Monitor and Control of Deep Space Communications through AI Planning.

Forest Fisher, Russell Knight, Barbara Engelhardt, Steve Chien, Niko Alejandre

Jet Propulsion Laboratory,
California Institute of Technology
4800 Oak Grove Drive, M/S 126-347
Pasadena, CA 91109-8099
Firstname.lastname@jpl.nasa.gov

ABSTRACT

In recent years with the large increase in the number of space missions at NASA, the demand for deep space communications services to command and collect data from these missions has become more difficult to manage. In an attempt to increase the efficiency of operating deep space communications antennas, we are developing a prototype system to perform monitoring, control, execution and recovery to automate the operations of the Deep Space Network (DSN) communication antenna stations. This paper describes the application of planning techniques for antenna track plan generation, monitoring, control, and execution for a NASA Deep Space Communications Station. The described system, CLEaR (Closed Loop Execution and Recovery), will enable an antenna communications station to automatically respond to a set of tracking goals by generating a plan and monitoring the plan during execution to correctly configure the appropriate hardware and software while adapting itself to its dynamic environment.

INTRODUCTION

The Deep Space Network (DSN) [5] was established in 1958 and since then it has evolved into the largest and most sensitive scientific telecommunications and radio navigation network in the world. The purpose of the DSN is to support unmanned interplanetary spacecraft missions and support radio and radar astronomy observations in the exploration of the solar system and the universe. The DSN currently consists of three deep-space communications facilities placed approximately 120 degrees apart around the world: at Goldstone, in California's Mojave Desert; near Madrid, Spain; and near Canberra, Australia (figure 1). This strategic placement permits constant observation of spacecraft as the Earth rotates, and helps to make the DSN the largest and most sensitive scientific telecommunications network in the world. Each DSN complex operates a collection of deep space stations consisting of 70-meter, 34-meter, 26-meter, and 11-meter antennae (figure 2). The functions of the DSN are to receive telemetry signals from spacecraft, transmit commands that control the spacecraft operating modes, generate the radio navigation data used to locate and guide the spacecraft to its destination, and acquire flight radio science, radio and radar astronomy, very long baseline interferometry, and geodynamics measurements.

From its inception, the DSN has been driven by the need to create increasingly more sensitive telecommunications devices and better techniques for navigation. The operation of the DSN communications complexes requires a high level of manual interaction with the devices in the communications link with the spacecraft. In more recent times NASA has added some new drivers to the development of the DSN: (1) reduce the cost of operating the DSN, (2) improve the operability, reliability, and maintainability of the DSN, and (3) prepare for a new era of space exploration with the New Millennium program: support small, intelligent spacecraft requiring very few mission operations personnel [11].

In order to address these new requirements for the DSN, we have worked on antenna station automation. In this paper we describe the Closed Loop Execution and Recovery (CLEaR) system being developed to address the problem of automated track plan generation (i.e. automatically determining the necessary actions to set up a communications link between a deep space antenna and a spacecraft), and monitor, control, execution and recovery for the DSN. In our approach we are utilizing artificial intelligence (AI) planning and scheduling techniques to generate the track plans, and we are utilizing a continuous planning approach to provide monitor, control, execution and recovery. Similar to many planning problems, track plan generation involves elements such as subgoaling to achieve preconditions and decomposing high-level (abstract) actions into more detailed sub-actions. However, unlike most classical planning problems, the problem of track generation is complicated by the need to reason about issues such as metric time, DSN resources and equipment states. To address this problem, we have applied the Continuous Activity Scheduling Planning Execution and Replanning (CASPER) engine, a generic framework for automated planning, scheduling, execution and replanning, to generate antenna track plans on demand [3,4]. CASPER has been adapted to input antenna-tracking goals and automatically produce the required command sequence to set up and perform the requested communications link.

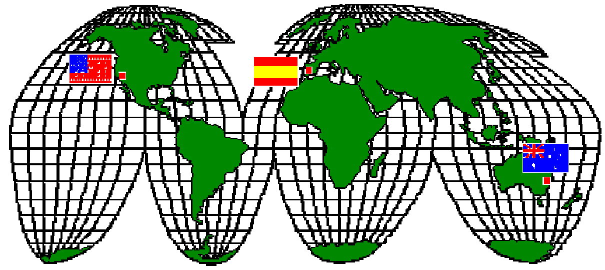

Figure 1 World Map of Deep Space Network

This work is one element of a far-reaching effort to upgrade and automate DSN operations building on previous work. The ASPEN Track Plan Generator, which was demonstrated in support of the Deep Space Terminal (DS-T), an autonomous prototype 34-meter deep space communications station [6,7,8,9], produced batch plans with limited conditionals for error recovery. CLEaR is the continuation of the automation concepts introduced during DS-T but is intended to demonstrate a greater level of automation and robustness while providing a larger class of communication services.

The rest of this paper introduces the reader to the deep space communications domain, describes the continuous planning techniques of the CASPER system and how we use CASPER to automate communication antenna stations operations.

HOW THE DSN OPERATES

The DSN track process occurs daily for dozens of different NASA spacecraft and projects, which use the DSN to command spacecraft, as well as capture spacecraft and science data. There are many earthside challenges that must be addressed before a spacecraft's signal is acquired and successfully transformed into useful information.

The first step in performing a DSN track is called network preparation. Here, a project sends a request for the DSN to track a spacecraft involving specific tracking services (e.g. downlink, uplink). The DSN responds to the request by attempting to schedule the necessary resources (i.e. an antenna and other shared

Figure 2 70-Meter Deep Space Communication Antenna

equipment) needed for the track. Once an equipment schedule and other necessary information has been determined, the next step is the data capture process, which is performed by operations personnel at the deep space station. During this process, operators determine the correct steps to perform the following tasks: configure the equipment for the track, establish the communications link, and perform the actual track by issuing control commands to the various subsystems.

Throughout the track the operators continually monitor the status of the communications link and handle exceptions (e.g. the receiver looses signal lock with the spacecraft) as they occur. To perform all of these actions, human operators manually issue tens to hundreds of command directives via a computer terminal. Instead, the CLEaR system can be used as a prototype monitor, control, and execution system for DSN communication antenna automation. The CLEaR system requires a flexible and robust planning system to perform these operations, such as CASPER.

CASPER: INTEGRATED PLANNING AND EXECUTION

Motivation

Traditionally, much of planning and scheduling research has focused on a batch formulation of the problem. In this approach (see Figure 3), time is divided up into a number of planning horizons. When the end of the current horizon is near, the planner can project what the state will be at the end of the execution of the current plan. The planner is invoked with a new set of goals and this state as the initial state. The Deep Space One Remote Agent Experiment operated in this fashion [14].

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, hence there is a significant delay from the time the planner is invoked to the time that the planner produces a new plan. If a plan failure occurs, the response time until a new plan is generated may be significant. During this period the system being controlled might not be operated appropriately.

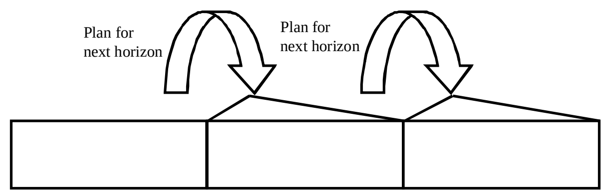

Figure 3 Traditional Batch "Plan then Execute" Cycle

If a positive event occurs (e.g., a fortuitous opportunity), the current plan may not be able to take advantage of the opportunity because the opportunity may be a small window, and the response time is too slow. Because the planning process is initiated long 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 current plan may be infeasible given the new state.

Consider the operations of a spacecraft. In a traditional plan-sense-act cycle, planning occurs on a relatively long-term planning horizon. In this approach, operations for a spacecraft would be planned on the ground on a weekly or daily basis. The science and engineering operations goals would be considered, and a plan for achieving the goals would be generated. This plan would then be uplinked to the spacecraft for execution. The plan would then be executed onboard the spacecraft with little or no flexibility. If an unexpected event occurred due to environmental uncertainty or an unforeseen failure, the spacecraft would be taken into a safe state by fault protection software. The spacecraft would wait in this state until the ground operations team could respond and determine a new plan.

The CASPER System

To achieve a higher level of responsiveness in a dynamic planning situation, we utilize a continuous planning approach with the CASPER system [3,4]. 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 update to the goals or system state may effect the feasibility of the plan. This update may be an unexpected event or simply time progressing forward. The planner is responsible for maintaining a consistent, feasible plan with the most current information. This current plan is feasible if the state projection is accurate, and the plan is quickly modified if the state projection is not accurate. In each planning cycle the following occurs:

- changes to the goals and the initial state are first posted to the plan,
- effects of these changes are propagated through the current plan projections
- plan repair algorithms remove conflicts and make the plan appropriate for the current state and goals.

This approach is shown in Figure 4. At each step, the plan is updated and elaborated using iterative repair with the portion of the old plan within the current planning horizon, the updated goals and state; and the extended planning horizon.

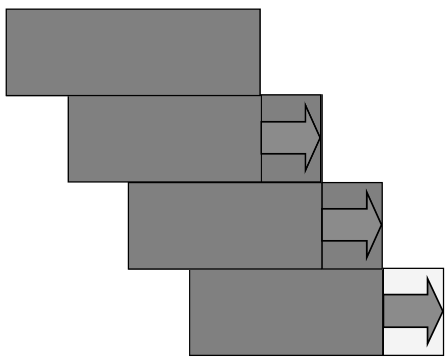

Figure 4 Continuous Planning Incremental Extension

The synchronization between planning and execution is handled by an activity commitment process. During execution of the current plans, there is an activity commitment window that represents the near future. When an activity overlaps with this window it is committed, and the planner is forbidden from altering any aspect of this activity. Thus far we have focused on time-based commitment strategies, but our architecture supports more complex commitment strategies such as parameter-based commitment strategies.

In addition to increasing the responsiveness of planning, the continuous planning approach has additional benefits:

- The planner can reduce reliance on predictive models, since it will be updating its plans continually.
- Fault protection and execution layers need only to be concerned with the immediate horizon since the planner will replan within a short time span.
- Because of the hierarchical reasoning taking place in the architecture there is no hard distinction between planning and execution.

In conjunction with this incremental, continuous planning approach, CASPER supports a hierarchical approach to planning. In this approach, the long-term planning is performed at a very abstract level. Shorter planning horizons are planned in greater detail, until finally at the most specific level the planner plans only a short time in advance (just in time planning). This paradigm is illustrated in Figure 5. Within each of these layers, the planner operates continuously in the mode described above. However, the length of the planning horizon and the frequency with which the plan is updated varies. In the longer-term, the planning horizon is longer and the abstract plan is updated less frequently. In the short-term level, the plans are updated rapidly.

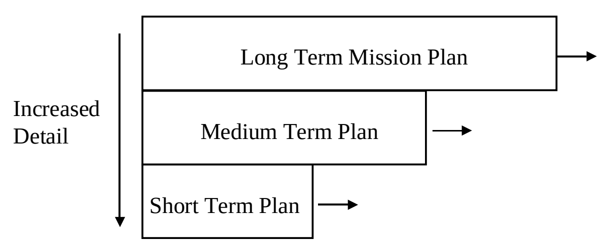

Figure 5 Hierarchical Planning Horizons

The idea behind this hierarchical approach is that only very abstract projections can be made over the long-term because prediction is difficult due to limited computational resources and timely response requirements. Hence there is little utility in constructing a detailed plan far into the future, since it might be infeasible.

An Architecture for Integrated Planning and Execution

Our approach to integration of planning and execution relies on four separate classes of processes.

- The Planner Process - this process represents the planner, and is invoked to update the model of the plan execution, to refine the plan, or when new goals are requested.
- The Execution Process - this process is responsible for committing activities and issuing actual commands corresponding to planned activities.
- The State Determination Process - this process is responsible for monitoring and estimating states and resource values and providing accurate and timely state information updates.
- The Synchronization Process - this process enforces synchronization between the execution, planner, and state determination processes.

The overall architecture for the continuous planning approach is shown in Figure 6. The planner process maintains a current plan that is used for planning. It responds to requests to replan initiated by the execution processes, activity commitments from the execution module, state and resource updates from state estimation, and new goals. All of these requests are moderated by the synchronization process that queues the requests and ensures that one request is complete before another is initiated. The planner's copy of the current plan is also where state projection takes place.

The execution process is only concerned with the current state of the system and executing activities. The execution module maintains a copy of the plan that is incrementally updated whenever the planner changes a goal, a state, or an activity. The execution module has three general responsibilities:

1. to commit activities in accordance with the policy as they approach their execution time;
2. to actually initiate the execution of commands at the associated activity start times,
3. to request re-planning when conflicts exist in the current plan.

The execution module performs 1 & 2 by tracking the current time and indexing into relevant activities to commit and execute them. The execution module also tracks conflict information as computed by state projections and submits a replanning request to the synchronization module when a conflict exists.

The state estimation module is responsible for tracking sensor data and summarizing that information into state and resource updates. These updates are made to the synchronization module that passes them on to the planners plan database when coordination constraints allow.

The synchronization module ensures that the planner modules are correctly locked while processing. At any time the planner can only be performing one action: (re)planning, updating its goals or current system states, revising the execution

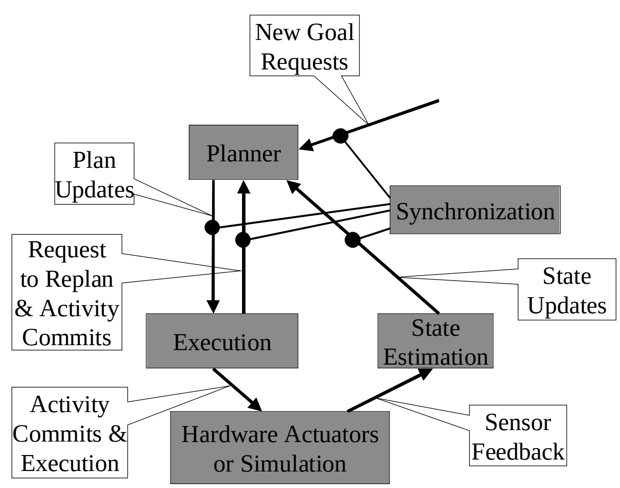

Figure 6 CASPER Architecture

module's plan for execution, or updating commitment status. The synchronization module serializes these requests by maintaining a FIFO task queue for the planner and forwarding the next task only when the previous task has finished.

The execution module also has a potential synchronization issue. The planner must not be allowed to modify activities if those activities might already have been passed on to execution. We enforce this non-interference by committing all activities overlapping a temporal window extending to a few seconds from the current time. CASPER ensures that each replan request always returns within this time bound. Additionally, we use the synchronization process to ensure that the execution module does not commit activities while the planner is replanning.

MONITOR AND CONTROL THROUGH CONTINUOUS PLANNING

In this section we explain how to automate monitor and control of communication services using CLEaR. The CLEaR system views the antenna station as an autonomous unit within the DSN. The CLEaR automation engine is intended to be deployed at this deep space station. The station has a controller that is responsible for determining and monitoring the antennae behavior. The CLEaR system provides this functionality as the primary control module of the station controller's automation software.

Given a set of inputs: a station schedule, service request, spacecraft sequence of events (SOE), equipment configuration, an antenna operations knowledge base (KB), a track plan if one exist, and station state information, the automation system produces a track plan (or control script) (figure 7). These control scripts are referred to as Temporal Dependency Networks (TDNs). The TDN scripts are made up of smaller components, called ALMO blocks which are executable scripts implemented in the Automation Language for Managing Operations (ALMO) scripting language [15]. The ALMO blocks are represented in the knowledge base as planning activities. The knowledge base expresses the behavior of the blocks (pre- and post-conditions) as well as temporal relation, temporal estimates on execution times, resource usage, and domain knowledge which are all used to determine the necessary steps involved in providing the high level service request.

The service request represents the high-level communication services that must be performed, such as downlinking data at a given frequency and bit-rate or uplinking a spacecraft command sequence. The service request is used in conjunction with the spacecraft SOE to create the planning goals inputted into the planning engine. It is necessary to use the spacecraft SOE because, in order to maintain the communication link with the spacecraft, the ground system must be aware of (and synchronized with) the communications activities of the spacecraft. The types of information that are expressed in the SOE are the current modes of the spacecraft and the times which those modes change.

From the set of inputs mentioned above, CLEaR considers the goals, which are extracted from the service request and SOE, within the context of the current station configuration and then produces an initial track plan based on the available operations defined in the antenna operations knowledge base (KB). CLEaR utilizes the CASPER system to perform the planning and scheduling.

Once the TDNs are produced, the executive component of the CLEaR system begins stepping

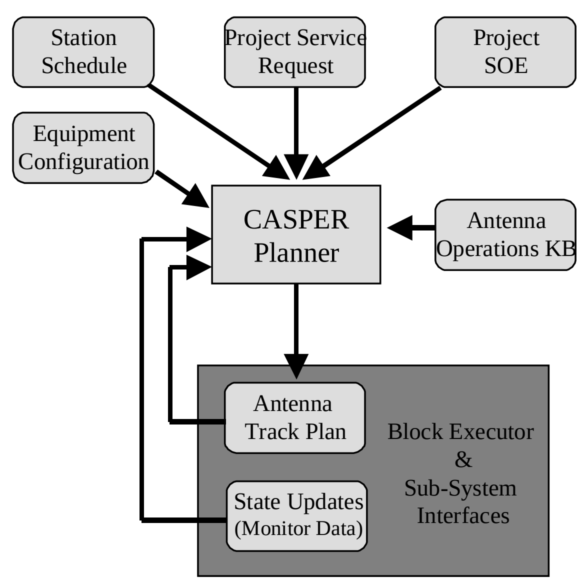

Figure 7 Inputs and Outputs to CLEaR

through the plan. As time progresses through the plan and the start time of a block (activity) arrives the block is sent to the ALMO script interpreter and the block is executed, which results in command directives being sent to the appropriate subsystems. Each of these subsystems in turn produces monitor data which is fed back into CASPER so it can update the state representation and revise the plan if necessary. It is important that the plan converges quickly because the spacecraft will continue with its given communication sequence regardless of the state of the ground systems. As this might imply, the ground station is not responsible for determining what communication should take place, but how to control the ground station equipment to provide the requested communication service.

A communication track or pass is broken up into three portions: pre-track, in-track and post-track. Pre-track consists of configuring the station's equipment to perform the requested communications services. During the configuration phase, equipment is powered on, warmed up, configuration files loaded, etc. Once the appropriate configuration has been performed the station is ready to perform the desired communication services. The next portion of the track is the in-track phase. It is here where the actually communication service is performed. At a high level, this consists of transmitting and receiving of data. During this phase the station must be commanded to maintain the antenna pointing at the spacecraft, to acquire, maintain, and transmit signals. At the conclusion of the track, the post-track phase returns the station to a standby state to wait for the next communication pass. This phase includes archiving data, generating reports, data deliver and commanding the sub-systems into a standby state.

The steps in each of these three phases vary depending on the types of service requested. The services can all be categorized into one of four basic service types (within which there are a large number of possible variations): Doppler, Telemetry, Commanding, and Ranging.

Doppler service refers to tracking the spacecraft as it moves across the sky and adjusting the receiver's frequency to adjust for the Doppler shift. The receiver is used to confirm that the spacecraft is being "tracked" by the ground station. Telemetry service refers to the collection (downlink) of spacecraft health data (engineering telemetry) and science data (science telemetry). Commanding service refers to the transmitting (uplink) of command sequence to the spacecraft. Ranging service refers to the process of confirming the position of the spacecraft and is used to confirm the spacecraft's trajectories. Each of these four services includes the previous service.

Many of the services, and the vast number of possible equipment configurations, result in complex interactions between command directives.

STATUS

While the CLEaR task is ongoing with considerable implementation still to be done, successful preliminary work has been done.

The current status of CLEaR is an initial knowledge base model has been built to support the current static script TDNs used to configure the station in the pre-track and post-track phases in order to support the four basic classes of service.

Currently the CLEaR knowledge base is being extended to support recovery for each of these four station configuration scenarios of pre-track and post-track. The next step, which we are addressing in parallel, is to produce control scripts for the in-track phases of the communication service as well as the knowledge to perform recovery for events during execution.

In order to validate our approach, we are integrating CLEaR into a prototype Deep Space Station Controller (DSSC). The DSSC system will consist of the CLEaR system to perform monitor, control, execution and recovery through the use of AI planning and scheduling techniques in order to perform the decision making process, and a FDI component mentioned in the next section. These technologies are being integrated with the existing automation control software in order to enhance the capabilities of the automation infrastructure that has already been developed.

FUTURE WORK

This CLEaR effort is also being integrated with a Fault Detection, Isolation and Recovery (FDIR) system [12]. FDIR is an expert system providing monitor data analysis. As is often the case with large complex systems, monitor (sensor) data is often related in unintuitive ways that are difficult for humans to detect. The advantage of combining these two systems is that FDIR can first interpret the vast amount of data and summarize it into a set of meaningful values for a planning system to react to. We think of this union as intelligent analysis and intelligent response, much like a careful design and implementation; one without the other is of little use.

Another area of future work is in the area of mixed-initiative control. This deals with how a system capable of autonomous operations interacts with an operator such that neither interferes with the other, and once control is returned to the autonomous system the system must understand both the state of the world and the changes that the user has made.

RELATED WORK

While the automation techniques utilized in the development of CLEaR rely heavily on AI planning and scheduling, we ask the reader to look to our papers on ASPEN [10] and CASPER [3,4] for work related to our planning and scheduling techniques.

There are a number of existing systems built to solve real-world planning or scheduling problems [16,17,18]. The problem of track plan generation combines elements from both these areas and thus traditional planners and schedulers cannot be directly applied. Another approach to DSN antenna automation was taken by the Network Monitor and Control (NMC) task. NMC approach uses canned control scripts to automate antenna operations, compared to the CLEaR approach of dynamically constructing the control script out of smaller static scripts. It is because of this infeasible approach that we are integrating CLEaR into the larger NMC infrastructure.

Two other systems were previously designed to generate antenna track plans, the Deep Space Network Antenna Operations Planner (DPLAN) [2]. DPLAN utilizes a combination of hierarchical-task network (HTN) and operator-based planning techniques. Unlike DPLAN, CASPER has a temporal reasoning system for expressing and maintaining temporal constraints and also has the capability for representing and reasoning about different types of resources and states. CASPER can also utilize different search algorithms such as constructive and repair-based algorithms, while DPLAN uses a standard best-first based search. CASPER is currently being extended to perform dynamic planning for closed-loop error recovery, while DPLAN has only limited replanning capabilities.

For the reasons stated above the DS-T automation controller was developed using ASPEN [9]. This greatly improved the capabilities for generating track plans over the DPLAN system. DS-T utilized a classical approach of batch planning to produce the control script and then handed the script over to an execution environment. Unlike the DS-T approach, CLEaR utilizes the CASPER planning and scheduling engine which enables CLEaR to perform dynamic replanning in response to changes detected during the execution of the control scripts.

CONCLUSION

This paper has described the Closed Loop Execution and Recovery (CLEaR) system and the manner in which it performs the monitor and control functionality for track automation of DSN communication antennas. Through the use of CASPER, CLEaR utilizes a knowledge base of information on tracking activity requirements and a combination of planning and scheduling techniques to generate antenna track plans that will correctly setup and perform a communications link with spacecraft. The monitor and control capabilities are further enhanced by dynamically feeding monitor data back into the planning system as state updates, which enables the planning system to validate and repair the current plan. Through this continual planning approach, CLEaR makes a DSN antennae station to function autonomously making it much more responsive and reactive to changes in the dynamic world.

ACKNOWLEDGEMENTS

The work described was funded by the TMOD Technology Program and performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract of the National Aeronautics and Space Administration. We thank members of the CASPER & ASPEN planning scheduling development team, Telecommunications and Missions Operations Directorate (TMOD), our sponsors in the Telecommunications and Missions Operations Technology (TMOT) office, members of the FDIR development team, and members of the Network Monitor and Control (NMC) automation team for contributing and supporting this work.

REFERENCES

[1] S. Chien, D. Decoste, R. Doyle, and P. Stolorz, “Making an Impact: Artificial Intelligence at the Jet Propulsion Laboratory,” AI Magazine, 18(1), 103-122, 1997.

[2] S. Chien, R. Hill Jr., A. Govindjee, X. Wang, T. Estlin, A. Griesel, R. Lam and K. Fayyad, “A Hierarchical Architecture for Resource Allocation, Plan Execution, and Revision for Operation of a Network of Communication Antennas,” Proceedings of the 1997 IEEE Conference on Robotics and Automation, Albuquerque, NM, April 1997.

[3] S. Chien, R. Knight, A. Stechert, R. Sherwood, and G. Rabideau, “Integrated Planning and Execution for Autonomous Spacecraft,” To appear in the Proceedings of the 1999 IEEE Aerospace Conference, Aspen, CO, March, 1999.

[4] S. Chien, R. Knight, A. Stechert, R. Sherwood, and G. Rabideau, "Using Iterative Repair to Increase the Responsiveness of Planning and Scheduling," Artificial Intelligence Planning Systems, Bolder, CO, April 2000 (to appear).

[5] Deep Space Network, Jet Propulsion Laboratory Publication 400-517, April 1994.

[6] F. Fisher, R. Knight, B. Engelhardt, S. Chien, and N. Alejandre, “A Planning Approach to Monitor and Control for Deep Space Communications,” Proceedings of the 2000 IEEE Aerospace Conference (to appear).

[6] F. Fisher, S. Chien, L. Paal, E. Law, N. Golshan, and M. Stockett, “An Automated Deep Space Communications Station,” Proceedings of the 1998 IEEE Aerospace Conference, Aspen, CO, March 1998.

[7] F. Fisher, D. Mutz, T. Estlin, L. Paal, and S. Chien, “The Past, Present and Future of Ground Station Automation within the DSN,” To appear in the Proceedings of the 1999 IEEE Aerospace Conference, Aspen, CO, March 1999.

[8] F. Fisher, T. Estlin, D. Mutz, S. Chien, "Using Artificial Intelligence Planning to Generate Antenna Tracking Plans", Eleventh Annual Conference on Innovative Applications of Artificial Intelligence, Orlando, Florida, July 1999.

[9] A. Fukanaga, G. Rabideau, S. Chien, and D. Yan, "Toward an Application Framework for Automated Planning and Scheduling," Proceedings of the 1997 International Symposium of Artificial Intelligence, Robotics and Automation for Space, Tokyo, Japan, July 1997.

[10] R. W. Hill, Jr., S. A. Chien, K. V. Fayyad, C. Smyth, T. Santos, and R. Bevan, "Sequence of Events Driven Automation of the Deep Space Network," Telecommunications and Data Acquisition 42-124, October-December 1995.

[11] Mark L. James, Lydia P. Dubon, "An Autonomous Diagnostic and Prognostic Monitoring System for NASA's Deep Space Network," Proceedings of the 2000 IEEE Aerospace Conference, Big Sky, MT, March 2000.

[12] S. Minton and M. Johnston, "Minimizing Conflicts: A Heuristic Repair Method for Constraint Satisfaction and Scheduling Problems," Artificial Intelligence, 58:161-205, 1988.

[13] B. Pell, D. Bernard, S. Chien, E. Gat, N. Muscettola, P. Nayak, M. Wagner, and B. Williams, " An Autonomous Spacecraft Agent Prototype," Autonomous Robots, March 1998.

[14] T. Santos, P. Pechkman, "Automation Language For Managing Operations (ALMO) User Guide, Delivery 1.1, " January, 1999.

[15] A. Tate, B. Drabble and R Kirby, "O-Plan2: An Open Architecture for Command Planning and Control," Intelligent Scheduling (Eds. M. Fox & M. Zweben), Morgan Kaufmann, 1994.

[16] D. Wilkins Practical Planning: Extending the Classical AI Planning Paradigm, Morgan Kaufmann, 1994.

[17] M. Zweben, B. Daun, E. Davis, and M. Deale, Scheduling and Rescheduling with Iterative Repair, in Intelligent Scheduling, Morgan Kaufmann, 1994.