Automated Generation of Tracking Plans for a Network of Communications Antennas

S. Chien, A. Govindjee, T. Estlin[a], X. Wang[b], and R. Hill Jr.[c]
Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Drive, M/S 525-3660
Pasadena, CA 91109-8099 {steve.chien, anita.govindjee}@jpl.nasa.gov

Abstract

This paper describes the Deep Space Network Antenna Operations Planner (DPLAN) : a system that automatically generates antenna tracking plans for a set of highly sensitive radio science and telecommunications antennas. DPLAN accepts as input an equipment configuration and a set of requested antenna track services. The system then uses a knowledge base of antenna operations procedures to produce a plan of activities that will provide the requested services using the allocated equipment. DPLAN produces this plan using an integration of hierarchical task network (HTN) and operator-based planning. A prototype of the DPLAN system was successfully demonstrated in February 1995 at NASA's experimental DSN station, DSS-13, on a series of Voyager tracks. Based on this successful demonstration, DPLAN is being considered for inclusion in the larger Network Monitor and Control (NMC) upgrade underway projected to save NASA over $9 million per year in operations costs.

Introduction*

The Deep Space Network (DSN) [6] was established in 1958 and since 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 to support radio and radar astronomy observations taken 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 function of the DSN is to receive telemetry signals from spacecraft, transmit commands that control spacecraft operating modes, generate the radio navigation data used to locate and guide a spacecraft to its destination, and acquire flight radio science, radio and radar astronomy, very long baseline interferometry (VLBI), 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. Currently, operation of the DSN communications complexes requires a high level of manual interaction with the devices involved in communications links with spacecraft. Recently, 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.

This paper describes the Deep Space Network Antenna Operations Planner (DPLAN), which automatically generates plans for individual antenna tracks based on requested services and equipment allocation. The DPLAN system is one element of a far-reaching effort to upgrade and automate DSN operations in order to achieve the three NASA goals mentioned above. A prototype of the DPLAN system was successfully demonstrated in February 1995 at NASA's experimental DSN station, DSS-13 [12,13], on a series of Voyager tracks. Due to this successful demonstration, DPLAN is being considered for inclusion as part of the Network Monitor and Control (NMC) upgrade to DSN stations. The NMC upgrade is projected to enable automation saving NASA over $9 million per year in DSN operations costs.

This paper is organized as follows. We begin by describing the problem of generating Deep Space Network tracking plans. Next, we describe the DPLAN system, including: (1) the track plan generation problem; (2) an overview of artificial intelligence hierarchical task network (HTN) and operator-based planning; (3) the DPLAN system; and (4) an example of operation. Finally, we

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

This paper appears in Proceedings of the 1997 Conference on Innovative Applications of Artificial Intlligence, Providence, RI, AAAI Press.

a - Current Address: Department of Computer Science, University of Texas, Austin, TX, esltin@cs.utexas.edu

b - Current address: Rockwell Science Center, Palo Alto, CA, mei@rpal.rockwell.com

c - Current address: USC/ISI, Marina Del Rey, CA, hill@isi.edu

describe current efforts to deploy the DPLAN system in the operational DSN and other areas of current work.

Track Plan Generation: Problem Description

Each day, at sites around the world, NASA's Deep Space Network (DSN) antennas and subsystems are used to perform scores of tracks that support earth orbiting and deep space missions. Due to the complexity of this equipment, the large set of communications services (in the tens), and the large number of supported equipment configurations (in the hundreds), correctly and efficiently operating this equipment to fulfill tracking goals is a daunting task. An additional requirement is that the antenna operations knowledge embodied in the system must be easily understandable and maintainable. This requirement must also be met as equipment upgrades, services, protocols, and software changes evolve.

The Deep Space Network Antenna Operations Planner (DPLAN) is an automated planning system developed by the Jet Propulsion Laboratory (JPL) to automatically generate antenna tracking plans that satisfy DSN service requests. In order to generate these antenna operations plans, DPLAN uses a number of information sources, including: the project generated service request, the spacecraft sequence of events, the track equipment allocation, and an antenna operations knowledge base.

The project service request represents the basic communications services requested during the track (telemetry/downlink, commanding/uplink, ranging (uplink and downlink), etc.). The project sequence of events indicates the relevant spacecraft mode changes (such as transmission bit rate changes, modulation index changes, etc.). The equipment allocation dictates the antenna and subsystem configuration available for the track. The antenna operations knowledge base provides necessary information on the requirements of antenna operation actions. In particular, this information dictates how these actions can be combined to provide essential communications services.

Artificial Intelligence Planning Techniques

AI planning researchers have developed numerous approaches to the task of correct and efficient planning. Two main planning methods are *operator-based* planning and *hierarchical task network* (HTN) planning. DPLAN uses a combination of both these approaches, exploiting the advantages of each.

Both HTN and operator-based planners typically construct plans by searching through a plan-space. However, they differ considerably in how they search. HTN planners specify plan modifications in terms of flexible task reduction rules and work in a forward-chaining, top-down fashion. In contrast, operator-based planners work in a backward-chaining manner by taking a given goal and attempting to resolve its preconditions. Operator-based planners perform all reasoning at the lowest level of abstraction and provide a strict semantics for defining operator definitions.

An HTN planner [7] uses task reduction rules to decompose abstract goals into lower level tasks. HTN planners can encode many different types of information into task reductions. By defining or not defining certain reduction refinements, the designer can direct the planner towards particular search paths in certain contexts. The user can also directly influence the planner by explicitly adding an ordering constraint or goal protection that would not strictly be derived from goal interaction analyses. Search-control knowledge can also be encoded by writing explicit action sequences to achieve goals, thereby avoiding considerable search.

In contrast, an operator-based planner [15][1] reasons at a single level of abstraction -- the lowest level. Actions are strictly defined in terms of preconditions and effects. Plans are produced through subgoaling and goal interaction analyses. In this framework, all plan constraints (protections, ordering, and codesignation) are a direct consequence of goal achievement and action precondition and effect analysis. Thus, an operator-based planner generally has a strict semantics grounded in explicit state representation, *i.e.* defining what is and is not true in a particular state (or partial state).

The DPLAN planner combines these two planning methods, utilizing the advantages of each. For instance, an operator-based planner requires a very rigid representation which is both a strength and a weakness. It is an advantage in that there is usually one obvious method of encoding each subproblem. However this rigidity can also make certain aspects of a problem difficult to represent. Known ordering constraints and operator sequences can be difficult to encode if they cannot easily be represented in terms of preconditions and effects. Such constraints can and are often forced by adding “dummy” preconditions - in which an operator A is made to precede an operator B by forcing A to achieve a condition C for B. However this solution can often create a misleading representation in that other occurrences of A don’t require C to be true. An HTN planner, on the other hand, allows the easy representation of known ordering constraints. Domain information, such as constraints, is easily added to domain rules in the HTN framework. This type of representation allows the user to easily direct the planner’s search by explicitly defining items such as ordering constraints and goal protections.

By using a combination of both HTN planning and operator-based planning we can easily direct search and can define knowledge in an understandable top-down fashion. In a hybrid representation we also have the ability to define knowledge in the more structured operator-based fashion when appropriate.

DPLAN’s algorithm is a combination of both hierarchical task network (HTN) planning techniques and operator-based planning techniques. In HTN planning, abstract actions such as "calibrate receiver" or "configure sequential ranging assembly" are decomposed into specific directives for specific hardware types. In operator-based planning, requirements of specific actions, such as "move antenna to point", are satisfied using means-end analysis, which matches action preconditions to effects and resolves any occurring ordering conflicts.

The DPLAN Planning Algorithm

The DPLAN planning algorithm uses a unique combination of the HTN and operator-based planning techniques discussed above. DPLAN operates by refining a set of input top-level goals into a set of low-level operational goals. Plans are represented by a three-tuple: <U,C,S> where U is a set of non-operational (or high-level) goals, C is a set of constraints, and S is a set of operational-goals. At the end of planning, U should be empty and the goals in S are returned as the final plan steps.

A overview of DPLAN algorithm is shown in Figure 1. The main inputs to DPLAN are: a set of high-level goals G, a set of decomposition rules R, and the set of all possible operational goals O. Search is implemented by keeping a queue of partial plans to be explored. Currently, plans are selected from the queue using a best-first heuristic; however, other search techniques could easily be employed. Step 1 and Step 2 of the main loop remove the best plan off the queue, and Step 3 checks if that plan is a solution. If no solution has been found then a new goal is selected for refinement in Step 4. Step 5 chooses a refinement strategy for that goal, and in Step 6, any new plans created through that strategy are inserted into the plan queue.

Algorithm DPLAN(G,R,O)
Initialize the plan queue Q := (<G,{}, {}>)
While Q is not empty and the resource bound has not been exceeded,

1. Select a promising plan P in Q using heuristics,
2. Remove P from Q
3. If P contains only operational-goals, then check context goals in P. If the context goals are achieved, return P. Otherwise goto 1.
4. Choose a non-operational goal **g** from U.
5. Refine **g**.
6. Insert any new plans generated by refinement into Q.

Figure 1 - The DPLAN Search Algorithm

A plan is considered a solution if two conditions are true. The first is that there are no non-operational goals left to be refined. The second condition is that all context goals have been achieved or are directly achievable in the current plan. Context goals are goals which were needed for applying a decomposition rule, but are supposed to be accomplished by some other part of the plan. If all context goals have been achieved, then the plan is returned as a success.

DPLAN can use several different refinement strategies to handle non-operational goals. There are two main types of goals in DPLAN: *activity-goals* and *state- goals*. *Activity-goals* correspond to operational or non-operational activities and are usually manipulated using HTN planning techniques. Operational activity-goals are considered primitive tasks that can be directly executed. Non-operational activity-goals must be further decomposed into operational ones through HTN reduction rules. *State-goals* correspond to the preconditions and effects of activity-goals, and are achieved through operator-based planning. State-goals that have not yet been achieved are also considered non-operational. Figure 2 shows the procedures used for refining these two types of goals. As soon as a refinement strategy is applied to an activity-goal or state-goal, it is removed from the list of non-operational goals.

If g is an Activity-Goal,

1. <u>Decompose</u>: For each decomposition rule r in R which can decompose g, apply r to produce a new plan P', If all constraints in P' are consistent, then add P' to Q.
2. <u>Simple Establishment</u>: For each activity-goal g' in U that can be unified with g simple establish g using g' and produce a new plan P'. If all constraints in P' are consistent, then add P' to Q.

If g is a State-Goal,

1. <u>Step Addition</u>: For each activity-goal effect that can unify with g, add that goal to P to produce a new plan P'. If the constraints in P' are consistent, then add P' to Q.
2. <u>Simple Establishment</u>: For each activity-goal g' in U that has an effect e that can be unified with g, simple establish g using e and produce a new plan P'. If all constraints in P' are consistent, then add P' to Q.

Figure 2 - Goal Refinement Strategies

DPLAN can also use additional domain information for more efficient and flexible planning. For instance, a planning problem can specify a list of static context facts. These facts represent operational goals that are always considered to be true. Such goals are easy for DPLAN to verify during planning and can help in pruning off search branches. Other possible inputs include sets of preconditions and effects for operational activities, a set of final goals that must be true in the plan solution, and a set of initial goals that are true at the beginning of planning. This information is not required for standard DPLAN operation, but can be very beneficial during planning.

An Example of DPLAN Representation

As mentioned in the preceding section, DPLAN uses several different types of knowledge to construct a plan. A main component of this knowledge is a set of decomposition rules. These rules specify how the planner can break down nonoperational activity-goals into lower-level operational goals. A sample rule for performing a telemetry antenna track is shown in Figure 3. This rule defines how the general telemetry operation is broken down into steps. The left-hand side (LHS) of a decomposition rule consists of a set of initial goals, and possibly, a number of other constraints that specify when the rule should be applied. All initial goals and specified constraints must be true in the current plan for the rule to be selected. The initial goals of a rule are the nonoperational goals that the rule "decomposes" into lower-level goals. The rule shown above has only one initial goal that checks if a *telemetry track-goal* is present in the current plan.[1]

The right-hand side (RHS) of a rule contains a set of new goals and constraints over those goals. Once a rule is applied, these new goals replace the LHS initial goals in the current plan. The RHS also contains ordering constraints and protections that specify information about the new goals. An ordering constraint specifies that two goals must be placed in a certain partial order in the final TDN. A protection specifies a causal link that exists between goals. This link explains how the effect of one goal achieves the precondition of another goal. Causal links must always be preserved in order to generate a correct plan. Ordering constraints and protections are added to the current plan and must always be kept consistent during planning. For instance, if an ordering constraint is somehow violated during planning, then the current plan is discarded, and the planner selects another plan from the queue to work on.

Sometimes there may be several different rules that can be used to decompose the same initial goal. For instance, in tracks for 70m antennas, there are several different methods for configuring a receiver depending on the type of receiver being used. To represent these different methods, there are several different rules that can be used to decompose the *perform-receiver-configuration* goal (which was asserted by the telemetry rule in Figure 3).

```
(decomprule default-telemetry-track
 lhs
    (initialgoals ((track-goal spacecraft-track telemetry
          ?track-id)))
 rhs
    (newgoals
         ((g1 (perform-antenna-controller- configuration
          ?track-id))
          (g2 (configure-metric-data-assembly ?track-id))
          (g3 (perform-microwave-controller-configuration
                ?track-id))
          (g4 (perform-receiver-configuration ?track-id))
          (g5 (perform-telemetry-configuration ?track-id))
          (g6 (move-antenna-to-point ?track-id))
          (g7 (perform-receiver-calibration ?track-id)))
     constraints
         ((before g1 g6)
          (before g7 g3)
          (before g4 g7))))
```

Figure 3 - Decomposition rule for telemetry track

[1] Other possible LHS constraints include additional goal conditions that must be present in the plan, context goals, and codesignation constraints, which check whether two variables can or cannot be unified.

Conversely, the utilization of goal schemas and operator-based planning techniques allows certain constraint information to be more easily expressed in the domain. Ordering constraints that are due to precondition-effect interactions are directly deduced during planning, instead of having to be explicitly listed by the user. In particular, ordering constraints that apply to very specialized goals, as opposed to very general ones, can be more easily expressed through precondition/effect schemas than through decomposition rules. For more information on the advantages and disadvantages of employing HTN and operator-based planning techniques for this type of domain see [4].

An Operations Example

In order to begin the planning process, DPLAN is provided with a problem specification that contains several lists of information. Specifically, each problem contains a list of decomposition goals, along with possible lists of initial state predicates, static state predicates, and final state predicates. A sample problem for performing telemetry and ranging with a 70m antenna is shown in Figure 4.

The *init-state* field specifies a list of propositions that are true in the initial state of the planner. For instance, as shown in Figure 4, the exciter drive is assumed to be off prior to when the track is performed. The *static-state* field specifies a list of propositions that are always true during planning (*i.e.* can never be deleted), and is commonly used to list equipment types available to the track. The *decompgoals* field holds the list of nonoperational goals that are to be broken down into lower-level goals through the use of decomposition rules. The *final-state* field is a list of propositions that must be true in the final plan.

```
(decompproblem TELEM70
   (init-state ((exciter-drive-off track1)
              (range-mode-off track1)
              (test-translator-off track1)))
   (static-state
      ((CCN-equipment-assignment track1 bstring1)
       (isa bstring1 type-B-telemetry-string)
       (CCN-equipment-assignment track1
            APA-70m)
       (isa APA-70m APA)
       (CCN-equipment-assignment track1 bvr1)
       (isa bvr1 BVR)
       (CCN-equipment-assignment track1 rec1)
       (isa rec1 REC)
       (CCN-equipment-assignment track1 ugc1)
       (isa ugc1 UGC)))
   (decompgoals
      ((perform-pre-cal track1)
       (track-goal spacecraft-track telemetry track1)
       (track-goal spacecraft-track ranging track1)))
   (final-state () ))
```

Figure 4 - Problem specification for a telemetry and ranging track

A final plan contains a large amount of information, including a list of operational goal names (corresponding to TDN blocks), a list of ordering constraints over those goals, and a list of annotations that describes how the plan was built (*i.e.*, what rules and operations were used). Currently the planner outputs this information in the following way. Three output files are created, a text output file, an annotation file, and a graph-input file. The text output file contains a textual listing of blocks and parameters where blocks are listed in a correct ordering (*i.e.*, blocks do no violate any plan ordering constraints). The annotation file contains a textual list of annotations describing the plan and how it was constructed. The graph-input file contains a list of node names and ordering constraints, which can be used to construct a graphical representation of the plan. See Figure 5 for an example of a plan (or TDN) that was generated for a problem specification such as that shown in Figure 4.

Application Use And Payoff

The DPLAN system was successfully demonstrated in February 1995 at NASA's experimental DSN station, DSS-13, on a series of Voyager tracks. Based on this successful demonstration, DPLAN is being evaluated for inclusion as part of the larger Network Monitor and Control (NMC) upgrade underway projected to save NASA over $9 million per year in operations costs.

The current DPLAN knowledge base for the planner currently supports the 34-meter (34m) and 70-meter (70m) antenna types at the DSN. All valid types of spacecraft passes for each antenna type are implemented in the knowledge base. Spacecraft passes include the following:

- *telemetry* : Telemetry is a downlink with the spacecraft where information is relayed from the spacecraft to the DSN station on earth.
- *ranging* : Ranging is a method of finding the distance between the spacecraft and the earth which requires both an uplink and a downlink to the spacecraft.
- *commanding*: Commanding is an uplink to the spacecraft where commands are sent from the DSN station to the spacecraft, which instructs the spacecraft to carry out given tasks.
- *VLBI ΔDOR:* VLBI (Very Long Baseline Interferometry) uses quasars --- distant space objects --- in order to determine the location of a spacecraft. A VLBI ΔDOR: (ΔDOR = Differential One-way Ranging) service provides information on the spacecraft's angular position by performing: simultaneous observations from two antenna stations of the spacecraft and a quasar, followed by a second observation of the spacecraft to gather doppler data. This data is then used to determine how to maneuver the spacecraft through space to its destination.
- *VLBI clock sync:* VLBI clock sync gives the instantaneous position of two stations relative to a quasar. This pass is performed in order to determine the rate of change of the clocks at the two DSN stations.
- *radio science:* For radio science, the antenna station is used to gather Radio Frequency (RF) signal information from spacecraft transmissions or natural sources (such as a planet or star).

Not all antenna types perform all types of spacecraft passes. For example, the 34m STD (Standard) antenna is not used for any type of VLBI activity. For each of the antenna types all the types of spacecraft passes that the antenna is used for are covered in the DPLAN knowledge base:

- *34m BWG*: 34-meter Beam Wave Guide. Telemetry, commanding, and ranging.
- *34m STD:* 34-meter Standard. Telemetry, commanding, and ranging.
- *34m HEF:* 34-meter High Efficiency. Telemetry, commanding, ranging, VLBI ΔDOR, and radio science.
- *70m BVR:* 70-meter with Block IV or Block V Receiver. Telemetry, commanding, ranging, VLBI ΔDOR, VLBI clock sync, and radio science.

Generating a plan to make an antenna operational and ready to communicate with a given spacecraft is a complex process - requiring careful coordination of multiple pieces of equipment and subsystems.

The total number of rules in the knowledge base (covering all antenna and track types) is 197: 91 decomposition rules (average of 23 decomposition rules per antenna type) and 106 goal schemas. The knowledge base is modular and easily extended to accommodate new antenna types and new subsystems or equipment types. Also, as changes are made to existing antennas, equipment, and subsystems, the rules can easily be modified. For example, if a new type of antenna controller is added to the 34m-HEF antenna, then a new rule is simply added that configures the new antenna controller. Other rules which use the antenna controller rule do not need to be changed because of the decomposition structure of the knowledge base.

All the plans generated by the planner for the different antenna types and their valid spacecraft passes (including a majority of the multiple combinations of passes) have been verified by the DSN operator experts from all three of the DSN complexes: Goldstone, California (October 1995), Madrid, Spain (January 1996), and Canberra, Australia (May 1996). For example, the 34m-STD antenna can support telemetry, ranging, and commanding spacecraft passes, and any combination of those three types of passes. DPLAN generated all of the resulting 7 combinations of spacecraft passes (telemetry, ranging, telemetry & ranging, telemetry & commanding, etc.). These passes were then verified on paper by the various operator experts as being correct, executable plans in terms of the ordering of the TDN blocks and the inclusion (or exclusion) of sufficient and necessary TDN blocks.

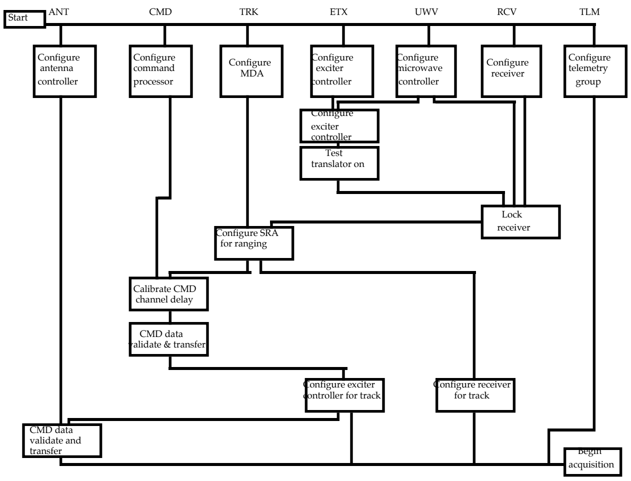

Figure 5 - Temporal Dependency Network for 34 M Beam Wave Guide Antenna Pre-Track for Telemetry, Commanding, and Ranging Services

More testing will occur during the integration phase. During integration, the plans generated by DPLAN are executed by the Automation Engine (AE), which fires scripts associated with each TDN block in the plan. The scripts execute 'operator directives' which turn on and off pieces of equipment, configure subsystems, move the antenna, etc.

A preliminary demonstration was successfully done that integrated the planner with the other elements that comprise the DSN automation. The planner successfully constructed a plan, which was then executed (in simulation) by the AE. This demonstration took place in December 1995. Further testing of the planner took place in August 1996 in a computer simulated antenna environment with simulated subsystems and equipment. DPLAN is currently being considered for inclusion into the NMC's AE for the D2 delivery which is scheduled to occur in August 1998. When fully integrated, the AE will call the planner to generate a given plan and then execute that plan, firing off the necessary scripts for the TDN blocks.

Maintenance

One key issue for DSN antenna operations is maintenance of the software systems necessary to automate DSN antenna operations. It was our experience that DSN operations personnel were quite confident that antenna operations could be automated - the key point was reducing the costs of maintaining expert knowledge regarding operations. In our development of the extensive DSN antenna operations knowledge base, we have collaborated with DSN antenna operation experts (operators and engineers) and it is expected that they will maintain and extend the antenna operations knowledge base. In this section we discuss several issues relevant to maintenance of the DSN antenna operations planner knowledge base including: representation for maintainability and comparisons to other automation representations considered by the DSN.

Representation for Maintainability

An important aspect of the DPLAN representation is that it allows for natural encoding of abstract objects and procedures (*e.g.* receiver calibration). By allowing decomposition rules to refer to abstract objects, changes to DSN procedures involve fewer knowledge base updates than if the knowledge base contained a large number of very specific rules. For instance, a change relating to a specific equipment type need not affect more general domain information. If a new receiver type called a BLOCK-VI receiver were added to the DSN equipment list, more general rules, such as the telemetry rule shown in Figure 3, would not need to be modified. Instead, only a few more specific rules need be constructed or edited. In this case, a new *configure-receiver* rule would be added. Therefore, many such changes would cause only a few specialized rules to be created or updated instead of causing numerous rules to be modified. Even with the current DSN goal to automate all TDN generation, the planning knowledge base must be constantly updated and verified. Fewer more general rules are cheaper to update and verify, and can thus support more efficient knowledge base maintenance.

Another benefit to this type of representation is that domain information is more easily understood. By keeping domain details separate from more general knowledge, it is easier for a user to understand the general aspects of an antenna track. For example, to understand the general steps of a telemetry operation, a user only has to view the main telemetry track decomposition rule. If more low-level knowledge is desired, such as how to operate a particular piece of equipment, the user could then search for rules that directly pertain to that equipment type.

Comparison to Scripts

One option considered by DSN personnel was to implement the higher level of track automation by a hierarchy of scripts. There would be scripts for general activities, such as calibrating a Block V Receiver in the context of a ranging track. This scripting approach can be viewed as similar to the HTN planning approach, but with two key differences. First, there is no explicit representation of the context in which a script will necessarily achieve the goal. The set of situations in which a script S is expected to work is represented only implicitly in the set of scripts which call S. The intended coverage, conditions, etc. are not explicitly represented as they are in HTN rules. The second difference is that the planner allows a "call by goal" usage in operator-based planning. In this way, the planner can invoke routines (or operators) based on the conditions it desires to achieve, and the planner will automatically detect and resolve any conflicting interactions with other activities. Not only does the planner representation allow for encoding of conditions and assumptions of when particular activities are appropriate (through conditions on HTN rules or preconditions on operators), it actually requires such definitions in order to operate correctly. Therefore it encourages correct documentation of operations requirements for all activities - which should lead to more maintainable operations procedures.

Comparison to End-to-End TDNs

Another option considered by DSN Operations was to simply encode end-to-end TDNs for each supported combination of the cross product between service requests and equipment allocation. Unfortunately, this option has several drawbacks. First, articulating all of the relevant knowledge in this format can be very tedious and prone to error. While generating the initial set of end-to-end TDNs, the expert operators said that they often found it difficult to keep all of the different TDNs straight. Second, this representation is not amenable to maintenance. If an equipment type is added or changed, it must be changed in every TDN that is relevant. The knowledge pertaining to the equipment type is not centralized in a set of rules or activity definitions as it is in the planning representation.

Discussion

In this section we discuss several issues relevant to the DSN planner including representing and reasoning about plan quality and replanning.

Representing and Reasoning about Plan Quality

Representing and reasoning about plan quality [11][16][18] is another key concern of DSN operations. Because of space constraints we only describe the most important aspects of reasoning about plan quality in the DSN antenna operations application - for further details the reader is referred to [3]. Since there is often more than one correct plan for a particular antenna operation, it is important for a planning system to be able to compare a set of final plans using user identified plan quality measures. There are a number of quality measures that can be emphasized during planning, including producing more robust, flexible and/or efficient plans. One important quality goal is to minimize the overall plan execution time. In particular, the time to setup (pre-calibration) and reset (post-calibration) the communications link can often be reduced. For instance, it can take up to two hours to manually pre-calibrate a DSN 70-meter antenna communications link for certain types of mission. By using a plan generated by DPLAN, this time can be reduced to approximately 30 minutes, where further reductions in set-up time are limited by physical constraints of the subsystems themselves.

Plan execution time is often significantly reduced by exploiting parallel path possibilities, especially where the control of multiple subsystems is involved. DPLAN currently uses the critical path length of a plan to help identify better plans. Critical path length is calculated using time information attached to a TDN block, which specifies the average time it should take to execute the block. By comparing critical path lengths of competing plans, DPLAN could choose a highly efficient final plan that will provide a minimal execution time. Minimizing plan execution time allows more data to be returned per operating time for the link.

Another issue for plan quality is plan flexibility. There certain standard TDN blocks that may be inserted into a plan at various points (such as transmission rate changes, etc.). If such commands are executed in the middle of an inflexible plan, it may not be possible to continue execution. Depending on the steps inserted, preconditions, postconditions, and time tags of other blocks may become invalid. Flexible plans that allow for the insertion of common steps while still retaining their applicability are greatly valued.

Replanning for Antenna Tracks

Additionally, DPLAN is 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 sometimes change. Often, shortly prior to or during a track, a project may submit a request to add services to the track. This request corresponds to additional goals that must be incorporated into the track plan. In the case where goals are added before the track actually begins, DPLAN adds these unachieved goals to the current plan and restarts the planning process. Unfortunately, this method is incomplete in theory because the planner may have previously made choices that are incompatible with the new goals. However, for the specific sets of goals and domain theories (related to antenna operations) that we have examined, we have been able to use encodings in which completeness has not been a problem. 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 a set of plan modification operators.

Another replanning issue is caused by 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 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 have been undone and require re-establishment. DPLAN then uses a replanning technique [20] that re-uses parts of the original plan to re-achieve the undone goals as necessary. This technique takes advantage of the fact that the original plan begins from a state that is equivalent to resetting all of the subsystems.

Conclusions

This paper has described the DSN Antenna Operations Planner (DPLAN) which automatically generates communications antenna tracking plans based on requested services and equipment allocation. DPLAN uses a knowledge base of information on tracking activities and a combination of artificial intelligence planning methods to generate appropriate tracking plans. We have also described the deployment status of the DPLAN system and outlined areas of current work including: representation and reasoning about plan quality, replanning, and representation to support maintainability. The DPLAN system was successfully demonstrated in February 1995 at NASA's experimental DSN station, DSS-13, on a series of Voyager tracks. Based on this successful demonstration, DPLAN is being evaluated for inclusion as part of the larger Network Monitor and Control (NMC) upgrade underway projected to save NASA over $9 million per year in operations costs.

References

[1] J.G. Carbonell, J. Blythe, O. Etzioni, Y. Gil, R. Joseph, D. Kahn, C. Knoblock, S. Minton, M.A. Perez, S. Reilly, M. Veloso, X. Wang, "Prodigy 4.0: The Manual and Tutorial," *Technical Report*, School of Computer Science, Carnegie Mellon University.

[2] 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.

[3] S. Chien, A. Govindjee, X. Wang, T. Estlin, R. Hill, Jr., "Automated Generation of Tracking Plans for a Network of Communications Antennas," *Proceedings of the 1997 IEEE Aerospace Conference*, Aspen, CO, February 1997.

[4] S. Chien, X. Wang, and T. Estlin, "Hierarchical Task Network and Operator-Based Planning: Competing or Complementary?," *JPL Technical Doc. D-13390*, NASA Jet Propulsion Laboratory, Pasadena, CA, January 1996.

[5] S. Chien, R. Lam, and Q. Vu, "Resource Scheduling for a Network of Communications Antennas," *Proceedings of the 1997 IEEE Aerospace Conference*, Aspen, CO, February 1997.

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

[7] 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.

[8] Fayyad, K. E. and L. P. Cooper. "Representing Operations Procedures Using Temporal Dependency Networks," *SpaceOps '92*, Pasadena, CA, November 1992.

[9] K. Fayyad, R.W. Hill, Jr., and E.J. Wyatt. "Knowledge Engineering for Temporal Dependency Networks as Operations Procedures." *Proc. AIAA Computing in Aerospace 9 Conf.* , 1993, San Diego, CA.

[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] 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.

[12] M. Williamson and S. Hanks, "Optimal Planning with a Goal-directed Utility Model," *Proceedings of the Second International Conference on AI Planning Systems*, Chicago, IL, June 1994, pp. 176-180.

[13] A. Perez and J. Carbonell, "Control Knowledge to Improve Plan Quality," *Proceedings of the Second International Conference on AI Planning Systems*, Chicago, IL, June 1994, pp. 323-328.

[14] Final Report of the Services Fulfillment Reengineering Team, *JPL Interoffice Memorandum*, March 14, 1995.

[15] X. Wang and S. Chien, "Replanning for the Deep Space Network (DSN) Antenna Operations Planner: Preliminary Report," *JPL Tech. Doc. D-13388*, Jet Propulsion Lab., Calif. Inst. of Tech., Pasadena, CA, January 1996.