An Automated Deep Space Communications Station*

Forest Fisher [1], Steve Chien, Leslie Paal,
Emily Law, Nasser Golshan, Mike Stockett

Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109-8099
firstname.lastname@jpl.nasa.gov

ABSTRACT

This paper describes an architecture being implemented for an autonomous Deep Space Tracking Station (DS-T). The architecture targets fully automated routine operations encompassing scheduling and resource allocation, antenna and receiver predict generation, track procedure generation from service requests, and closed loop control and error recovery for the station subsystems. This architecture is being validated by construction of a prototype DS-T station which will be demonstrated in two phases: down-link (March 98) and up-link/down-link(July 98).

INTRODUCTION

The Deep Space Network (DSN) [9] 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 unmanned interplanetary spacecraft missions and to support radio and radar astronomy observations taken in the exploration of space. 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.

This paper describes the Deep Space Terminal (DS-T), a prototype 34-meter deep space communications station under development which is intended to be capable of *fully autonomous, lights-out*, operations. In the DS-T concept, a global DSN schedule is disseminated to a set of autonomous DS-T stations. Each DS-T station operates autonomously, performing tracks in a largely independent fashion. When requested to perform a track, the DS-T station performs a number of tasks (at appropriate times) required to execute the track. First, the DS-T station uses appropriate spacecraft navigation ephemeris and predict generation software in order to produce necessary antenna and receiver predict information required to perform the track. Next, the DS-T station executes the pre-calibration process, in which the antenna and appropriate subsystems (e.g., receiver, exciter, telemetry processor, etc.) are configured in anticipation of the track. During the actual track, the signal from the spacecraft must be acquired and the antenna and subsystems must be commanded to retain the signal as well as adjust for changes in the signal (such as changes in bit rate or modulation index as transmitted by the spacecraft). Finally, at the completion of the track, the station must be returned to an appropriate standby state in preparation for the next track. All of these activities require significant automation and robust execution including closed loop control, retries and contingency handling.

In order to provide this autonomous operations capability, the DS-T station employs tightly coupled state of the art hardware and software. The DS-T software architecture encompasses three major levels: the network level, the complex level and the station level (Figure 1). Within this paper we focus primarily on the station level, but also describe the aspects of the network and complex layer as relevant to the integration of the DS-T into the overall Deep Space Network architecture.

The network layer represents the Deep Space Network wide operations capability necessary to determine the DS-T operations activities over a medium range time scale (a weekly basis) at a high level of activity (the services the DS-T station is to provide to spacecraft over each specific period of time during the week).

The signal processing complex layer represents a layer of control for a group of communications stations at a single physical location. For example, at Goldstone California, USA, there are 6 antennas grouped into a single signal processing complex (SPC). These antennas may need to be coordinated because they may be synchronized to create an antenna array. Also, stations at a single SPC may compete for shared resources (e.g., ground communication channel bandwidth).

Within the DS-T station itself, there are three layers within the software and hardware: the DS-T automation layer, the DS-T application layer, and the DS-T subsystem layer.
First, at the network layer the JPL scheduler layer accepts track requests (along with service definitions) from the flight projects and produces a local schedule for each DS-T station. Second, the DS-T automation layer resides locally at the DS-T site and accepts a local schedule from the scheduler layer. This schedule is interpreted by a schedule executive, that will cause for each track: predict generation, track script generation, and execution of the track script.

* This work was performed by the Jet Propulsion Laboratory, under contract with the National Aeronautics and Space Administration.
This paper appears in the proceeding of the 1998 IEEE Aerospace Conference.
[1] Correspondence Author: Forest Fisher, JPL M/S 525-3660, 4800 Oak Grove Drive, Pasadena, CA, 91109-8099, forest.fisher@jpl.nasa.gov.

The final component of the DS-T automation layer is the Downlink Monitor which runs the scripts that perform the actions for each specific track. The Downlink Monitor is also part of the DS-T application layer where it interfaces to the subsystems.

The DS-T prototype is scheduled to demonstrate automated down-link capability for the Mars Global Surveyor (MGS) spacecraft in March 1998. In this demonstration, a service request for down-link services, a track sequence of events, and spacecraft ephemeris will be used to automatically down-link data from the MGS spacecraft. This demonstration will be enhanced to add up-link capability in the July 1998 time frame. As a further test of the DS-T capability, autonomous down-link and up-link tracking of the New Millennium Deep Space One (NM DS1) Spacecraft is planned (NM DS1 is scheduled for launch in July 1998). Included in NM DS1 support is support of the Beacon Monitor Experiment, in which the spacecraft will initiate a track request by communicating a low bandwidth signal to a small antenna which will automatically trigger scheduling of a demand access track and subsequent automated execution of the track at the DS-T station.

In the remainder of the paper we describe the overall architecture and how it fits into the DSN operations architecture. First we describe each of the layers in the DS-T architecture: the network layer, the antenna complex layer, and the layers comprising an individual station layer (the automation layer, protocol layer, and subsystem layer.) We then describe in further detail the current status of the implementation of the architecture proposed, and finally we make comparisons to other systems.

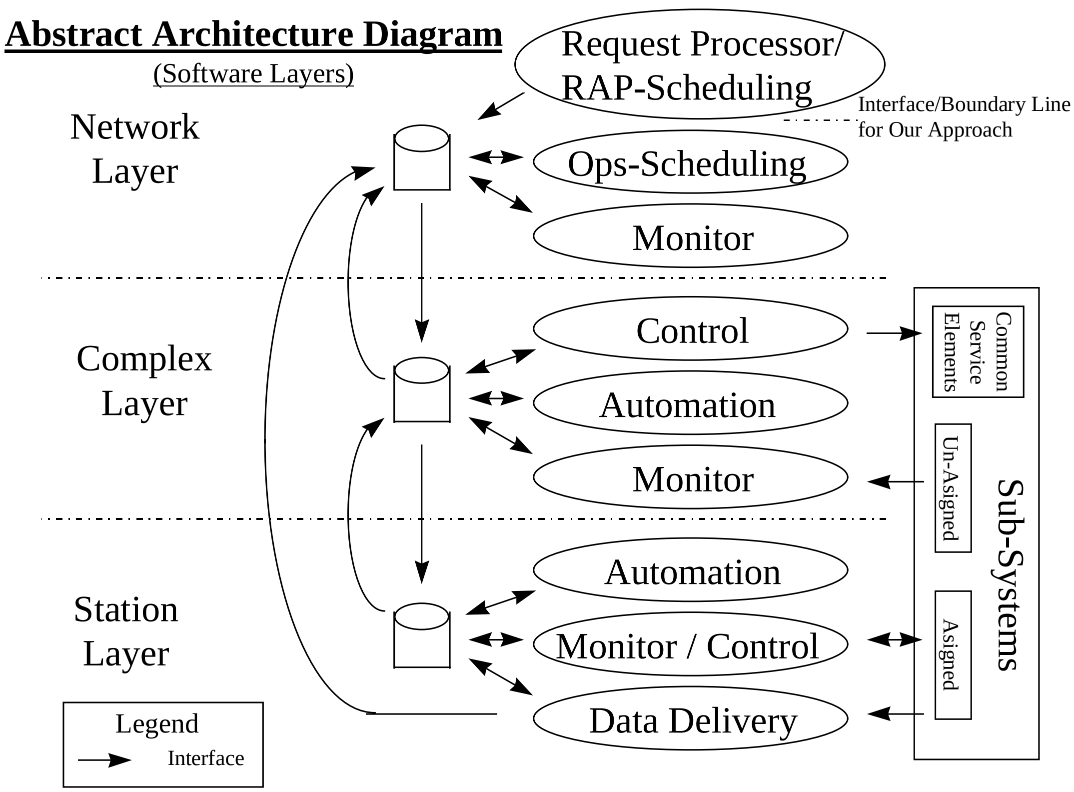

Figure 1: Overall Deep Space Network Automation Architecture

THE NETWORK LAYER

Each day, at sites around the world, NASA's Deep Space Network (DSN) antennas and subsystems are used to perform scores of tracks to support earth orbiting and deep space missions [6, 13]. However, this is merely the culmination of a complex, knowledge-intensive process which actually begins years before a spacecraft's launch. When the decision is made to fly a mission, a forecast is made of the DSN resources that the spacecraft will require. In the Resource Allocation Process (RAP), the types of services, frequency, and duration of the required tracks are determined as well as high-level resource requirements(e.g., antenna). While the exact timing of the tracks is not known, a set of automated forecasting tools are used to estimate network load and to assist in ensuring that adequate network resources will be available. The Operations Research Group has developed a family of systems which use operations research and probabilistic reasoning techniques to allow forecasting and capacity planning for DSN resources [Fox & Borden 1994, Loyola 1993]. These tools are currently being folded into a unified suite called TMOD Integrated Ground Resource Allocation System (TIGRAS) [4].

As the time of the actual tracks approaches, this estimate of resource loading is converted to an actual schedule, which becomes more concrete as time progresses. In this process, specific project service requests and priorities are matched up with available resources in order to meet communications needs for earth-orbiting and deep space spacecraft. This scheduling process involves considerations of thousands of possible tracks, tens of projects, tens of antenna resources and considerations of hundreds of subsystem configurations. In addition to adding the detail of antenna subsystem allocation, the initial schedule undergoes continual modification due to changing project needs, equipment availability, and weather considerations. Responding to changing context and minimizing disruption while rescheduling is a key issue.

The Demand Access Network Scheduler (DANS) [7] is an evolution of the OMP-26M system designed to deal with the more complex subsystem and priority schemes required to schedule the larger 34 and 70 meter antennas. Because of the size and complexity of the rescheduling task, manual scheduling is prohibitively expensive. Automation of these scheduling functions is projected to save millions of dollars per year in DSN operations costs.

DANS uses priority-driven, best-first, constraint-based search and iterative optimization techniques to perform priority-based rescheduling in response to changing network demand. In these techniques, DANS first considers the antenna allocation process, as antennas are the central focus of resource contention. After establishing a range of antenna options, DANS then considers allocation of the 5-13 subsystems per track (out of the tens of shared subsystems at each antenna complex) used by each track. DANS uses constraint-driven, branch and bound, best first search to efficiently consider the large set of possible subsystems schedules.

The network layer has three principle interfaces to lower levels in the automation architecture (as shown in Figure 2). In addition to resource allocation, the network layer is responsible for storing information on the tracking services required by the spacecraft, current spacecraft configuration, planetary and spacecraft ephemeris, and telecommunications models. This information (as well as the current schedule) is stored in a globally accessible database called the Mission and Assets Database (MADB). The MADB is a major interface point from the network layer to the automation element of the station layer.

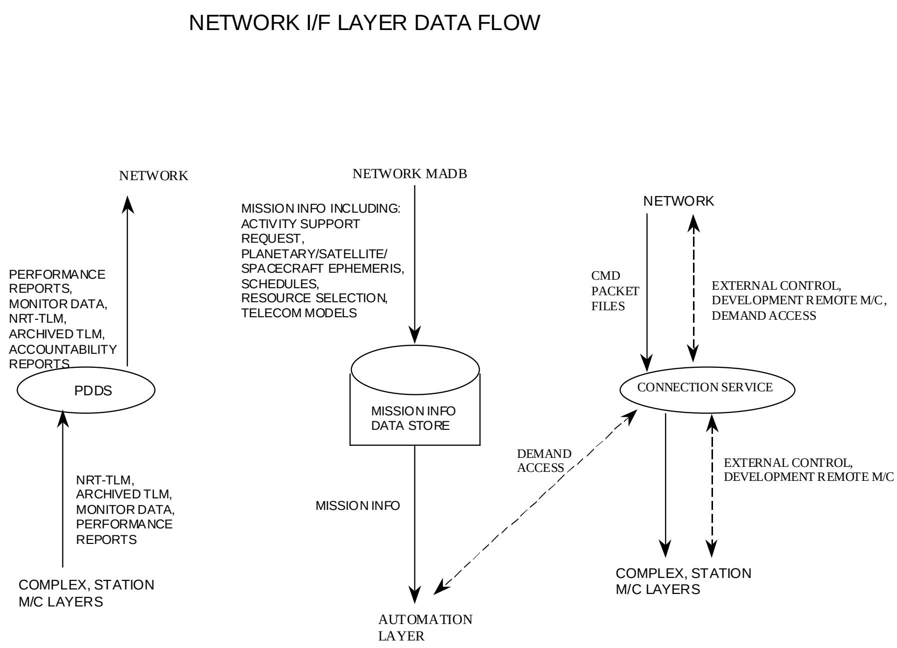

Figure 2: Interface from the Network Layer to the Complex and Station Monitor and Control Layers and the Station Automation Layer

Another required capability of the DSN is to generate near real time telemetry and monitor data as well as performance summarizations. These are generated by the monitor and control layers of the complex and station layers respectively and are forwarded on to the network layer for appropriate distribution.

A third interface point of the network is for delivery of real time commanding to the spacecraft or ground equipment. Some experiments that use the DSN antennas with special purpose equipment require remote control by a project's principal investigator. In order to support this requirement, DS-T allows spacecraft commands to be delivered to the Station just in time for up-link at the desired time

THE COMPLEX LAYER

The complex layer of the architecture provides a local copy of the MADB for the Station controllers, provides reliable data connection to the network layer, and monitors and controls equipment that is either a common resource (e.g. air-conditioning, precise timing, etc.) or not currently assigned to a Station (e.g. downlink equipment, array processor, etc.).

As part of the reliable data connection, the complex layer monitors the telemetry data flow out of the complex so all project commitments are met. Temporary data storage is performed by the Stations but the data accounting and delivery process is done in the complex layer. The monitor data that is generated by the Stations is stored at complex level for later review by an analyst if necessary. At the same time, the monitor data is compressed and summarized before it is sent to the network layer.

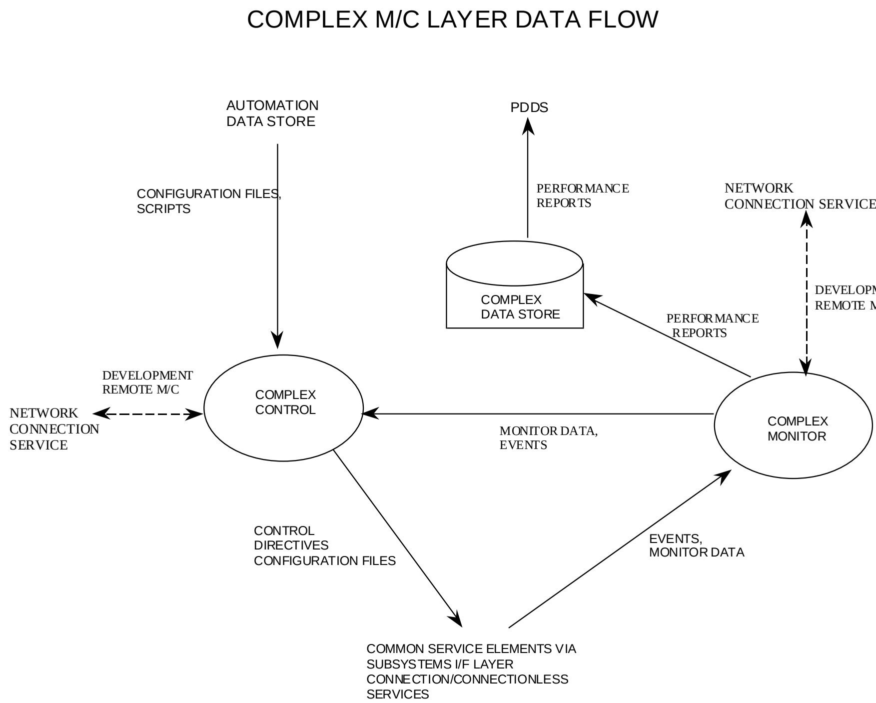

Figure 3: The Complex Layer Architecture

THE STATION LAYER

The station layer repesents the actual hardware and software dedicated to a single DS-T station. There are three principal components to the station layer: the automation layer, the monitor and control layer, and the subsystem layer. The automation layer is responsible for the high level control and execution monitoring of the DS-T station. As such it is capable of configuring the station by requesting the use of assignable subsystems from the complex layer and triggers key pieces of software to generate predicts, generate station operations scripts, as well as be responsible for invoking these processes at the appropriate times. The monitor and control layer is responsible for low level control of the antenna track as well as logging and archiving relevant monitor data. The subsystem level provides a uniform interface to the antenna subsystems to facilitate modular software design and reduce the effort needed to interchange and upgrade hardware.

The Automation Layer

The automation layer performs several functions within the DS-T UNIX workstation; all relating to automation and high level monitor and control for the DS-T station.

The automation layer has five components: the schedule executive, configuration engine, predict generators, script generator, and the station controller.

The schedule executive (SE) sets up the schedule for execution and provides the means for automated rescheduling and/or manual schedule editing in the event of changes to the master schedule. Schedule execution is set up by parsing the schedule and scheduling the sub-tasks which need to be performed in order to accomplish the originally scheduled activity. Each subtask is placed into the crontab file at the appropriate time relative to the Aquisition Of Signal (AOS). In this manner, each of the remaining components of the automation layer are invoked at the appropriate time by the UNIX crontab facility.

The configuration engine (CE) is the first to be started up by the cron facility. This component is responsible for retrieving all the necessary data/data files needed for station operations, from a collection of data stores. These files contain information about: spacecraft trajectory, needed to calculate antenna pointing predicts; spacecraft view periods (when the spacecraft is visible to the antenna); models of planetary orbits, to determine if the spacecraft view is obstructed; precise location of the ground station; and activity service packages (ASP). The ASPs contain the service request which define the type of activity desired by a mission/project and activity details like carrier frequency, symbol rate, and project mission profiles. The CE examines this vast collection of data and extracts the relevant information into configuration files for the remaining modules of the automation layer.

After the CE creates the needed configuration files for the predict generators (PG) and the script generator (SG), the cron facility will invoke each of these processes with their respective configuration files. The PG functionality consists of three predict generators used to calculate: antenna pointing predicts (AP-PDX), radiometric predicts (RAD-PDX), and telemetry predicts (TEL-PDX).

The SG is where the majority of the control autonomy comes from. The SG uses Artificial Intelligence Planning techniques to perform a complex software module reconfiguration process. This process consists of piecing together numerous highly interdependent smaller control scripts in order to produce a single script to control the operations of the DS-T station.

The core engine used in the SG is the Deep Space Network Antenna Operations Planner (DPLAN) [8] developed for generating Temporal Dependency Networks (TDNs). TDNs are a form of control script that are used to perform pre-calibration and post-calibration of DSN antennas. As part of the DST SG, DPLAN uses both hierarchical task network (HTN) and operator-based planning techniques to reason about DST station operations using a model of the station actions. The HTN portion of the planner decomposes hierarchical rules in a forward-chaining fashion, while the operator-based portion of the planner works in a back-chaining fashion from the goal and applies operators whose goals satisfy the preconditions of the previous goal(s). In this fashion the operator applied will have pre-conditions and as such those become the new unachieved goals; this process is referred to as sub-goaling. Through the process of HTN planning and sub-goaling DPLAN generates a plan (in our case a control script) which when executed will satisfy the objectives for the track activities requested within the ASPs.

As previously mentioned, the station controller (SC) spans both The Automation Layer and The Station Monitor and Control Layer. As such the explanation of the SC functionality is left for The Station Monitor and Control Layer section of this paper.

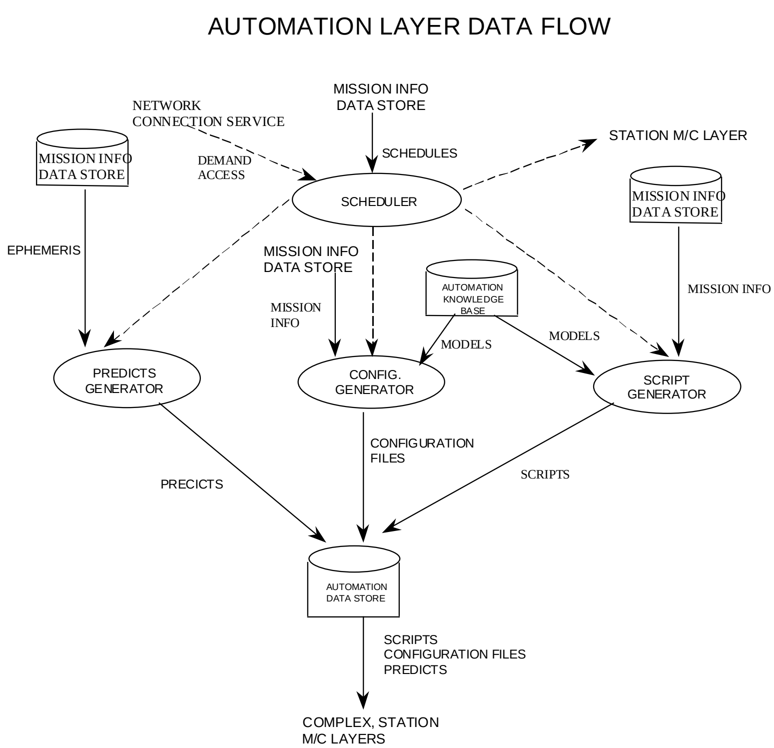

Figure 4: The Station Automation Layer for the Deep Space Terminal

The Station Monitor and Control Layer

The Station Monitor and Control process acts as an agent for the Automation Layer, executing the generated scripts. The Monitor and Control (M&C) layer expands the high level directives of the script into subsystem dependent directives, isolating the automation layer from the lower levels. By using the monitor information from the Station Monitor process, the script execution path is altered as necessary to accommodate external events.

All subsystem generated monitor information (monitor data packets and event notices) is processed in the Station Monitor process. The monitor data is recorded in a data store and condensed performance reports are generated for the higher level processes.

The Up-link/Down-link process handles the spacecraft command and telemetry data flow. The command data is accepted as Command Link Transmission Units (CLTUs) or as command packet files and processed according to Consultative Committee for Space Data Systems (CCSDS) standards. Telemetry data is formatted in the subsystem into frames or packets. These are archived until the data is delivered to the mission or the Product Data Deliver System (PDDS).

For debugging and experimental use the M&C layer has the capability to handle low level directives for the subsystems in bypass mode. .

STATION M/C LAYER DATA FLOW

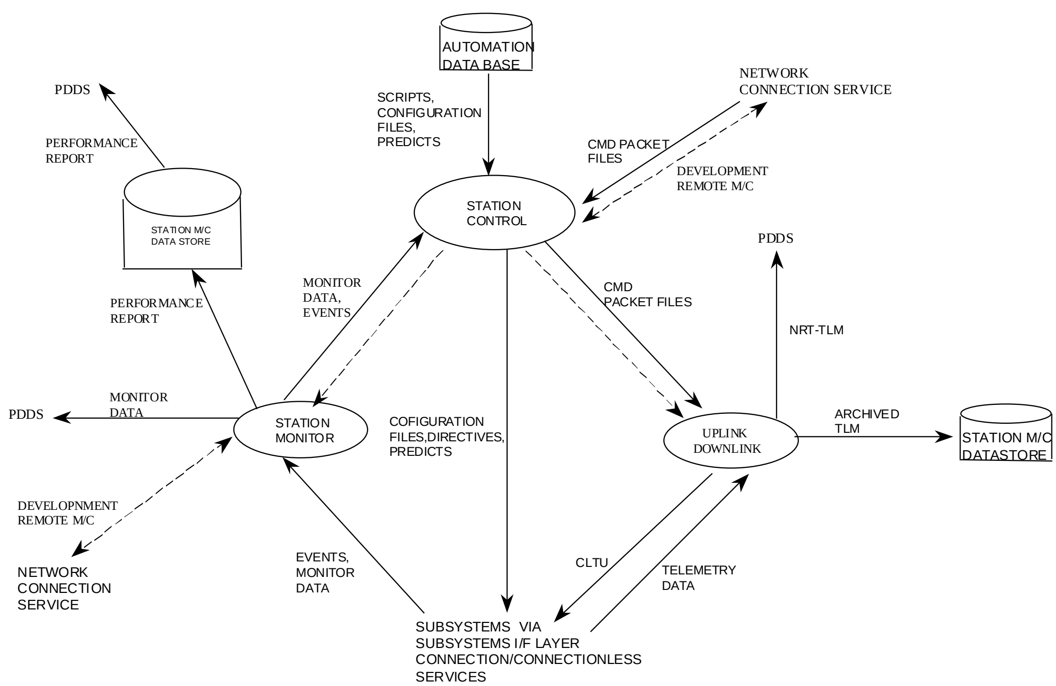

Figure 5: The Station Monitor and Control Layer for the DS-T Station

The Station Subsystem Layer

The Subsystem I/F layer handles all communication protocol and connection related work. This is necessary because the DS-T is a mix of COTS (commercial off the shelf) and custom JPL designed equipment using a variety of protocols. The inherited JPL equipment uses a proprietary communication protocol, while some COTS units use TCP/IP, and others use either the IEEE-488 or RS-232 low level protocols. The JPL protocol also requires the equipment "to be assigned" to a track, requiring some hereditary connection management.

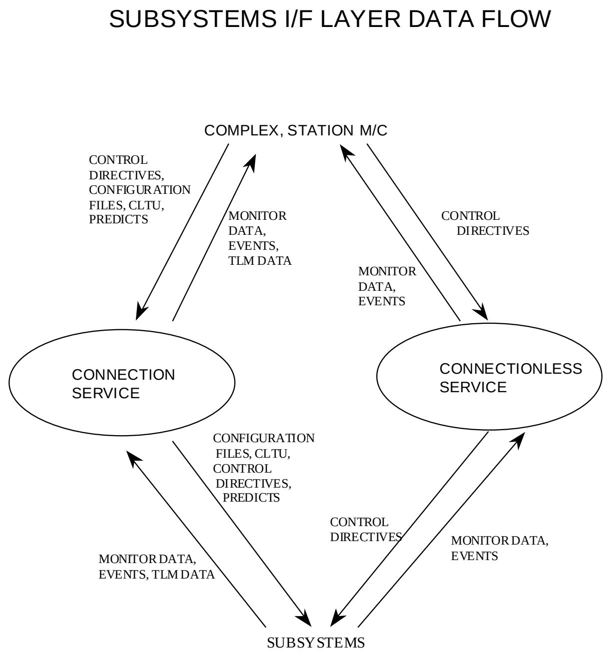

Figure 6: The Station Subsystem Layer for the DS-T Station

Schedule for Prototyping and Demonstration

The DS-T is being developed using an iterative rapid prototype design methodology. As such DS-T is demonstrating its functionality incrementally. In March 1998, DS-T will perform a one week demonstration revealing the station's unattended, lights-out mode of operation during down-link operations with the Mars Global Surveyor (MGS). In the July 1998 time frame, DS-T will demonstrate its up-link capabilities with MGS. In August 1998, DS-T will be used to support the NM DS1 Beacon Monitor mode of operations. In this third demonstration, DS-1 will initiate a track and the DS-T will respond to it. This is a bold new mode of operations in space flight. In this mode, the ground reacts to the spacecraft when the spacecraft decides it needs attention; as compared to current operations, where the spacecraft reacts to the ground when the project schedules interaction between a station and a spacecraft.

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 [5] contains no hierarchy of planning, scheduling, or control. This type of architecture has been used for mobile robot navigation, where re-planning 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 [15] 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 returns 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.

$_{3}$T [3] 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 $_{3}$T is a RAP [10] 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 $_{3}$T 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 [11] is also a three-tiered architecture, similar to $_3$T. 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 [16] 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 [1, 2] 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 [17] 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 [14] and Guardian [12] 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 re-planning 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 then 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 an architecture for an autonomous Deep Space Tracking Station (DS-T). This DS-T station automates routine operations such as: scheduling and resource allocation, antenna and receiver predict generation, track procedure generation from service requests, and closed loop control and error recovery for the station subsystems. This architecture is being validated by the construction of a prototype DS-T station to be demonstrated at NASA's experimental DSN research station, DSS-26. This validation will occur in two phases: down-link (March 98) and up-link/down-link (July 98).

REFERENCES

[1] R. Arkin. Motor schema-based mobile robot navigation. *International Journal of Robotics Research*, 8(4), 1989.

[2] 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.

[3] 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.

[4] C. Borden, Y.-F. Wang, and G. Fox, "Planning and Scheduling User Services for NASA's Deep Space Network," *Working Notes of the 1997 International Workshop on Planning and Scheduling for Space Exploration and Science*, Oxnard, CA, 1997.

[5] R. Brooks. A Robust Layered Control System for a Mobile Robot. *IEEE Journal of Robotics and Automation*, 2(1), 1986.

[6] 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 Direc-*

tions in AI Planning, M. Ghallab and A. Milani, ed., IOS Press, Washington, DC, 1996, pp. 287-298.

[7] S. A. Chien, R. W. Hill, Jr., A. Govindjee, X. Wang, T. Estlin, M. A. Criesel, R. Lam, and K. Fayyad, "A Hierarchical Architecture for Resource Allocation, Plan Execution, and Revision for Operations of a Network of Communications Antennas," *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, Albequerque, NM, 1997.

[8] S. A. Chien, A. Govindjee, T. Estlin, X. Wang, F. Fisher, and R. W. Hill Jr. Automating Generation of Tracking Plans for a Network of Communications Antennas. *International Workshop on Planning and Scheduling for Space Exploration and Science*, 1997.

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

[10] R.J. Firby. Adaptive Execution in Complex Dynamic Worlds. PhD Thesis, Yale University, 1989.

[11] 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.

[12] B.Hayes-Roth. An Architecture for Adaptive Intelligent Systems. *Artificial Intelligence*, 72, 1995.

[13] 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.

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

[15] 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.

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

[17] 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.

BIOGRAPHIES

***Forest Fisher** is a member of technical staff in the Artificial Intelligence Group of the Jet Propulsion Laboratory, California Institute of Technology where he performs research and development of automated planning and scheduling systems for science data analysis and ground station automation. He is also the automation lead for the DS-T task. He holds a B.S. in Computer Science from the University of Texas, and is currently completing an M.S. in Computer Science at the University of Southern California. His research interests are in the areas of: planning and scheduling, operations research, and monitor and control.*

***Steve Chien** is Technical Group Supervisor of the Artificial Intelligence Group of the Jet Propulsion Laboratory, California Institute of Technology where he leads efforts in research and development of automated planning and scheduling systems for science data analysis, ground station automation, and highly autonomous spacecraft . He is also an adjunct assistant professor in the Department of Computer Science at the University of Southern California. He holds a B.S., M.S., and Ph.D. in Computer Science from the University of Illinois. His research interests are in the areas of: planning and scheduling, operations research, and machine learning and he has published numerous articles in these areas. In 1995 he received the Lew Allen Award for Excellence and in 1997 he received a NASA Exceptional Engineering Achievement Medal both for his research and engineering work in automated planning and scheduling systems.*

***Leslie Paal** is a member of technical staff in the Advanced Communications Concepts Group of the Jet Propulsion Laboratory, California Institute of Technology where he does research and development of automated, autonomous Deep Space ground stations. He is the systems engineer for the DS-T task. He holds a B.S, in Electrical Engineering and Computer Science.*

Emily Law *is a senior software engineer in the Telecommunication Signal Processing Engineering Group of the Jet Propulsion Laboratory, California Institute of Technology where she leads efforts in software architecture and development of the Block V Receiver. She is also the software lead for the DS-T task. She holds a B.S. in Math/Computer Science from the California State University and a M.S. in Computer Science from the University of Southern California.*

Nasser Golshan *is a member of the Advanced Communications Concepts Group of the Jet Propulsion Laboratory, California Institute of Technology where he is the task lead for the DS-T task. He holds a Ph.D. in Electrical and Electronic Engineering from the University of Illinois, an M.S. from Seattle University, and a B.A. from The American University of Beirut.*

Mike Stockett *is a member of technical staff at the Jet Propulsion Laboratory, California Institute of Technology.*