

A Hierarchical Architecture for Resource Allocation, Plan Execution, and Revision for Operation of a Network of Communications Antennas

**S. A. Chien, R. W. Hill Jr., A. Govindjee, X. Wang,
T. Estlin, M. A. Griesel, R. Lam, and K. V. Fayyad**

Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109-8099

Abstract

This paper describes a hierarchical scheduling, planning, control, and execution monitoring architecture for automating operations of a worldwide network of communications antennas. We describe the network automation problem and current mode of operations. We then describe a three layer hierarchical architecture for automating network operations. In particular we describe the notion of plan/schedule generation, execution, and revision at each level. Finally, we describe the current state of deployment for each segment of the automation architecture.

Introduction

The Deep Space Network (DSN) (DSN, 1994) 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 unpiloted interplanetary spacecraft missions and support radio and radar astronomy observations in the exploration of the solar system and the universe. There are three deep space communications

complexes, located in Canberra, Australia, Madrid, Spain, and Goldstone, California. Each DSN complex operates four deep space stations -- one 70-meter antenna, two 34-meter antennas, and one 26-meter antenna. 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 require 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 requirements 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 whose goal is to support small, intelligent spacecraft requiring very few mission operations personnel.

The purpose of this paper is to describe an architecture for automating the process of capturing spacecraft data. In particular, we will describe how the components of the architecture transform a flight project service request into an executable set of DSN operations that fulfill the request through automated resource allocation, goal-driven plan generation, and plan execution and monitoring. The architecture was designed to assist the DSN in meeting the three NASA goals mentioned above. We successfully demonstrated a prototype of this architecture in February 1995 at NASA's experimental DSN station, DSS-13, on a series of Voyager tracks and

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

For further information contact:
steve.chien@jpl.nasa.gov.

efforts are currently underway to insert the technologies used in this demonstration into the operational DSN (Hill et al. 1995a, Hill et al. 1996).

This paper is organized in the following manner. We begin by characterizing the operation of the DSN at the time that this research was performed. Next we describe the architecture of our automated prototype -- we give a functional description of each of the components, which includes the Demand Access Network Scheduler (DANS) system for automated resource allocation, DPLAN (Estlin et al. 1996, Hill et al. 1995b), an automated procedure generation system, and LMCOA, a plan execution and monitoring system. In addition we provide examples of the inputs and outputs to each of the components to illustrate what occurs at each step in the process of capturing spacecraft data. Next, we describe the results of the technology demonstration at DSS-13 as well as ongoing efforts to incorporate this technology into the operational DSN. Finally, we describe aspects of plan execution and plan revision in the various components and levels of the DSN antenna operations automation architecture¹.

How the DSN Operates

Network Preparation at the Network Operations Control Center

Figure 1 shows a partial map of some of the operational processes of the DSN (see (RET, 1995) for a more complete description of the DSN processes). The first stage is called Network Preparation which occurs at the Network Operations Control Center (NOCC). A flight project, e.g., the Voyager project, initiates Network Preparation by sending a request for the DSN to track a spacecraft involving specific tracking services. The DSN responds to the request by attempting to schedule the resources (i.e., an antenna and other shared equipment) needed for the track.

Along with this request, the project prepares a Sequence of Events (SOE) describing the time-ordered activities that should occur during the track. The SOE includes actions that the DSN should take, (e.g., begin tracking the project's spacecraft at 1200 hours), and it also includes events that will occur on the spacecraft being tracked (e.g., the spacecraft will change frequency or mode at a designated time). These events are important because they affect how the DSN provides the services. The project SOE is sent to the DSN, which expands the activities from high level descriptions (e.g., begin tracking

the spacecraft) into a finer level of detail for use by the operations personnel at the deep space station. This expanded version is sent to the Deep Space Station (DSS) or antenna.

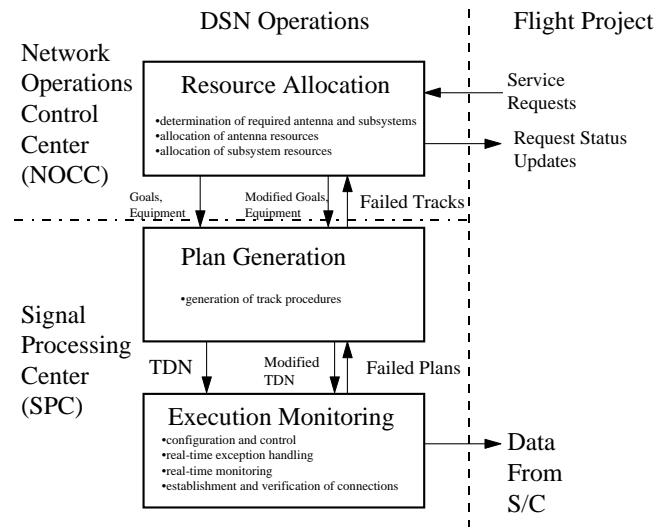


Figure 1: An Automation Oriented View of Deep Space Network Operations

Data Capture at the Signal Processing Center

The data capture process is performed by operations personnel at the deep space station - they determine the correct steps to perform to configure the equipment for the track, perform the actual establishment of the communications link, which we hereafter refer to as a 'link', and then perform the track by issuing control commands to the various subsystems comprising the link. Throughout the track the operators continually monitor the status of the link and handle exceptions (e.g., the receiver breaks lock with the spacecraft) as they occur. All of these actions are currently performed by human operators, who manually issue tens or hundreds commands via a computer keyboard to the link subsystems. The monitoring activities require the operator to track the state of each of the subsystems in the link (usually three to five subsystems), where each subsystem has many different state variables that change over time.

Automation of the DSN

In the last section we described the process for transforming a flight project service request into an executable set of DSN operations. As we have already pointed out, many of the steps of the described processes are intensely manual. We now describe an architecture for automating these processes in the manner described in

¹ For a further description of DSN antenna operations as an applications are for planning and plan execution see (Chien et al. 1996).

Figure 1. The specific systems and their functions are shown in Figure 1 -- DANS is applied to the resource scheduling process, DPLAN is used for automatically generating DSN operations procedures, and LMCOA/NMC automatically executes the operations procedures.

OMP/DANS: Automated Scheduling

The high level resource allocation problem for the smaller DSN antennas is currently handled by the OMP scheduler. In the future, an evolution of OMP called the Demand Access Network Scheduler (DANS) scheduling system is designed to address rescheduling for the entire DSN (e.g., all antennas and antenna subsystems). OMP accepts generalized service requests from spacecraft projects of the form "we need three 4-hour tracks per week" and resolves conflicts using a priority request scheme to attempt to maximize satisfaction of high priority projects. OMP deals with schedules for NASA's 26-meter subnet involving thousands of possible tracks and a final schedule involving hundreds of tracks.

The DSN scheduling problem is further complicated by three factors: (1) context-dependent priority; (2) subsystem allocation; and (3) the possibility of reducing the length of the tracks. DSN track priorities are context dependent in that they are often contingent on the amount of tracking the project has received so far in the week. For example, a project might have priority 3 to get 5 tracks, priority 4 to get 7 tracks and priority 6 to get 9 tracks (where lower priorities represent more important tracks). This reflects that 5 tracks are necessary to maintain spacecraft health and get critical science data to ground stations; 7 tracks will allow a nominal amount of science data to be downlinked; and 9 tracks will allow for downlinking of all science data (e.g., beyond this level additional tracks have little utility). An important point is that specific tracks are not labeled with these priorities (e.g., the project is allowed to submit 5 tracks at priority 3, 2 at priority 4 and so on). Rather when considering adding, deleting, or moving tracks the scheduler must consider the overall priority of the project in the current allocation context. In addition to allocating antennas, DSN scheduling involves allocating antenna subsystems which are shared by each Signal Processing Center (such as telemetry processors, transmitters and exciters). Allocating these complicates the scheduling problem because it adds to the number of resources being scheduled and certain subsystems may only be required for parts of the track. Finally, the DSN scheduling problem is complicated by the fact that the track duration can be relaxed. For example, a project may request a 3 hour track but specify a minimum track time of 2 hours. When evaluating potential resource conflicts the scheduler must consider the option of shortening tracks to remove resource conflicts. Currently OMP and DANS use a

linear weighting scheme in conjunction with a modified SIMPLEX algorithm to trim tracks in accordance with prioritizations.

DPLAN: Automated Procedure Generation

The automated track procedure generation problem involves taking a general service request (such as telemetry - downlink of data from a spacecraft) and an actual equipment assignment (describing the type of antenna, receiver, telemetry processor, and so on), and generating the appropriate partially ordered sequence of commands (called a Temporal Dependency Network or TDN; see Figure 3) for creating a communications link to enable the appropriate interaction with the spacecraft. The DSN Antenna Operations Planner (DPLAN) uses an integration of AI Hierarchical Task Network (HTN) and partial order operator-based planning techniques to represent DSN antenna operations knowledge and to antenna operations procedures on demand from the service request and equipment assignment.

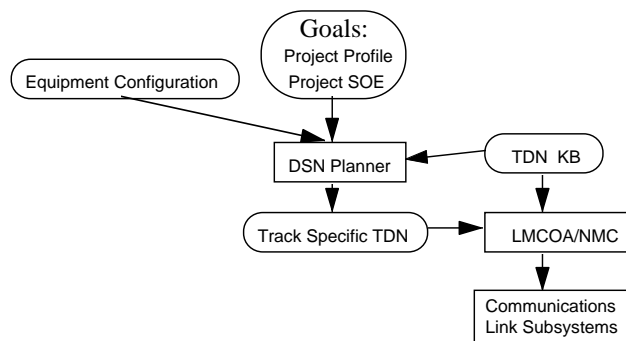


Figure 2: DPLAN and LMCOA/NMC Inputs and Outputs

The DPLAN planner uses high level track information to determine appropriate steps, ordering constraints on these steps, parameters of these steps to achieve the high level track goals given the equipment allocation. In generating the TDN, the planner uses information from several sources (see Figure 2):

Project SOE: The project sequence of events specifies events from the mission/project perspective. As a result, the project SOE contains a great deal of information regarding the spacecraft state which is relevant to the DSN track, as well as a large amount of spacecraft information unrelated to DSN operations. Relevant information specified in the project SOE includes such items as the one-way light time (OWLT) to the spacecraft, notifications of the beginning and ending times of tracks, spacecraft data transmission bit rate changes, modulation index changes, and carrier and subcarrier frequency changes.

Project profile: This file specifies project specific information regarding frequencies and pass types. For

example, the Project SOE might specify frequency = HIGH, and the project profile would specify the exact frequency used. The project profile might also other signal parameters and default track types.

TDN KB: The Temporal Dependency Network (TDN) knowledge base (Fayyad & Cooper, 1992, Fayyad et al. 1993) stores information on the TDN blocks available for the DSN Planner and LMCOA to use. This knowledge base includes information regarding preconditions, postconditions, directives, and other aspects of the TDN blocks. It also includes information on how to expand the block parameters and name into the actual flatfile entry in a TDN.

Equipment Configuration: This details the types of equipment available and unique identifiers to be used to specify the exact pieces of equipment to be used in the track. These include the antenna, antenna controller, the receiver, and so on.

DPLAN uses the SOE and project profile to determine the overall goals of the track. The DSN planner reduces the high level track goals into executable steps by applying knowledge about how to achieve specific combinations of track goals in the context of specific equipment combinations. This information is represented in the form of task reduction rules, which detail how a set of high level goals can be reduced into a set of lower level goals in a particular problem-solving context. Each task reduction rule rigorously details the scope of its expertise in terms of track and equipment combinations. The information of scope of applicability of the rule can be considered in terms of a track goal hierarchy and equipment goal hierarchy, where the rule applies to all contexts below the rule in the relevant hierarchy (all specializations of its scope).

Using this problem specification, the DSN planner then uses task reduction planning techniques (also called hierarchical task network or HTN) (Erol et al. 1994) and operator-based planning techniques (Pemberthy & Weld 1992) to produce a parameterized track-specific TDN to be used to conduct the track. This track-specific TDN and the SOE can then be used by LMCOA/NMC to operate the actual antenna to conduct the requested antenna track.

LMCOA/NMC: Automated Procedure Execution

The automated execution component uses the TDN (Figure 3) generated by DPLAN to perform the actual track and is responsible for monitoring the execution of the TDN. Previously an experimental system called the Link Monitor and Control Operator Assistant (LMCOA) was developed to automated this process; currently the key concepts and capabilities of the LMCOA system are being transferred to the operational DSN in the delivery of the Network Monitor and Control (NMC) Automation

Engine. For track operations, automated procedure execution and execution monitoring involves ensuring that the expected conditions and subsystem states are achieved, certain types of closed-loop control and error recovery are performed in a timely fashion, and the correct dispatching of commands to the subsystems controlling the link occurs. LMCOA uses an operator-based representation of the TDN to represent necessary and desired conditions for execution of procedures and tracks relevant subsystem state.

The LMCOA performs the operations procedures for a tracking activity by executing a Temporal Dependency Network (TDN), which is a procedure that is automatically generated by DPLAN, as described in the last section. DPLAN composes the TDN so that it contains the procedures (TDN blocks) needed for a specific tracking activity, and it orders them according to its knowledge of the dependencies that are defined among the blocks as well as by what it knows about the pre- and postconditions of the blocks. The knowledge about interblock dependencies and about block pre- and postconditions is passed to the LMCOA, whose task it is to execute the end-to-end procedure. The LMCOA receives the TDN in the form of a directed graph, where the precedence relations are specified by the nodes and arcs of the network. The blocks in the graph are partially ordered, meaning that some blocks may be executed in parallel. Temporal knowledge is also encoded in the TDN, which includes both absolute (e.g. Acquire the spacecraft at time 02:30:45) and relative (e.g. Perform step Y 5 minutes after step X) temporal constraints. Conditional branches in the network are performed only under certain conditions. Optional paths are those which are not essential to the operation, but may, for example, provide a higher level of confidence in the data resulting from a plan if performed. More details about TDNs are provided in (Fayyad & Cooper 1992).

To execute a TDN, LMCOA performs the following functions: (1) it loads the parameterized TDN into the Execution Manager; (2) it determines which TDN blocks are eligible for execution and spawns a process for each TDN block; (3) it checks whether the preconditions of each TDN block have been satisfied; (4) once the preconditions are satisfied, it issues the TDN block commands; and (5) it verifies whether the commands had their intended effects on the equipment. The Operator interacts with the LMCOA by watching the execution of the TDN; the operator can pause or skip portions of the TDN, check the status of commands within individual blocks, and provide inputs where they are required. When part of a TDN fails, the LMCOA supports manual recovery by the operator by highlighting the point of failure and providing information about the preconditions or postconditions that failed to be satisfied.

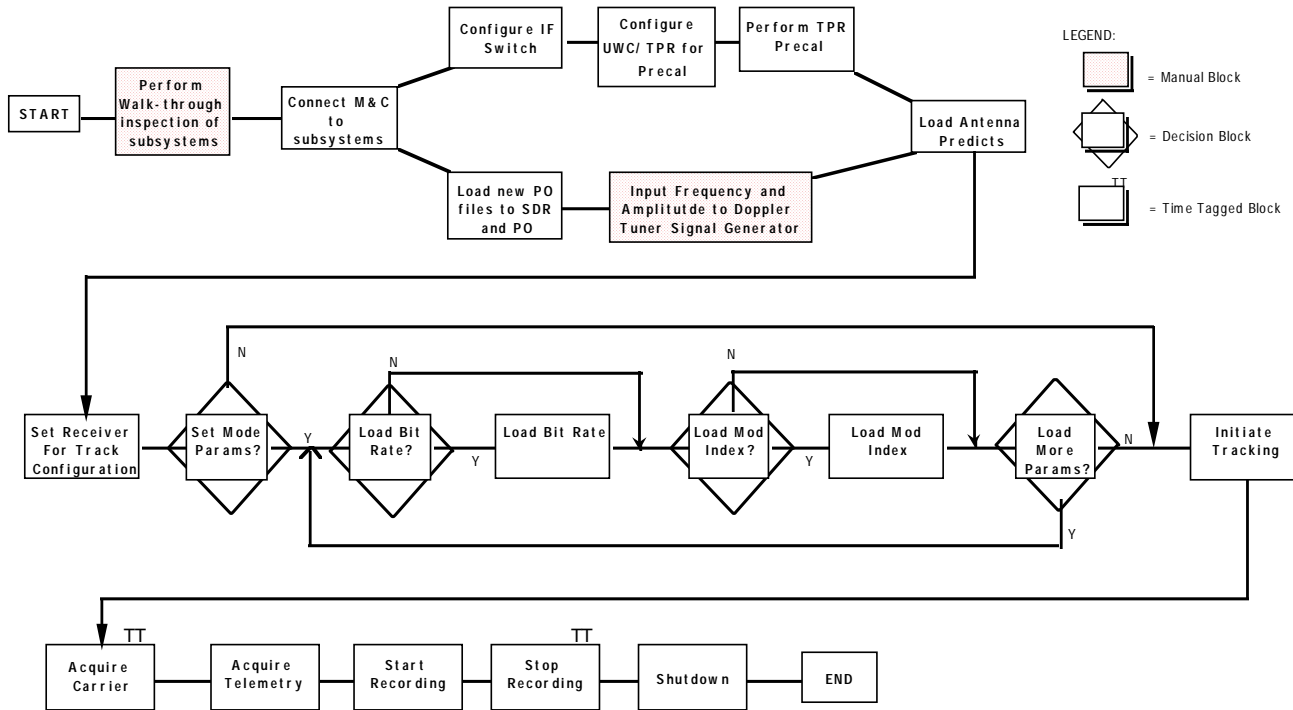


Figure 3: Temporal Dependency Network for Voyager Track

Results

In February 1995 a comprehensive demonstration was conducted to validate the concept of integrating and using the AI software described in the preceding paragraphs to track a spacecraft with the DSN. In the demonstration, DPLAN generated the TDN shown in Figure 3 for a Voyager Telemetry Downlink track using the equipment configuration at Deep Space Station 13 in Goldstone, California, which included a 34-meter beam-wave guide (BWG) antenna and a telemetry processor. The TDN generated by DPLAN was successfully executed by LMCOA--a communications link was established with Voyager and the 34-meter BWG antenna tracked the spacecraft, with minimal human control. As a result of this demonstration, DPLAN and the concepts implemented in LMCOA are currently being transferred and implemented in the Network Control Project (NCP), which will replace two major DSN subsystems--the Monitor and Control (M&C) subsystem and the Network Planning and Preparation (NPP) subsystem.

Plan Execution and Replanning in DSN Antenna Operations

In this section we describe the aspects of plan execution and replanning from the perspective of each of the three components involved in DSN Antenna Operations:

resource allocation, procedure generation, and execution monitoring.

At the high level of resource allocation, schedule execution does not involve execution monitoring. However, rescheduling is often necessary due to: equipment outages, last minute track requests, last minute changes to scheduled tracks, and atmospheric conditions impact on tracking capabilities. Rescheduling can occur in two ways: (1) it can be initiated top-down due to a change to a previously scheduled track or addition of another request; and (2) it can occur bottom-up in that equipment outages can occur or tracks can fail necessitating rescheduling. In the event of a new or modified request, DANS uses localized branch and bound search to consider alternative methods for satisfying the new request. This search uses as its bounding function a disruption cost measure which accounts for the overhead involved in moving already scheduled tracks and also a satisfaction measure accounting for what level of requests have been satisfied. Because we use branch and bound techniques DANS can guarantee that it will provide a reschedule optimal with respect to the combined disruption and satisfaction cost function

In the event of a change in equipment availability DANS first updates all resource timelines to reflect the new resource level. Then, depending on the size of the change it has two options. First, if the change is localized DANS can perform branch and bound search to re-evaluate requests in light of the new equipment situation. However, if the change is too large in scope this search is intractable. For example, if an antenna unexpectedly goes down for a

several day period the cascading effect on tracks can be quite great and thus rule out exhaustive search techniques. In these cases DANS can instead first performs prioritized pre-emption to remove low-priority tracks to remove conflicts (by removing the lowest priority tracks participating in each conflict) and then re-evaluate project requests. This approach requires far less search but can produce suboptimal results (with respect to the twin goals of minimizing disruption and maximizing request satisfaction).

At the level of track procedure generation DPLAN is also required to replan during the course of typical antenna operations. Replanning occurs in two general cases. First, after a plan has been generated, the objectives may change. Often shortly prior to or during a track a project may submit a request to add services to a track. These correspond to additional goals to be incorporated into the track plan. In the case of goals added before the track actually begins, DPLAN addresses this problem by adding these additional unachieved goals to the current plan and restarting the planning process with this single parent plan. This method is incomplete in theory because the planner may have made choices which are incompatible with the new goals. However, for the specific sets of goals and domain theories related to antenna operations we have examined we have been able to use encodings for which completeness has not been a problem. But this is an area of current work. Another area of current work is replanning in the case where goals are added during actual track execution. One approach to dealing with this would be to allow the planner the ability to backtrack or repair the current plan so as to adapt to the current situation. The planner might do this using set of plan modification operators.

Another case for replanning for DPLAN is due to dynamism. After a plan has been generated, a block (plan step) may fail, a piece of equipment may require resetting (due to general unreliability), or a piece of equipment may be fail or be pre-empted by a higher priority track. In the case of a simple plan step failure DPLAN simply calls for re-execution of the block. If a piece of equipment requires resetting, DPLAN has knowledge describing which achieved goals are undone and require re-establishment. DPLAN then uses a replanning technique (Wang & Chien 1996) which re-uses parts of the original plan as necessary to re-achieve the undone goals. This technique takes advantage of the fact that the original plan begins from a state which is equivalent to resetting all of the subsystems.

The plan execution component, either LMCOA or the NMC automation engine, performs the lowest level of plan execution and control. The NMC engine is responsible for closed loop monitor and control functions required by antenna and related subsystem operation as well as execution monitoring and subsystem coordination. The NMC engine takes the plan generated by DPLAN and performs the specified steps - expanding each plan step

(called a TDN block) into the appropriate parameterized procedure implemented in a scripting language. The scripts may perform monitoring, control, and communication-type lower level directives. Additionally, the scripts often embody such plan execution and recovery methods such as closed loop control, retries, and cognizant failure. However, all of these are within the scope of a single TDN block. In the event of a block procedure being unable to achieve its objectives, the block will report failure and LMCOA or NMC will correspondingly report failure and the planner will be notified that replanning is necessary.

Comparison to Other Work

There are a number of existing systems which also integrate scheduling, planning, control, and execution monitoring. We do not attempt to review them all, but focus on a few representative systems. To begin with, the main distinction between this architecture and other work is the hierarchical structure and the complexity of the DSN antenna operations domain.

Brooks' subsumption architecture (Brooks, 1986) contains no hierarchy of planning, scheduling, or control. This type of architecture has been used for mobile robot navigation, where replanning and rescheduling is a more constrained problem as compared to antenna operations which must schedule and plan for multiple resources (antennas and subsystems), and with both hard and soft temporal constraints.

CIRCA (Musliner et al. 1993) has a three-tiered architecture comprised of a planner, scheduler, and an executor which interacts with the environment through actuators and sensors in a mobile robot navigation domain. CIRCA does planning then scheduling, versus the DSN automation architecture which must first schedule and then plan. CIRCA's scheduling enforces hard real-time constraints, but returns failure if it cannot meet the time constraints. DANS/OMP, on the other hand, enforces hard real-time constraints, but always return a schedule, by using the priority scheme which maximizes the number of project requests that it accommodates. If some project requests cannot be accommodated, DANS/OMP will still return a schedule, even though it is sub-optimal.

$_3T$ (Bonasso et al. 1997) is a three-tiered architecture with a planner, sequencer, and a reactive skills module which interacts with the environment. Planning occurs hierarchically before sequencing, unlike the architecture which we describe in this paper which does scheduling then planning. The sequencer in $_3T$ is a RAP (Firby, 1989) interpreter which encodes all the timing information within the RAPs. DANS/OMP does not use RAPs, and uses a more complex algorithm to schedule the projects' requests. Unlike the DSN automation architecture, in $_3T$ all three of its tiers do not need to be used for a given task. In the DSN domain necessarily scheduling, then planning, then control

and execution must happen for successful antenna operations.

ATLANTIS (Gat, 1992) is also a three-tiered architecture, similar to T_3 . It is comprised of a controller which acts at the lowest reactive level, a sequencer which is a special-purpose operating system based on the RAP system, and a deliberator which does planning and world modeling. In ATLANTIS, it is the sequencer which does the brunt of the work; the deliberator is under the control of the sequencer. In fact, the deliberator's output is merely used as advice by the sequencer, and the entire system is able to function without the deliberator, if necessary. In the DSN automation architecture, as mentioned above, scheduling occurs hierarchically before planning; both steps are necessary. Also, there is a control and execution tier which is separate from the scheduling tier, unlike ATLANTIS which combines sequencing with control.

TCA (Simmons, 1994) has no real tiers, but many distributed modules working with a central control module via message-passing. There is no hierarchy that sets up schedules or plans; TCA operates by setting up a task tree instead.

AuRA (Arkin, 1989, Arkin and Balch, 1997) has three-tiers: planning, sequencing, and execution for use in mobile robot navigation. Its sequencer simply traverses a FSA expression of a plan, unlike the more powerful algorithms used for scheduling in DANS/OMP. Also, AuRA first plans and then sequences, whereas the DSN automation architecture first schedules, then plans.

The Cypress (Wilkins et al. 1995) architecture has plan and execution modules which operate asynchronously. There is also an uncertainty reasoning module which communicates with both the plan and execution modules. The DSN Automation architecture's scheduling, plan and execution modules can operate asynchronously, but there is no separate uncertainty reasoning module. Each tier handles uncertainty independently. Cypress is also not truly a hierarchical architecture and has no scheduling component. The military domain that Cypress has been used for is fairly complex, but since there is no scheduling component, Cypress doesn't tackle as comprehensive a problem as that described in this paper.

Both SOAR (Laird et al. 1987) and Guardian (Hayes-Roth, 1995) are general reasoning systems that can be adapted to a given task environment. The algorithms of the planner and the scheduler in the DSN automation architecture could be applied to a number of domains. The execution tier in our architecture, though, is particular to the antenna operations domain. Guardian does not have a hierarchical architecture, but uses a blackboard architecture with one module devoted to scheduling, planning, and control. SOAR also collapses all the tiers into a single mechanism.

The DSN automation architecture uniquely combines a scheduler, planner, and execution module to automate a

complex domain with many conflicting, hard constraints, handling replanning and rescheduling as necessary. The systems which have been designed for mobile robot navigation do not operate in as complex a domain as the DSN antenna operations domain. Examining the general reasoning systems, these are not hierarchically organized into separate planning, scheduling, and execution tiers. This hierarchical organization is a necessary part of the DSN antenna operations domain. The DANS/OMP scheduler uses more powerful algorithms than any of the other described systems' schedulers or sequencers. Unlike most of these systems, in the DSN antenna operations domain, it is necessary to first schedule and then plan, rather than plan and then schedule. Lastly, during execution, none of the other systems described appear to be capable of communicating with as large a set of external equipment as there are in the DSN antenna operations domain, monitoring for possibly multiple antenna or subsystem failures.

Conclusions

This paper has described a hierarchical scheduling, planning, control, and execution monitoring architecture for automating operations of a worldwide network of communications antennas. This architecture consists of three levels: resource allocation, track plan generation, and track execution. We then described aspects of plan execution and replanning at each of these three levels - primarily due to changing objectives and dynamism. This automation architecture has been demonstrated in an actual operations setting and is in the process of being fielded at operational DSN sites.

References

- (Arkin, 1989) R. Arkin. Motor schema-based mobile robot navigation. *International Journal of Robotics Research*, 8(4), 1989.
- (Arkin and Balch, 1997) R. Arkin and T. Balch. AuRA: Principles and Practice in Review. to appear in *Journal of Experimental and Theoretical Artificial Intelligence*, Vol. 9, No. 2, 1997.
- (Bonasso et al. 1997) R.P. Bonasso, R.J. Firby, E. Gat, D. Kortenkamp, D.P. Miller, and M.G. Slack. Experiences with an Architecture for Intelligent, Reactive Agents. to appear in *Journal of Experimental and Theoretical Artificial Intelligence*, March 1997.
- (Brooks, 1986) R. Brooks. A Robust Layered Control System for a Mobile Robot. *IEEE Journal of Robotics and Automation*, 2(1), 1986.
- (Chien et al. 1996) S. A. Chien, R. W. Hill Jr., X. Wang, T. Estlin, K. V. Fayyad, and H. B. Mortensen, "Why Real-

world Planning is Difficult: A Tale of Two Applications," in *New Directions in AI Planning*, M. Ghallab and A. Milani, ed., IOS Press, Washington, DC, 1996, pp. 287-298.

(DSN, 1994) Deep Space Network, Jet Propulsion Laboratory Publication 400-517, April 1994.

(Erol et al. 1994) K. Erol, J. Hendler, and D. Nau, "UMCP: A Sound and Complete Procedure for Hierarchical Task Network Planning," *Proceedings of the Second International Conference on AI Planning Systems*, Chicago, IL, June 1994, pp. 249-254.

(Estlin et al. 1996) T. Estlin, X. Wang, A. Govindjee, and S. Chien, "DPLAN Deep Space Network Antenna Operations Planner Programmers Guide Version 1.0," JPL Technical Document D-13377, Jet Propulsion Laboratory, California Institute of Technology, February 1996.

(Fayyad & Cooper, 1992) Fayyad, K. E. and L. P. Cooper. "Representing Operations Procedures Using Temporal Dependency Networks," SpaceOps '92, Pasadena, CA, November 1992.

(Fayyad et al. 1993) K. Fayyad, R.W. Hill, Jr., and E.J. Wyatt. "Knowledge Engineering for Temporal Dependency Networks as Operations Procedures." *Proceedings of AIAA Computing in Aerospace 9 Conference*, 1993, San Diego, CA.

(Firby, 1989) R.J. Firby. Adaptive Execution in Complex Dynamic Worlds. PhD Thesis, Yale University, 1989.

(Gat, 1992) E. Gat. Integrating planning and reacting in a heterogeneous asynchronous architecture for controlling real-world mobile robots. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 1992.

(Hayes-Roth, 1995) B.Hayes-Roth. An Architecture for Adaptive Intelligent Systems. *Artificial Intelligence*, 72, 1995.

(Hill et al. 1996) R. W. Hill, Jr., S. A. Chien, and K. V. Fayyad, "Automating Operations for a Network of Communications Antennas," *Proceedings of the 1996 IASTED International Conference on Artificial Intelligence, Expert Systems, and Neural Networks*, Honolulu, HI, August 1996.

(Hill et al. 1995a) 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.

(Hill et al. 1995b) Hill, R.W., Jr., S. Chien, C. Smyth and K. Fayyad, "Planning for Deep Space Network Operations" *Proceedings of the 1995 AAAI Spring Symposium on Integrated Planning Applications*, Palo Alto, CA, 1995, AAAI Press.

(Laird et al. 1987) J.E. Laird, A. Newell, and P.S. Rosenbloom. SOAR: An Architecture for General Intelligence. *Artificial Intelligence*, 33(1), 1987.

(Musliner et al. 1993) D.J. Musliner, E. Durfee, and K. Shin. CIRCA: A cooperative, intelligent, real-time control architecture. *IEEE Transactions on Systems, Man and Cybernetics*, 23(6), 1993.

(Pemberthy & Weld 1992) J. S. Pemberthy and D. S. Weld, "UCPOP: A Sound Complete, Partial Order Planner for ADL," *Proceedings of the Third International Conference on Knowledge Representation and Reasoning*, October 1992.

(RET, 1995) Final Report of the Services Fulfillment Reengineering Team, JPL Interoffice Memorandum, March 14, 1995.

(Simmons, 1994) R. Simmons. Structured Control for Autonomous Robots. *IEEE Transactions on Robotics and Automation*, Vol. 10, No. 1, February 1994.

(Wang & Chien 1996) X. Wang and S. Chien, "Replanning for the Deep Space Network (DSN) Antenna Operations Planner: Preliminary Report," JPL Technical Document D-13388, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, January 1996.

(Wilkins et al. 1995) D.E. Wilkins, K.L. Myers, J.D. Lowrance, and L.P. Wesley. Planning and Reacting in Uncertain and Dynamic Environments. *Journal of Experimental and Theoretical Artificial Intelligence*, 7, 1995.