironment) encodes spacecraft operability constraints, flight rules, spacecraft hardware models, science experiment goals, and operations procedures to allow for automated generation of low-level spacecraft sequences. By automating the command sequence generation process and by encapsulating the operations specific knowledge, ASPEN enables space missions to be controlled by a small operations team - thereby reducing costs.

ASPEN is an object-oriented system that provides a reusable set of software components that implement the elements commonly found in complex planning/scheduling systems. These include:

- An expressive constraint modeling language to allow the user to define naturally the application domain
- A constraint management system for representing and maintaining spacecraft operability and resource constraints, as well as activity requirements
- A set of search strategies for plan generation and repair to satisfy hard constraints
- A language for representing plan preferences and optimizing these preferences
- A soft, real-time replanning capability
- A temporal reasoning system for expressing and maintaining temporal constraints
- A graphical interface for visualizing plans/schedules (for use in mixed-initiative systems in which the problem solving process is interactive).

Automated planning and scheduling technology offers considerable promise in automating spacecraft operations. Planning and scheduling spacecraft operations involves generating a sequence of low-level spacecraft commands from a set of high-level science and engineering goals (see (Chien et al., 1998b) for an overview). In this paper, we discuss ASPEN and its use of an *iterative repair* algorithm for planning and scheduling as well as for replanning and rescheduling.

ASPEN is a reconfigurable planning and scheduling software framework (Fukunaga et al., 1997). Spacecraft knowledge is encoded in ASPEN under seven core model classes: activities, parameters,

[·] This work was performed by the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

parameter dependencies, temporal constraints, reservations, resources and state variables. An activity is an occurrence over a time interval that in some way affects the spacecraft. It can represent anything from a high-level goal or request to a low-level event or command. Activities are the central structures in ASPEN, and also the most complicated. Together, these constructs can be used to define spacecraft components, procedures, rules and constraints in order to allow manual or automatic generation of valid sequences of activities, also called *plans* or *schedules*.

Once the types of activities are defined, specific instances can be created from the types. Multiple activity instances created from the same type might have different parameter values, including the start time. Many camera imaging activities, for example, can be created from the same type but with different image targets and at different start times. The collection of activity instances is what defines the plan.

The job of a planner/scheduler, whether manual or automated, is to accept high-level goals and generate a set of low-level activities that satisfy the goals, do not violate any of the spacecraft flight rules or constraints, and optimize the quality of the plan. ASPEN provides a Graphical User Interface (GUI) for manual generation and/or manipulation of activity sequences. However, the automated planner/scheduler will be the focus of the remainder of this paper.

We have taken an early-commitment, local, heuristic, iterative search approach to planning, scheduling and optimization. This approach has many desirable properties for spacecraft operations planning.

1. Using an iterative algorithm allows automated planning to be utilized at any time and on any given initial plan. The initial plan may be as incomplete as a set of goals, or it may be a previously produced plan with only a few flaws. Repairing and optimizing an existing plan enables fast replanning when necessary from manual plan modifications or from unexpected differences detected during execution.
2. Heuristics allow the search to be pruned, ruling out less promising planning choices. In addition, heuristics may also suggest particular choices that may lead to a solution in less time, or to a higher quality solution.
3. A local algorithm does not incur the overhead of maintaining intermediate plans or past attempts. This allows the planner to quickly try many plan modifications for repairing the conflicts or improving the preferences. However, unlike systematic search algorithms, it cannot be guaranteed that our iterative algorithms will explore all possible combinations of plan modifications or that it will not retry unhelpful modifications. In our experience, these guarantees are not worth the required overhead.
4. By committing to values for parameters, such as activity start times and resource usages, the effects of a resource usage and the corresponding resource profiles can be efficiently computed. Least-commitment techniques retain plan flexibility, but can be computationally expensive for large applications. Further discussions on this topic can be found in (Chien et al., 1998b).

In the remainder of this paper we decribe the ASPEN system and its applications. First, we describe the ASPEN framework for modeling constraints, activities, state, and resources. Next, we describe the iterative repair search framework. We then describe the CASPER soft, real-time replanning capability. Next, we provide an overview of the ASPEN plan optimization capability. We then describe a number of ASPEN and CASPER applications to spacecraft commanding, mission design, autonomous rovers, and ground station automation. We then describe related work and conclusions.

Model Components and Constraints

Spacecraft models are developed in the ASPEN Modeling Language (AML) (Sherwood et al., 1998, Smith et al., 1998). These models are parsed into data structures that provide efficient reasoning capabilities for planning and scheduling. The seven basic ASPEN model components, which were listed above, are used to describe what the spacecraft can and cannot do during operations.

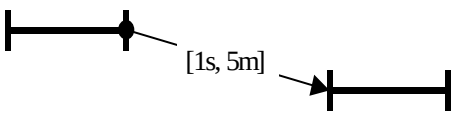

Figure 1: A temporal constraint with a required separation of at least 1 second and at most 5 minutes.

A *parameter* is simply a variable with a restricted domain. One parameter, for example, can be the range of integers between ten and twenty. Other parameter types include floating point numbers, booleans and strings. A *parameter dependency* is a functional relationship between two parameters.

An activity end time, for example, is a function (the sum) of the start time and the duration. A more complicated dependency might compute the duration of a spacecraft slew from the initial and final orientation.

In the model, relative ordering constraints can be specified for pairs of activities. A *temporal constraint* is a relationship between the start or end time of one activity with the start or end time of another activity (see Figure 1). One might specify, for example, that an instrument warming activity must end before the start of an activity that uses the instrument. Minimum and maximum separation distances can be specified in a temporal constraint. The warming activity for example, might be required to end at least one second but at most five minutes before using the instrument. Temporal constraints can be combined with conjunctive or disjunctive operators to form more complicated expressions.

A *resource* represents the profile of a physical resource or system variable over time (see Figure 2), as well as the upper and lower bounds of the profile. In ASPEN, a resource can either be depletable or non-depletable. A depletable resource is used by a reservation and remains used even after the end of the activity making the reservation. Examples of depletable resources on spacecraft include memory, fuel and energy. A non-depletable resource is used only for the duration of the activity making the reservation. Solar power is an example of a non-depletable resource. A resource can be assigned a capacity, restricting its value at any given time. A *state variable* represents the value of a discrete system variable over time. The set of possible states and the set of allowable transitions between states are both defined with the state variable. An example of a state variable is an instrument switch that may be either ON, WARMING, or OFF. This state variable may be restricted to transitions from OFF to WARMING but not directly to ON. *Reservations* are requirements of activities on resources or state variables. For example, an activity can have a reservation for ten watts of power. Some reservations are modeled as instantaneous effects (e.g., reservations that change the state on a state variable). The user can specify whether this effect occurs at the start or end of the activity.

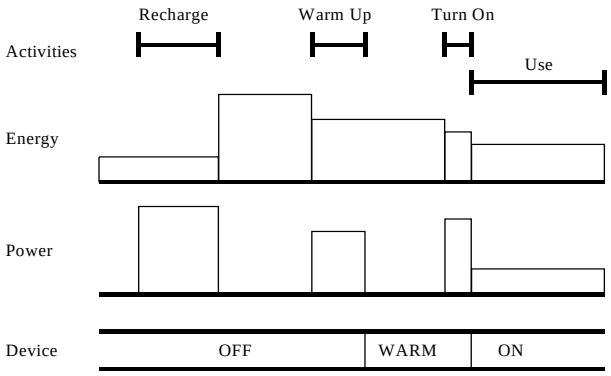

Figure 2: Timelines for activities, a depletable resource (energy), a non-depletable resource (power), and a state variable (device).

Activity hierarchies can be specified in the model using decompositions (see Figure 3). A decomposition is a set of sub-activities along with temporal constraints between them. In this way, one can define a high-level activity that decomposes into a set of lower-level activities that may be required to occur in some relative order. These activities in turn may have their own decompositions. In addition, an activity may have multiple decompositions to choose from. Thus, allowing an activity to be expanded in different ways.

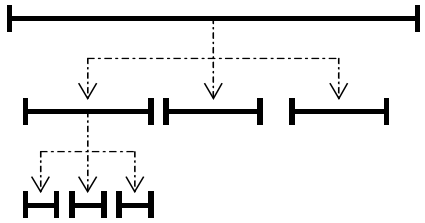

Figure 3: An activity hierarchy.

An *activity* has a set of parameters, parameter dependencies, temporal constraints, reservations and decompositions. All activities have at least three parameters: a start time, an end time, and duration. There is also at least one parameter dependency, relating these three parameters. In addition, all activities have at least one temporal constraint that prevents the activity from occurring outside of the planning horizon. Any additional components are optional.

Figure 4: The ASPEN graphical user interface (GUI). The pull-down menus and buttons give options for modifying the plan. Activities are shown as black horizontal bars in the middle section. The values of resources and state variables over time are shown as colored blocks in the bottom section.

Plan Conflicts and Repair

We define a conflict as a particular class of ways to violate a plan constraint (e.g., over-use of a resource or an illegal state transition). For each conflict type, there is a set of repair methods. The search space consists of all possible repair methods applied to all possible conflicts in all possible orders. We describe an efficient approach to searching this space. In ASPEN, the main algorithm for automated planning and scheduling is based on a technique called *iterative repair* [Zweben et al., 1994]. During iterative repair, the conflicts in the schedule are detected and addressed one at a time until no conflicts exist, or a user-defined time limit has been exceeded. A conflict is a violation of a parameter dependency, temporal or resource constraint. Conflicts can be repaired by means of several predefined methods. The repair methods are: moving an activity, adding a new instance of an activity, deleting an activity, detailing an activity, abstracting an activity, making a reservation of an activity, canceling a reservation, connecting a temporal constraint, disconnecting a constraint, and changing a parameter value. The repair algorithm first selects a conflict to repair then selects a repair method. The type of conflict being resolved determines which methods can repair the conflict. Depending on the selected method, the algorithm may need to make addition decisions. For example, when moving an activity, the algorithm must select a new start time for the activity.

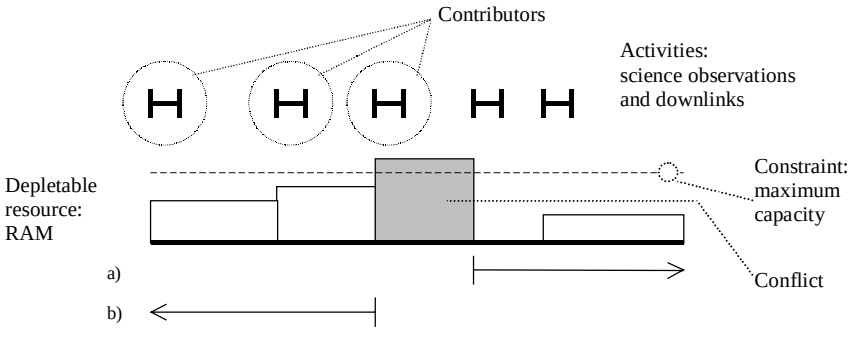

Figure 5: Repairing a depletable resource conflict. The arrows show time intervals that resolve the conflict by a) moving a positive contributor or b) adding a negative contributor.

Figure 5 shows an example situation for repair. On-board RAM is represented as a depletable resource. The shaded region shows a conflict where the RAM buffer has been oversubscribed. The

science activities using the resource prior to the conflict are considered contributors. Moving or deleting one of the contributors can repair the conflict. Another possibility would be to create a new downlink activity in order to replenish the resource and repair the conflict.

Soft Real-time Replanning

Traditionally, the majority of planning and scheduling research has focused on a batch formulation of the problem. In this approach, when addressing an ongoing planning problem, time is divided up into a number of planning horizons, each of which lasts for a significant period of time. When one nears the end of the current horizon, one projects what the state will be at the end of the execution of the current plan (see Figure 6). The planner is invoked with: a new set of goals for the new horizon, the expected initial state for the new horizon, and the planner generates a plan for the new horizon. As an example of this approach, the Remote Agent Experiment operated in this fashion (Jonsson et al. 2000).

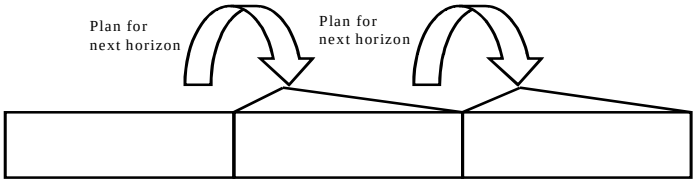

Figure 6: Traditional Batch "Plan then Execute" Cycle

This approach has a number of drawbacks. In this batch oriented mode, typically planning is considered an off-line process which requires considerable computational effort and there is a significant delay from the time the planner is invoked to the time that the planner produces a new plan.[1] If a negative event occurs (e.g., a plan failure), the response time until a new plan is generated may be significant. During this period the system being controlled must be operated appropriately without planner guidance. If a positive event occurs (e.g., a fortuitous opportunity, such as activities finishing early), again the response time may be significant. If the opportunity is short lived, the system must be able to take advantage of such opportunities without a new plan (because of the delay in generating a new plan). Finally, because the planning process may need to be initiated significantly before the end of the current planning horizon, it may be difficult to project what the state will be when the current plan execution is complete. If the projection is wrong the plan may have difficulty.

To achieve a higher level of responsiveness in a *dynamic planning* situation, we utilize a *continuous planning* approach and have implemented a system called CASPER (for Continuous Activity Scheduling Planning Execution and Replanning) (Chien et al., 2000). Rather than considering planning a batch process in which a planner is presented with goals and an initial state, the planner has a current goal set, a plan, a current state, and a model of the expected future state. At any time an incremental update to the goals, current state, or planning horizon (at much smaller time increments than batch planning)[2] may update the current state of the plan and thereby invoke the planner process. This update may be an unexpected event or simply time progressing forward. The planner is then responsible for maintaining a consistent, satisficing plan with the most current information. This current plan and projection is the planner's estimation as to what it expects to happen in the world if things go as expected. However, since things rarely go exactly as expected, the planner stands ready to continually modify the plan. From the point of view of the planner, in each cycle the following occurs:

- changes to the goals and the initial state first posted to the plan,
- effects of these changes are propagated through the current plan projections (including conflict identification)

[1] As a data point, the planner for the Remote Agent Experiment (RAX) flying on-board the New Millennium Deep Space One mission (Jonsson et al 2000) takes approximately 4 hours to produce a 3 day operations plan. RAX is running on a 25 MHz RAD 6000 flight processor and uses roughly 25% of the CPU processing power. While this is a significant improvement over waiting for ground intervention, making the planning process even more responsive (e.g., on a time scale of seconds or tens of seconds) to changes in the operations context, would increase the overall time for which the spacecraft has a consistent plan. As long as a consistent plan exists, the spacecraft can keep busy working on the requested goals and hence may be able to achieve more science goals.

[2] For the spacecraft control domain we are envisioning an update rate on the order of tens of seconds real time.

- plan repair algorithms[3] are invoked to remove conflicts and make the plan appropriate for the current state and goals.

This approach is shown in below in Figure 7. At each step, the plan is created by using incremental replanning from:

- the portion of the old plan for the current planning horizon;
- the change (Δ) in the goals relevant for the new planning horizon;
- the change (Δ) in the state; and
- the new (extended) planning horizon.

This incremental fast replanning approach as embodied in the CASPER system is being used in a range of applications as described later in this paper.

Plan Optimization

ASPEN also has facilities for representing and reasoning about plan quality (Rabideau et al. 2000). ASPEN adopts a local, early-commitment, iterative approach to optimization parallel to the iterative repair framework. During *iterative optimization*, low scoring preferences are detected and addressed individually until the maximum score is attained, or a user-defined time limit has been exceeded. A preference is a quality metric for a plan variable, and can be improved by making modifications to the plan similar to repair. For each preference, a domain-independent improvement expert automatically generates modifications that could potentially improve the preference score. For example, minimizing tardiness is a preference on the end time variables of activities and can be improved by moving activities to earlier times.

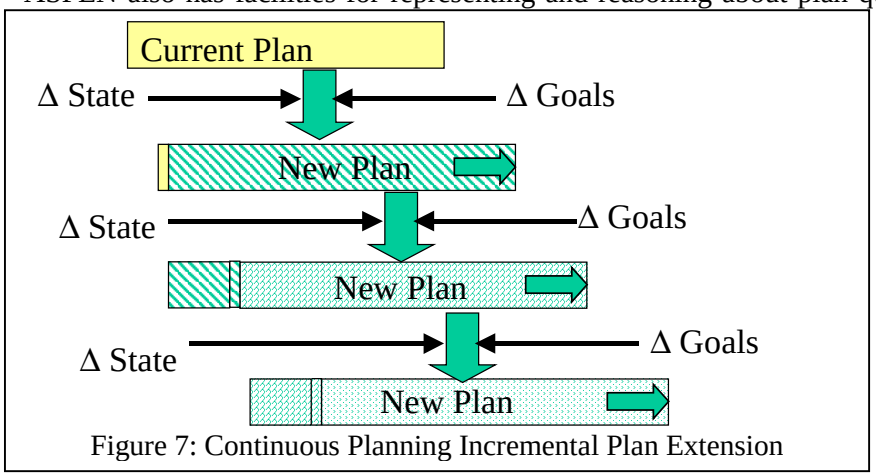

Figure 7: Continuous Planning Incremental Plan Extension

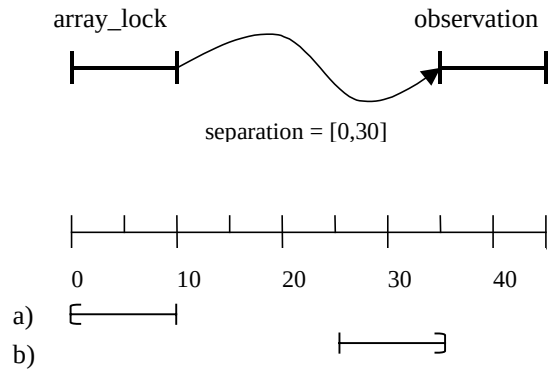

Figure 8: Start time intervals that improve a preference for centering the value of a separation variable by a) moving the `array_lock` or b) moving the `observation`.

In addition to establishing quality metrics, preferences can provide insight into how to improve plan quality. We define domain-independent *improvement experts* to aid in optimization. Improvement experts are based solely on the class of preference (and variable) for which it is constructed. An instance of an expert uses the preference specification to calculate plan modifications that will improve the score for the given preference and current plan. In other words, an expert is a link between changes in the plan and the change in quality.

For example, consider the plan fragment in Figure 8. The arrow represents a temporal constraint between the end of an array_lock activity and the start of an observation activity. The minimum, maximum and preferred separations are 0, 30, and 15 respectively. Locking the solar arrays reduces jitter caused by tracking the sun. However, with the arrays locked, available solar power drops as the spacecraft drifts. This is why the preferred value is 15, at the center of the upper and lower bounds.

[3] In this paper we do not focus on the state/resource representation or the repair methods, for details see (Rabideau et al. 1999).

Either activity can be moved to improve the preference. Alternatively, a different array_lock activity can be used for the observation.

Experts are local, however, and do not guarantee an increase in overall plan quality. Improvement experts provide a framework for optimization algorithms, defining the search space of possible improvements. We define a separate class of improvement expert for each class of preference. ASPEN currently supports five types of preferences. *Local activity variable preferences* represent preferences on parameters of an activity such as start time, end time, duration, relative time to other activities, etc. (e.g., minimize separation between science image activity and instrument calibration activity). *Activity/goal count preferences* are over the number of occurrences of certain types of activities (e.g., maximize science images, minimize maneuvers). *Resource/state variable preferences* specify desired values for resource levels and state variables (e.g., minimize power usage peak, maximize amount of energy in the battery, keep the buffer as empty as possible). *Resource/state change count preferences* specify a desire to maximize/minimize the number of times a state or resource changes (e.g., minimize the number of power switches for an instrument). A state duration preference implements a desire to maximize/minimize the amount of time that a given state is true (e.g., minimize instrument on time).

Recent ASPEN Applications

ASPEN and CASPER have been applied to a number of space related applications. These applications range from the core application of spacecraft command generation, to planning for rovers, to ground station automation and control of unmanned aerial vehicles. In the following section we describe how ASPEN has been applied in these domains and describe past, ongoing and future efforts.

ASPEN has been involved in a number of demonstrations on operations scenarios for spacecraft operations. Most notably, it has been demonstrated on early operations scenarios for the Earth Orbiting 1 (EO-1) mission (Sherwood et al. 1998) and on operations scenarios for the U.S. Naval Academy's UHF Follow On 1 (UFO-1) satellite. These early demonstrations provided valuable feedback on requirements for ASPEN's representational capabilities as well as planning and scheduling search algorithms.

More recently ASPEN has been used for the design of the Citizen Explorer (CX-1) mission and will be used for ground operations of the CX-1 spacecraft. CX is the first in a series of satellites being designed and built by students at The Colorado Space Grant Consortium (CSGC), University of Colorado at Boulder. Combined with ground measurements, CX-1 will enable significant studies of UV, aerosol, and ozone in the Earth's atmosphere. In the mission design phase, ASPEN enabled CX-1 designers to analyze the effects of various hardware configurations (most notably solar array sizing) and mission operations strategies (most notably downlink strategies) on science return and operations (Willis et al. 1999). After the CX-1 launch (scheduled in August 2000), ASPEN will be used in ground operations to provide manual and automated scheduling capabilities for satellite commanding over the course of mission operations (Wilklow et al. 2000). The CX-1 effort with CSGC represents a continuation of the highly successful DATA-CHASER collaboration (Chien et al. 1999). Continuing collaborations with CSGC are planned - CSGC has proposed three missions that baseline CASPER for flight (DISCO, DICE, and CANARE).

The ASPEN planning system is also being used to support mission design and operations for the 2nd Antarctic Mapping Mission (AMM-2) (Smith et al 2000). The core mission planning problem is to select a subset of the available swaths and data downlink opportunities that will cover the Antarctic within the 30 day mission horizon while satisfying operations constraints. The scientists select the swaths, and the automated ASPEN/AMM-2 planning system schedules the downlinks and identifies operations constraint violations. This system can process a 700 observation science cycle subplan in approximately 3 hours[4], which has enabled the AMM-2 operations team to rapidly plan the AMM-2 mission under short deadlines. The AMM-2 mission is scheduled for operations beginning in September 2000.

[4] The entire AMM-2 mission consists of 3 such subplans. As a point of comparison, the 1st Antarctic Mapping Mission, also consisting of 3 x 700 observation periods took months to plan manually.

ASPEN has also been used in a mission design framework. In this application, ASPEN enables mission engineers to quickly evaluate several designs. Engineers can evaluate several candidate designs against a given mission scenario by generating plans for each design and automatically evaluating them against objective criteria. Engineers can also use this system for "what-if" evaluations. They can see how a given design performs in the context of a mission scenario, and then modify the design or mission to improve performance. For example, a spacecraft may be limited to ten science images per orbit because of insufficient on-board data storage, even though there are opportunities for many more. The engineer increases the memory parameter and generates a new plan to see if the spacecraft can now take more science images.

ASPEN has been used to support an orbit trade study for the Space Interferometry Mission (SIM). The question was whether to use an inexpensive but highly constraining low-Earth orbit, or a more expensive but less constraining Earth-trailing orbit. We used the Aspen planner to generate a grid campaign for the Earth-trailing and low-Earth orbit cases, and for different exclusion angles. The objective was to determine whether Earth-trailing campaigns, which have fewer exclusion windows than Earth-orbit campaigns, were sufficiently faster to justify the more expensive orbit. The use of ASPEN enabled a more detailed projection of the exact impact of orbit on science return.

ASPEN was also used to support design analysis for the LightSAR mission (a mission to perform Synthetic Aperture Radar imagery of the Earth). The design questions are how the on-board storage constraints and downlink opportunities impact the science return. The storage capacity and downlink opportunities limit the number of swaths per orbit, and thus the total science return, but in a manner that is hard to predict. By generating plans for various storage capacities, available downlink stations, and goal distributions, mission designers can understand interactions that provide the best balance between science return and cost.

The MDS Project is an effort by the Jet Propulsion Laboratory to develop a common software flight and ground framework for mission operations for future spacecraft. Within the MDS Project, the Control Architecture team is developing a flight and ground software framework to allow automation of many mission operations functions (MDS 2000). This Control Architecture is designed to be compatible with automated planners to enable future missions to use such technologies (Knight et al. 2000). The CASPER system has been demonstrated on a number of MDS scenarios including a comet nucleus sample return landed operations scenario and science operations scenario for a virtual spacecraft. Additionally, a number of the components of ASPEN have been adopted in the design of the Mission Data Systems project. The timeline data structures and resource management interfaces have contributed significantly to the MDS architecture.

ASPEN has been applied to automated sequence generation for rovers and is also being use for onboard planning for rovers. Early efforts in this area involved operating the Rocky7 rover in the MarsYard testbed at JPL (Backes et al. 1999). This effort used the Web Interface for Telescience (WITS) to automatically command rover operations from high-level science goals entered through WITS. The resulting system enables faster generation of valid rover command sequences by a distributed planetary rover operations team. Users operated WITS to select science targets on a map of the terrain surrounding the rover. These science requests were transmitted to ASPEN which automatically reorders goals and adds new commands, generating an executable sequence that satisfies the goals while obeying rover flight rules and resource constraints. The current continuation of this effort involves building a model of the Marie-Curie rover (proposed for operation on Mars in -01 or -03) and evaluating the viability of automated command sequence generation in ground operations using ASPEN (Sherwood et al. 2000).

There is significant interest in using CASPER to provide onboard planning for single rover and multi-rover formations. In collaboration with the Long Range Science Rover effort, CASPER was integrated with onboard control software for the Rocky 7. CASPER generated validated rover-command sequences for Rocky 7 based on high level science and engineering activities. Once a plan has been generated it is continuously updated during plan execution to correlate with sensor and other feedback from the environment so that the planner may be responsive to unexpected changes.

CASPER has also been used in research demonstrations of autonomous spacecraft constellation (Barrett 1999, Barrett 2000) and rover swarms (Chien et al. 2000). In these efforts, CASPER is used in a distributed fashion to coordinate a team of rovers or spacecraft in achieving planetary science goals (for the remainder of this discussion we presume a distributed rover model but analogous efforts

are underway involving distributed spacecraft). For this application, a distributed version of CASPER was developed where it is assumed each rover has an onboard planner, which allows rovers to plan for themselves and/or for other rovers. Each onboard planner generates a rover command sequence for achieving science goals and can also perform execution monitoring and dynamic re-planning when necessary. This distributed planning environment is part of a multi-rover execution architecture being developed at JPL that integrates a number of systems including the ASPEN planning and scheduling system, a machine-learning data analysis system, Rocky 7 rover-control software, and a multi-rover simulation environment (Estlin et al. 1999).

The ASPEN planning and scheduling system was used to provide a key component of the DS-T (Deep Space Terminal) project (Fisher et al. 1999). The DS-T was a technology demonstration of autonomous control of a DSN (Deep Space Network) communications antenna. The ASPEN system was used to dynamically produce antenna control scripts from a set of high-level track goals. The DS-T concept was validated through a number of demonstrations beginning with partial tracks in April 1998, 1-day unattended operations in May, culminating in a 6-day autonomous "lights-out" demonstration in September 1998. Throughout these demonstrations ASPEN was used to automatically generate the necessary command sequences to fully automate a series of Mars Global Surveyor (MGS) downlink tracks using the equipment configuration at Deep Space Station (DSS) 26, a 34-meter JPL research antenna located in Goldstone, CA. These command sequences were produced and executed in a *fully autonomous* fashion with no human intervention.

After the completion of the DS-T task (during the lessons learned debriefing), the automated control script generation technology (made possible through the use of the ASPEN system) was identified as a key enabling component to the success of the DS-T task. Due to the success of the DS-T project, a more sophisticated autonomous DSS Controller utilizing CASPER for closed loop monitor, control, execution and recovery is being developed as a prototype (Fisher et al. 2000).

Related Work

There are a number of related planning and scheduling systems from the space community. The Remote Agent Experiment planner (RAX-PS) that flew onboard the Deep Space One Spacecraft in the Spring of 1999 (Jonsson et al. 2000) differs in that RAX-PS has focused on constraint-based, least commitment backward chaining search techniques. In contrast, although ASPEN does have a number of constraint propagation engines, ASPEN has focussed on search in committed representations (for reasons as to this approach see (Chien et al. 1998)). Another related system is the APGEN (Maldague et al. 1997) planning system, which is the mainstream planning system used by flight projects at JPL. APGEN has a scripting (e.g., procedural) language for automatically generating scripts but does not have a declarative, model-based planning and scheduling engine. Another planning and scheduling system that has been used at JPL is Plan-It2 (Eggemeyer et al. 1997). However, Plan-It2 also does not have a native generic model-based automated planning and scheduling capability.[5] SPIKE (developed for Hubble Space Telescope scheduling) automates elements of Hubble operations (Johnston & Miller 1994). SPIKE focuses on optimization of observation preferences where ASPEN focuses more on command generation. GPSS (Deale et al. 1994) uses iterative repair to support scheduling and rescheduling of Space Shuttle refurbishment activities. GPSS uses an iterative repair/optimization framework that can be viewed as the precursors for ASPEN's algorithms. OZONE (Smith et al. 1996) is another applications framework for scheduling systems. While ASPEN's application framework stance is derived from the OZONE system, OZONE (like RAX-PS) is more biased towards constraint-based scheduling approaches.

Conclusions

This paper has described the ASPEN system for automated planning and scheduling. ASPEN is a reconfigurable applications framework that includes: an expressive and easy to use modeling language, several search engines, soft real-time replanning, plan optimizations, and several constraint processing libraries. ASPEN has been used in a number of applications for spacecraft commanding,

[5] Although domain specific automated planning functions have been added in several demonstrations (Eggemeyer et al. 1997) and a generic iterative repair automated planning capability was also added to Plan-It2 for the DATA-CHASER deployment in DCAPS (Chien et al. 1999).

space mission design, autonomous rovers, ground station automation, and unmanned aerial vehicles. In particular, ASPEN is being used for the 2nd Antarctic Mapping Mission (AMM-2) and Citizen Explorer (CX-1) missions occurring in 2000. The multiple demonstrations combined with the two current deployments represent strong evidence to support the maturity of automated planning and scheduling for space applications and the utility of the ASPEN system.

References

P. Backes, G. Rabideau, K. Tso, S. Chien, "Automated Planning and Scheduling for Planetary Rover Distributed Operations," *Proc. of the IEEE Conference on Robotics and Automation*, Detroit, MI, May 1999.

S. Chien, N. Muscettola, K. Rajan, B. Smith, G. Rabideau, "Automated Planning and Scheduling for Goal-based Autonomous Spacecraft," *IEEE Intelligent Systems*, September/October 1998, pp. 50-55.

S. Chien, G. Rabideau, J. Willis, T. Mann, "Automating Planning and Scheduling of Shuttle Payload Operations," *Artificial Intelligence Journal* 114 (1999) 239-255.

S. Chien, A. Barrett, T. Estlin, and G. Rabideau, "A Comparison of Coordinated Planning Methods for Cooperating Rovers," *Fourth International Conference on Autonomous Agents*, Barcelona, Spain, June 2000.

S. Chien, R. Knight, A. Stechert, R. Sherwood, and G. Rabideau, "Using Iterative Repair to Improve Responsiveness of Planning and Scheduling," *Proceedings of the Fifth International Conference on Artificial Intelligence Planning and Scheduling*, Breckenridge, CO, April 2000.

C. Eggemeyer, S. Grenander, S. Peters, A. Amador, "Long Term Evolution of a Planning and Scheduling Capability for Real Planetary Applications, *Working Notes of the 1st International Workshop on Planning and Scheduling for Space*, Oxnard, CA 1997.

T. Estlin, T. Mann, A. Gray, G. Rabideau, R. Castano, S. Chien and E. Mjolsness, "An Integrated System for Multi-Rover Scientific Exploration," *16th Natl. Conf. Artificial Intelligence*, Orlando, FL, July 1999.

F. Fisher, T. Estlin, D. Mutz, S. Chien, "Using Artificial Intelligence Planning to Generate Antenna Tracking Plans", *Proc. 11th Conf. on Innovative Applications of Artificial Intelligence*, Orlando, FL, July 1999.

F. Fisher, R. Knight, B. Engelhardt, S. Chien, N. Alejandre, "A Planning Approach to Monitor and Control for Deep Space Communications", *Proc. IEEE Aerospace Conference*, Big Sky, MT, March 2000.

A. Fukunaga, G. Rabideau, S. Chien, D. Yan, "Toward an Application Framework for Automated Planning and Scheduling," *Proc. Intl. Symp. of Artificial Intelligence, Robotics & Automation for Space*, Tokyo, Japan, July 1997.

M. Johnston, G. Miller, "SPIKE: Intelligent Scheduling of Hubble Space Telescope Observations," in *Intelligent Scheduling*, Morgan Kaufman, 1994.

A. Jonsson, P. Morris, N. Muscettola, K. Rajan, and B. Smith, "Planning in Interplanetary Space: Theory and Practice," *Proceedings of the Fifth International Conference on Artificial Intelligence Planning Systems, Breckenridge*, CO, April 2000.

R. Knight, S. Chien, T. Starbird, K. Gostelow, R. Keller, W. Smith, et al., "Integrating Model-based Artificial Intelligence Planning with Procedural Elaboration for Onboard Spacecraft Autonomy," SpaceOps 2000, Toulouse, France

P. Maldague, A. Ko, D. Page, T. Starbird, "APGEN: A Multi-mission Semi Automated Planning Tool, *Working Notes of the 1st International Workshop on Planning and Scheduling for Space*, Oxnard, CA 1997.

G. Rabideau, R. Knight, S. Chien, A. Fukunaga, A. Govindjee, "Iterative Repair Planning for Spacecraft Operations in the ASPEN System," *International Symposium on Artificial Intelligence Robotics and Automation in Space*, Noordwijk, The Netherlands, June 1999.

G. Rabideau, B. Engelhardt, S. Chien, "Using Generic Preferences to Incrementally Improve Plan Quality," *Proceedings of the Fifth International Conference on Artificial Intelligence Planning and Scheduling*, Breckenridge, CO. April, 2000.

R. Sherwood, A. Govindjee, D. Yan, G. Rabideau, S. Chien, A. Fukunaga, "Using ASPEN to Automate EO-1 Activity Planning," *Proceedings of the 1998 IEEE Aerospace Conference*, Aspen, CO, March 1998.

R. Sherwood, A. Mishkin, T. Estlin, S. Chien, G. Rabideau, B. Engelhardt, B. Cooper , "An Automated Rover Command Generation Prototype for the Mars 2001 Marie Curie Rover," SpaceOps 2000, Toulouse, France, June 2000.

B. Smith, B. Engelhardt, R. Knight, D. Mutz, "Automated Planning for Spacecraft and Mission Design," *Third International Symposium on Intelligent Automation and Control*, Wailea, Hawaii, June 2000.

B. Smith, B. Engelhardt, D. Mutz, "Automated Planning for the Antarctic Mapping Mission," *Proc. 2nd International Workshop on Planning and Scheduling for Space*, San Francisco, CA, March 2000, pp. 184-186.

S. Smith, O. Lassila and M. Becker, "Configurable, Mixed-Initiative Systems for Planning and Scheduling", in *Advanced Planning Technology*, (ed.) A. Tate, AAAI Press, Menlo Park, CA, May, 1996

J. Willis, G. Rabideau, C. Wilklow, "The Citizen Explorer Scheduling System, "*Proceedings of the IEEE Aerospace Conference*, Aspen, CO, March 1999.

M. Zweben, B. Daun, E. Davis, and M. Deale, "Scheduling and Rescheduling with Iterative Repair," in *Intelligent Scheduling*, Morgan Kaufman, San Francisco, 1994.