Automating Operations for NASA's Deep Space Network (DSN)

Eva Bokor
Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena, CA 91109
(818)393-5448
Eva.Bokor@jpl.nasa.gov

Abstract - The DSN is the communication gateway to NASA's Earth Orbiting and planetary spacecraft. The DSN is unique among space communication systems in that it is link configurable. For each pass of a spacecraft, there is a set of antennas, receivers, transmitters, and other signal processing subsystems which can be combined to complete a tracking cycle. While in the past the number of orbiting satellites and planetary spacecraft were fairly small, manual control of this complex tracking system was manageable. However, with the explosion in planned missions for Mars, Venus, the sun, outer planets, comet flybys, and asteroid encounters, it is becoming increasingly clear that the complex tasks associated with antenna control/coordination and telemetry capture will eventually overwhelm existing tracking crews. Mistakes in data entry/retrieval can result in loss of valuable science data as well as spacecraft control blackout. This paper describes the currrent DSN automation system design, functions, test results, and actual field implementation experience.

TABLE OF CONTENTS

0-7803-6599-2/01/$10.00 © 2001 IEEE

1.INTRODUCTION

The NASA Deep Space Network (DSN) is an international network of spacecraft tracking and communication complexes located at Goldstone, California, Madrid, Spain, and Canberra, Australia. The complexes are placed approximately 120 degrees apart in longitude. This strategic placement permits constant observation of spacecraft as the Earth rotates. Each complex consists of several DSN stations equipped with large parabolic reflector antennas and ultra sensitive receiving systems that include 70-meter, 34-meter, 26-meter and 11-meter in diameter antennas. The network supports two-way communication with interplanetary space missions and some Earth-orbiting missions. The DSN is also used to support the Space Shuttle in emergency. The Jet Propulsion Laboratory (JPL) manages and operates the network for NASA from a central operations center in Pasadena, California. See Figures 1 and 2.

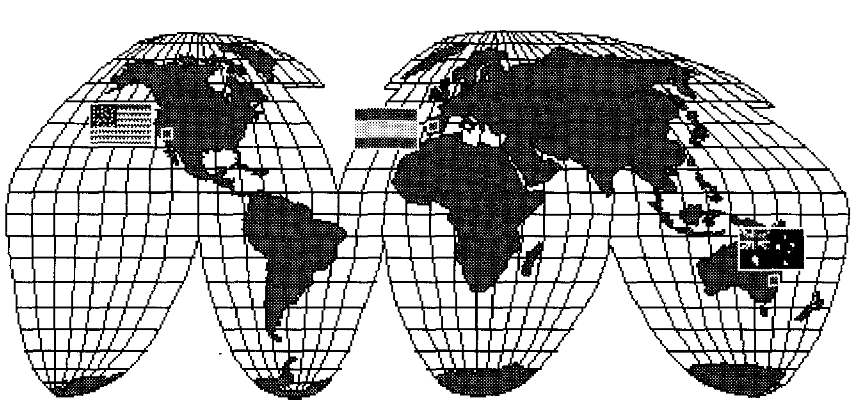

Figure 1: DSN communication complexes around the World

The DSN is unique among space communication systems.

The DSN is primarily the communication link for spacecraft located far out in the solar system. The reduction in strength of the signal by the time it has traveled complicates the mission support immensely. Furthermore Deep Space Mission data retrieval, spacecraft rendezvous, scheduling, and data reduction can vary depending on their trajectories. For example the Mars Global Surveyor is currently orbiting around Mars daily, while Voyager is heading into Deep Space at a steady pace and still sending data.

The communication is link configurable. The antennas of the DSN are operated from three Signal Processing Centers (SPCs), utilizing a combination of assigned signal processing equipment and a pool of shared data processing equipment. The goal of this centralization effort was to be able to switch devices to any antenna as needed. For each pass of a spacecraft, a unique collection of antennas, receivers, transmitters, and other signal processing devices can be assigned to support tracking. It is this capability to "mix and match" that creates the need for a high level of human effort by the station operators, and also adds a large degree of complexity when trying to manage several spacecraft encounters within a given time window of opportunity.

2.CONTROLLING THE DSN

The basic monitor and control activities at the DSN need to be explained in order to understand the complex task of implementing DSN Automation.

All of the signal processing devices in the DSN are computer controlled. Each computerized element is also called subsystem, which includes their respective software components as well. Human operators, working at computer consoles at the signal processing centers, issue directives to these subsystems to perform a specific activity. A directive is analogous to a command in UNIX or DOS. They receive confirmations from the subsystems on the directives' execution. The confirmation may be a direct response by the receiving subsystem, a text message, a change in a monitor data field displayed on the console, or an event message generated by the subsystem. This combination of directive-response is known as positive closed loop control.

Shortly before a monitor and control workstation is scheduled to perform a track, all necessary data is downloaded to the workstation. This includes the identity of the spacecraft to be tracked, the designated antenna, and information known as the sequence of events (SOE). The SOE contains the what, when, and how of each tracking activity. It specifies the equipment to be used, the time and point of spacecraft rise and set, and parameters needed for equipment calibration. In addition, it specifies the version of the device controller software to be used.

The monitor and control of DSN can be described at an abstract level using a systems approach. The activities are divided into pre-track, in-track, and post-track phases. Pre-track typically the most labor intensive, includes allocation, configuration, and calibration of equipment and movement of the antenna to point at the spacecraft's position. When a radio link with the spacecraft is acquired, track phase begins. While pre-track activities may take 30-60 minutes, track phase may last for hours, and even tens of hours. If all goes well, this phase consists of merely monitoring data flows. This is the critical phase when spacecraft state data and science data are retrieved. If this phase is not executed properly or during the correct time window, critical spacecraft safety data may be lost placing the spacecraft at risk. Similarly, important science data could be lost, never to be retrieved. The post-track phase begins when the spacecraft goes down over the horizon and communication is lost. The operator then returns the equipment to the available pool and returns the antenna to its stowed position.

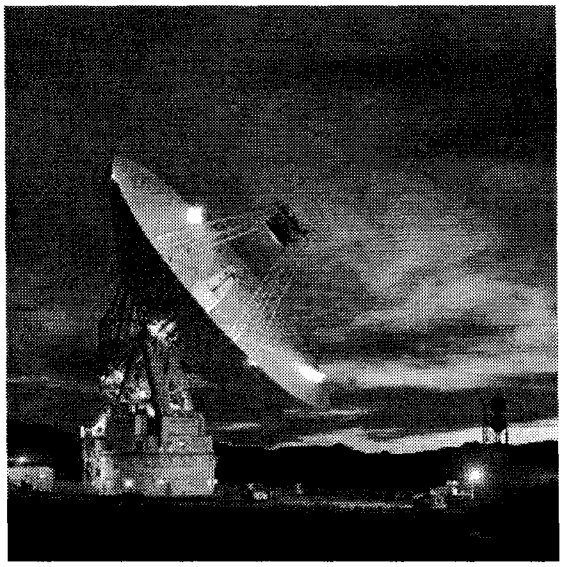

Figure 2: The 70 meter antenna at the Goldstone Tracking Station

3. BACKGROUND BEHIND THE DECISION TO MOVE TOWARDS AUTOMTATION

Because of the criticality of retrieving spacecraft data during periods of overlapping encounters, DSN engineers began to examine automation of routine, repetitive activities in order to allow operators to focus on crucial mission functions and reduce chances of operator errors. Exploring new means for automating activities of the DSN began in the late 1980s. The initial effort was based on software macros, which are collected sets of directives or commands, and they are executed as a script. These macros are currently being used at the DSN complexes in order to group activities. Eventually the macro language was found limiting and had little visibility into subsystem status [1]. Still, the macros at each station reflect operational procedures that were captured in software form and demonstrated that DSN activities could be organized. A new goal for the automation efforts was to consolidate these procedures into an easily updateable and configurable form that is robust and flexible enough to be used at any station. The existing common equipment pool architectural concept at the DSN made the automation attempts very difficult.

A development effort was started by the Network Monitor and Control (NMC) Project at the NASA Jet Propulsion Laboratory in the early 1990's. To improve efficiency, the NMC Project has developed the Automation Software package (referenced as Automation Assembly), which assists the DSN operator during pre-track, in-track and post-track activities. These activities perform functions which use repeatable knowledge patterns (planned activities of operations) and leave the operator free to concentrate on exception cases and subsystem anomalies. NMC Automation's goal is to provide the ability for simultaneous subsystem configuration, utilizing closed loop control, while reducing required manual input and subsequent potential for critical operator errors.

4.THE TASK OF AUTOMATING THE DSN

The goal to automate the DSN is overwhelming.

Each of the three Deep Space Communication Complexes consists of at least four antennas - of 4 types. The DSN Antenna Project will soon provide five additional antennas at the Goldstone tracking station to support the growing demands for more simultaneous encounters and more data.

A common pool of more then 11 data processing equipment is being used that could be switched to any antenna as needed. This common pool includes the telemetry processing, command processing, and the monitor and control devices. This approach provides the operator with common interfaces to all subsystems, and the capability of quickly replace failed devices during support. Unfortunately the centralized architecture did hurt the DSN automation efforts [2]. Much of the DSN's equipment is a diverse collection of aging, custom-built devices. Some of the computers are over 15 years old, while some mechanical devices are 30 years old. Elimination of equipment would mean loss of valuable science data as they are still used to communicate with older spacecraft. Many of these subsystems were not designed to support automation. The subsystems' behavior changes between instances and versions of subsystems' common software. Multiple versions of subsystems (both hardware and software) may be in use at any time.

Further complications were seen with the desire to support of all possible missions. Each mission is different, and additional support activities are required during a mission life cycle. The SOE describes the detailed events that are to occur during a spacecraft's pass. The SOE is available at the workstation in electronic format, but currently it is not consistent between missions.

The procedures for monitor and control are site dependent. Each communication complex has unique configurations of equipment, though many elements may be common across the sites. Each site has unique procedures for operations, although they all are supposedly designed to meet the same objectives.

The DSN is a rapidly changing environment. New antennas, new devices and new subsystems are introduced, new missions are supported and old ones are removed on a daily basis. As the reader can see, the task of automating the DSN, although well founded, is daunting due to the size of a very complex system.

To overcome these extreme difficulties the NMC Automation had to implement a global control strategy, that is manipulated by certain well defined operations.

The NMC Automation first developed an Automation Knowledge Base, which is a representation of the human operator's knowledge and actions. This Knowledge Base was then translated into a language which was understandable by computers. The following sections will describe the development of the knowledge engineering and representation, software implementation, and operational use of this approach in DSN automation.

5.KNOWLEDGE ENGINEERING AND KNOWLEDGE BASE

Knowledge Engineering is the process of capturing human expertise and transferring it into a form of Knowledge Base that is usable by a computer program. On the NMC Project this consisted of interviews with operations personnel and with the engineers that built the various devices of the DSN. NMC Automation developers invested tremendous efforts and time to understand the complex ways devices and links are operated at the different communication complexes for different tracking supports.

Several challenges were encountered during the knowledge engineering. The first was identification of the appropriate knowledge sources. No unified data source listed the names of the engineers who built the devices. In some cases, the person had left the Laboratory and maintenance had been handed off to a person who lacked in-depth understanding. After the experts were identified, extracting the knowledge provided additional challenges. Often the engineers responsible for the devices, and the operators who used them daily, disagreed on the proper operational procedures. Differences also existed between individual operator's views of the devices.

In some instances issues could only be resolved by "touching the elephant" so to speak [5]. Because the nearest tracking station was a three hour drive from JPL, much of it on two-lane desert roads, we consumed an immense amount of time with making numerous trips to the Goldstone tracking station to obtain needed information from operators and engineers. However, eventually we were successful at capturing a fairly complete set of engineering and operator data which represented the way the DSN system was operated on a day-to-day basis.

The acquired Knowledge Base in the Automation Assembly comprises of all operations logic and parameters to perform a full or partial support for a supported spacecraft. It is a flexible, user modifiable database. It should be easily updated when a new subsystem type or a subsystem version is delivered to the DSN, when any procedural change is introduced in operations, when a new spacecraft needs to be supported, or a new antenna is built.

6.KNOWLEDGE REPRESENTATION

The Knowledge Representation is the process to convert the acquired Knowledge Base into a representation language, organized, managed and stored by the NMC Automation Assembly.

To fractionate monitor and control activities, the end-to-end flow of operations for each spacecraft support is divided into three activity categories: pre-track, in-track and post-track activities. These activities are further categorized by specific support:

Tracking: Measures the angular position and velocity of the spacecraft.

Tracking and Telemetry: Adds one-way down-link communication from the spacecraft .

Tracking, Telemetry, and Command: Adds up-link communication with the spacecraft.

Tracking, Telemetry, Command, and Ranging: Adds measurement of the distance of the spacecraft from earth.

For any of these four scenarios, the actual equipment used (again, both hardware and software), may vary greatly depending on the spacecraft and antenna involved. Our Knowledge Representation takes considerable pain to hide this fact from the operator. The NMC Automation Assembly presents generic activities using generic equipment.

These generic activities are represented, both conceptually and graphically, by Temporal Dependency Networks (TDNs). A TDN is a directed acyclic graph containing the temporal and behavioral knowledge required to perform a specific task. A sample TDN is illustrated in Figure 3. Each arc within the graph represents a precedence relationship or a dependency, as one block is dependent on another. Each node represents a discrete task or event that must be performed or deliberately omitted before the graph can be further traversed. These nodes are called TDN Blocks.

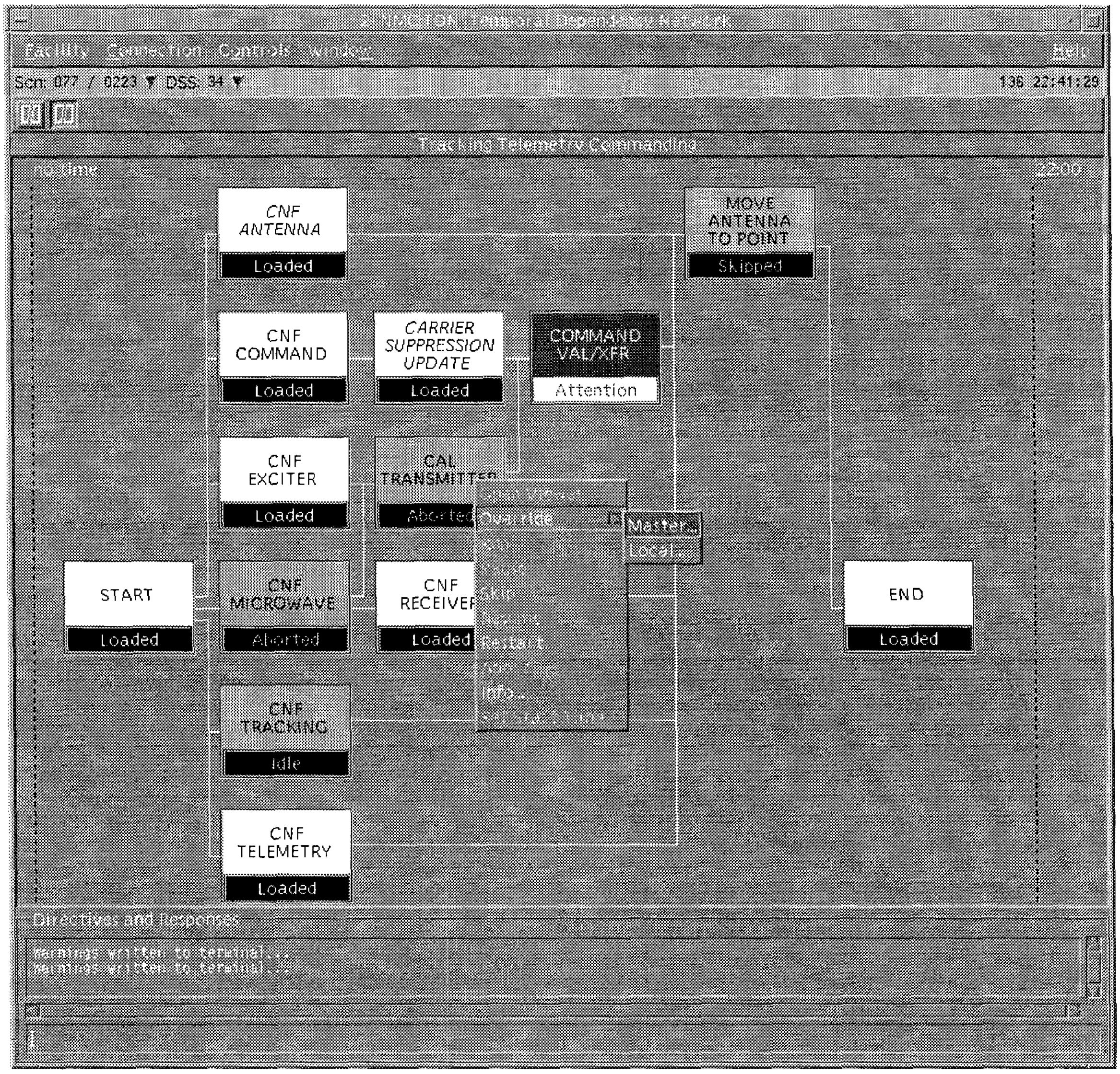

Figure 3: Pre Track TDN Display

A TDN Block is a self-contained and independent object within a TDN, which describes a logical set of activities for operating one or more subsystems (e.g. configure telemetry or move antenna to point). These blocks are the fundamental elements of control of the DSN. The TDN Blocks perform various tasks. They send directives to ground equipment, achieve closed-loop control via monitor data and event notifications, they allow the DSN operator to issue safety pages (alert for antenna movements), they check equipment status. The block comprises of pre-conditions, which are entry criteria that must be satisfied before the block can be executed, and post-conditions, which are exit criteria for the blocks.

As mentioned earlier, each TDN Block controls one or more DSN subsystems. The subsystems are computer controlled. This permits use of different software versions to achieve different technical objectives within the same hardware. Changing the version of software invoked provides the option of using any one of several versions (e.g. a transmitter for a given tracking activity). The flexibility inherent in this technique creates an additional challenge for the Automation Assembly. The TDN

Blocks are written at an abstract level for generic devices and do not have awareness of the multiple versions that exist in practice. Therefore, the Automation Assembly requires the intelligence to select the proper version of the signal processing device. The Automation Assembly uses an internal look-up table to select the correct version of the device controller's software.

The Automation Language for Managing Operations (ALMO) language was designed for developing TDN Blocks in the NMC environment [3]. ALMO is a superset of Tcl/Tk, the Tool Command Language [4]. The ALMO extensions permit the scripts to be written in a syntax that is very close to the macro language used in the previous monitor and control system. This makes it easier for tracking station operators to do script writing and maintenance.

7. SOFTWARE IMPLEMENTATION

The software collectively known as the NMC Automation Assembly consists of five processes. These are the TDN Engine, the Automation Engine, the SOE Parser, the Block Engine, and the Block Manager. The software architecture is illustrated in Figures 4.

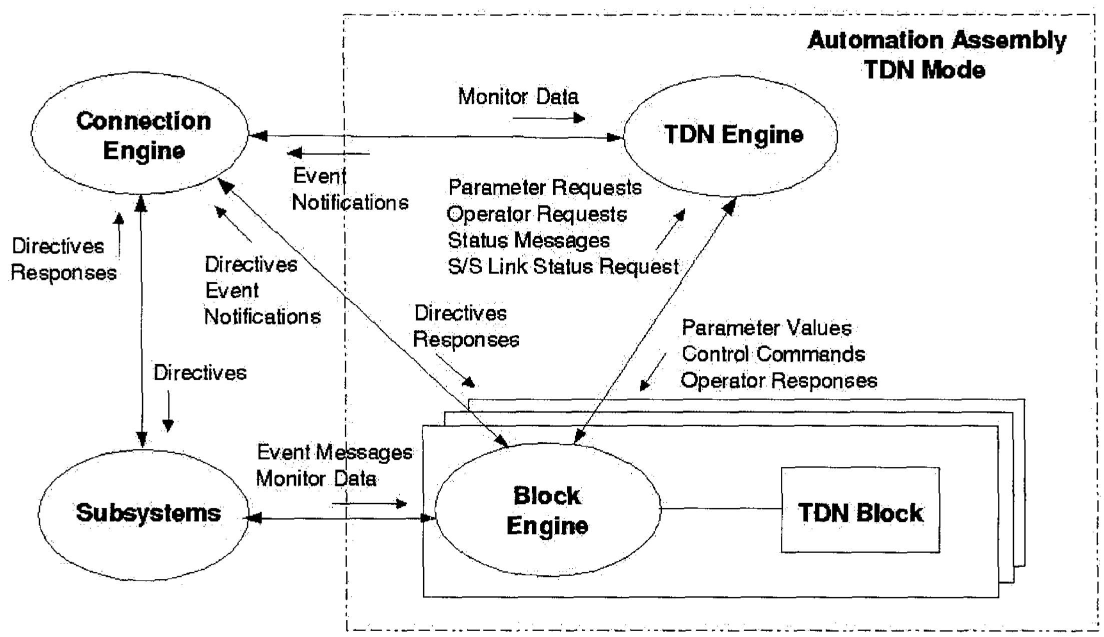

Figure 4. Automation Architecture

The **TDN Engine** controls the execution of pre-track, in-track and post-track activities. It executes an operational scenario which can be predefined or generated from an integrated parser and TDN generator. In the current NMC Automation the pre-track and post-track activities are predefined and user selectable. The in-track TDN is dynamically generated from the SOE. See Figure 5. The TDN Engine defines and graphically represents dependencies between TDN Blocks and the order in which TDN Blocks may be executed to accomplish an activity.

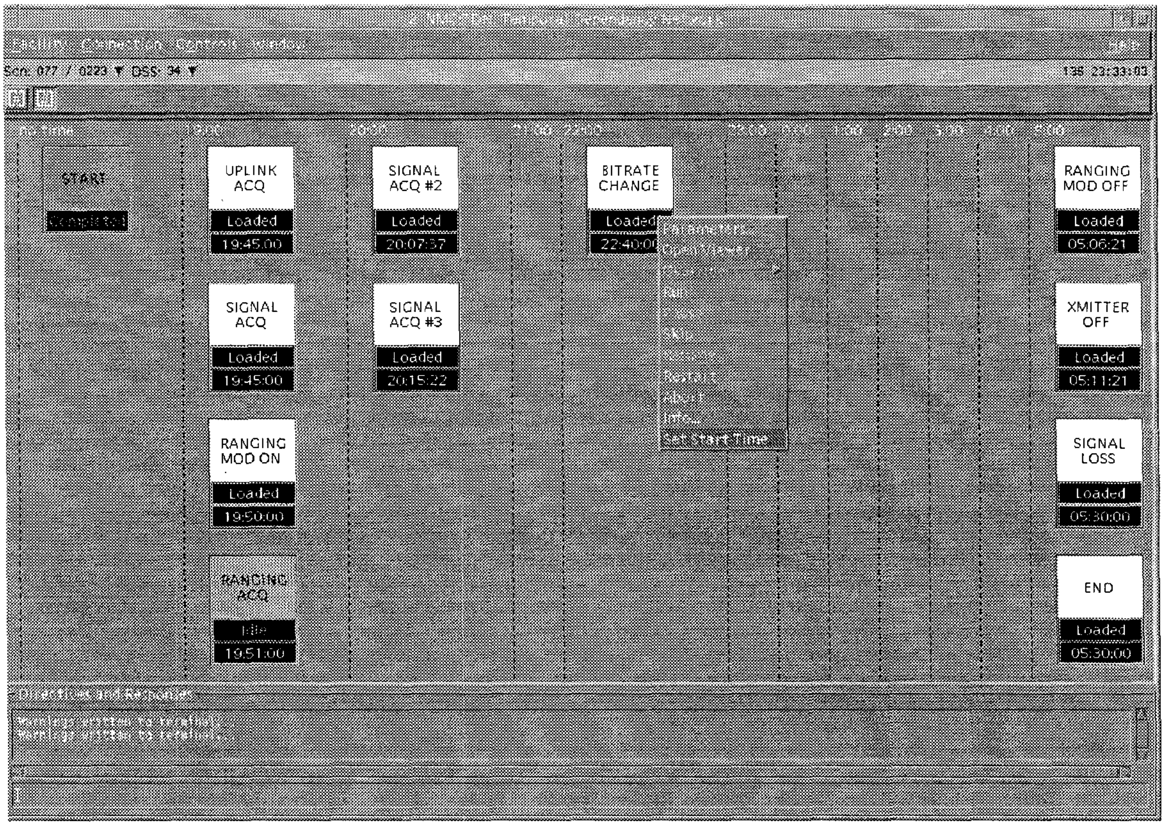

Figure 5: In Track TDN Display

Upon invocation, the TDN Engine performs:

a-basic initialization steps
b-invokes the SOE Parser process
c-obtains information on subsystems that are configured to support the link
d-loads the Automation Knowledge Base elements which includes the Automation Tables, the TDN Block Database, and the TDN Blocks.

When all the engine initialization criteria are met, the TDN Display appears, and the process begins to load the TDN Blocks. The engine uses a library call to the Block Manager. The Block Manager responds with the files containing the appropriate TDN Block. The Engine creates an instance of the Block Engine for each block. The TDN engine sends control messages (e.g. Run, Pause, Resume, Info) to the Block Engine and receives status information (e.g. Loaded, Aborted, Failed, Completed) from it. The TDN Display enables the operator to communicate with the executing TDN Blocks.

The operator has to initiate the execution of the TDN by clicking on the Start button. Either the hierarchy of blocks, time stamps or both control a TDN's activity flow. A block will only run when certain conditions have been satisfied. If a block has a dependency, then the block cannot run until all its parents have completed. If the block has a timestamp, the block will not run until the designated time rolls around. At the beginning of execution the operator is presented with information from various sources, which is necessary to run a specific block. This information is parsed from the SOE, extracted from Automation Tables, and from various electronic sources. The operator is given the choice to accept or override the presented data. This step is necessary due to unreliable data sources at the DSN.

If a problem occurs during block execution, the operator is prompted to intervene. He or she may correct the problem and restart the block, choose to continue in spite of the problem, or abort the block. When the terminal block End is reached, the activity is completed. The next TDN scenario can be started automatically by activating a linking option, or manual control can be resumed.

The **Automation Engine** is a subset of the TDN Engine. It is mainly used to run user defined and constructed Blocks outside of a TDN activity structure. The blocks are written in ALMO.

The **SOE Parser** is one of the most powerful features included in the Automation Assembly. It is a text parser that can parse specified keywords from the SOE file. From the parsed keywords, values can be extracted and then linked to TDN Blocks to start an activity or to store a parameter value in an accessible pool for future TDN Blocks. With this capability, it is possible to automatically produce a TDN scenario from the SOE. The SOE Parser is a Tcl-based parser engineered to the format of an electronically distributed SOE. The SOE Parser is completely user defineable and modifiable.

The **Block Engine**, invoked by the TDN Engine, executes a single TDN Block. It interprets ALMO commands, and interacts with the various subsystems. The process ends when the TDN Block completes, aborts or fails on an error. There is one-to-one correspondence between a Block Engine and a TDN Block. The Block Engine has its own user interface display which allows the operator to monitor and control individual block execution.

The **Block Manager** is a tool used for the configuration management of TDN Blocks for the Automation Assembly. It provides a graphical user interface to update the TDN Block Database such that the TDN Engine loads the correct TDN Block in any scenario. This tool further provides TDN Block editing and ALMO syntax checking capabilities.

8. OPERATIONAL USAGE

The NMC Automation Assembly is currently being used in real time mission supports at the Canberra Deep Space Communication Complex. This excludes the in-track automation support, since this part is being prepared for delivery. Further deliveries are planned to the Goldstone and Madrid complexes in the very near future.

The current NMC Automation supports 7 antennas and over 10 spacecraft for pre-track and post-track activities in real time.

9. CONCLUSION

Although the DSN provided unique challenges, reports indicate that the effort is successful. The pre-track and post-track Automation support is extensively used by our customers, the in-track support is now in the final stages of testing. Results indicate that time to conduct pre-track activities is reduced by 50% or more. The operators are able to manage multiple spacecraft at the same time, something that is not possible without automation. Nevertheless, automation is not a substitute for an experienced DSN operator. When a serious anomaly occurs, the expertise of the human is needed to correct the problem or devise a work around.

10. REFERENCES

[1] Cooper, Lynn P., "Operations Automation Using Temporal Dependency Networks", Proceedings of the Technology 2001 Conference, San Jose, CA, December 3-5, 1991.

[2] Osman, Jeffrey W., Barker, James T., Rohrer, Richard A., "Automating The Operations Of NASA's Deep Space Network 26-Meter Antennas", IEEE 2000.

[3] Santos, Patricia, Pechkam, Paul, "An Automation Language for Managing Operations in the Deep Space Network (ALMO)", NASA Tech Briefs, Engineering Solutions for Design & Manufacturing, Pasadena, CA, Sept. 1999.

[4] Osterhut, John K., Tcl and Tk Toolkit, Addition-Wesley, Reading, MA, 1994.

[5] Bokor, E., Santos, P., Pechkam, P., Thornton, M., Olguin, P., Camilli, B., Gomez, M., " Engineering Automated Operatios for NASA's Deep Space Network", NASA Tech Briefs, the Design/Engineering Technology Digest, 1998.

11. BIOGRAPHY

Eva Bokor is a senior Cognizant Development Engineer on the Network Monitor and Control project at JPL. Part of her responsibilities is to lead a team of developers to implement the Automation Assembly software for the Deep Space Network Monitor and Control Project. She received her undergraduate degree in Physics from the University of Kossuth Lajos, Debrecen, Hungary, and her graduate degree in Mathematics from the University of Kossuth Lajos, Debrecen, Hungary, in 1978. She has worked as a software engineer at the Jet Propulsion Laboratory for 13 years.

12. ACKNOWLEDGMENTS

The author would like to thank Sharon Anthony, Tim Gregor, and John Johnsson of the Goldstone Deep Space Communication Complex (DSCC), Ossi Larikka, Michael Smith, Phil Pope, and Scott McDonald of the Canberra DSCC, and Francisco Jimmenez of the Madrid DSCC for their significant contributions to this effort. She also thanks Hugh Henry, Supervisor of Network Monitor and Control Group, Jet Propulsion Laboratory, for his inputs in writing of this paper. The author wishes to acknowledge the excellent work of Paul Pechkam, Jet Propulsion Laboratory, that he contributed to the product described in this paper. Thank is expressed to former Automation Team members: Patricia Santos, Marla Thornton, Patrick Olguin, Bryan Camilli, and Manual Gomez.

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