N90-25532

Artificial Intelligence for Multi-Mission Planetary Operations

David J. Atkinson Denise L. Lawson Mark L. James

Jet Propulsion Laboratory/California Institute of Technology
M.S. 301-440
4800 Oak Grove Drive
Pasadena, California 91109
USA

ABSTRACT

This paper gives a brief introduction to an automated system called the "Spacecraft Health Automated Reasoning Prototype" (SHARP). SHARP is designed to demonstrate automated health and status analysis for multi-mission spacecraft and ground data systems operations. The SHARP system combines conventional computer science methodologies with artificial intelligence techniques to produce an effective method for detecting and analyzing potential spacecraft and ground systems problems. The system performs real-time analysis of spacecraft and other related telemetry, and is also capable of examining data in historical context. Telecommunications link analysis of the Voyager II spacecraft is the initial focus for evaluation of the prototype in a real-time operations setting during the Voyager spacecraft encounter with Neptune in August, 1989. This paper will report on the preliminary results of the SHARP project and discuss plans for future application of the technology.

INTRODUCTION

The Voyager 1 and Voyager 2 spacecraft were launched from Cape Canaveral, Florida, on August 20, 1977. The technology to track and monitor such probes was designed and developed in the early 1970's. During critical periods of the mission, up to 40 real-time operators are required to monitor the spacecraft's 10 subsystems on a 24-hour, 7-day-per-week schedule. This does not include the numerous subsystem and scientific instrument specialists who must constantly be available on call to handle emergencies. To accomodate the increasing load on mission operations in the 1990's when the Voyager, Magellan, Galileo, Ulysses, Mars Observer, CRAF, and Cassini spacecraft are all flying, JPL has initiated an effort to coordinate all missions through one multi-mission team in the Space Flight Operations Center (SFOC) that operates all spacecraft.

The Spacecraft Health Automated Reasoning Prototype (SHARP) is an on-going effort to apply artificial intelligence (AI) techniques to the task of multi-mission monitoring and diagnosis of spacecraft and ground systems, and to demonstrate those capabilities in tough, operational settings to prove their performance. The Voyager 2 spacecraft and its Telecommunications subsystem were targeted for the initial SHARP demonstration. The spacecraft's August 1989 encounter with the planet Neptune afforded an excellent opportunity to evaluate SHARP in an intense operational environment. The Telecommunications subsystem suffers from frequent anomalies and also requires coordination of monitoring and diagnosis efforts of both the spacecraft and ground data systems (GDS). Finally, due to cumbersome and time-consuming manual processes and obsolete technology, severe limitations exist on the current method of analyzing Voyager telecommunications data. Quite simply, this was both an operations area sorely in need of automation as well as one of the most challenging to automate.

DESCRIPTION OF SHARP

SHARP introduces automation technologies to the spacecraft monitoring process to eliminate much of the tedious analysis associated with analyzing and responding to spacecraft and ground system anomalies. SHARP also automates many of the more mundane, manual processing activities in mission operations. The SHARP system features on-line data acquisition of all required information for monitoring the spacecraft and diagnosing anomalies. The data is centralized into one workstation, which serves as a single access point for the aforementioned data as well as for the diagnostic heuristics. Figure 1 illustrates a

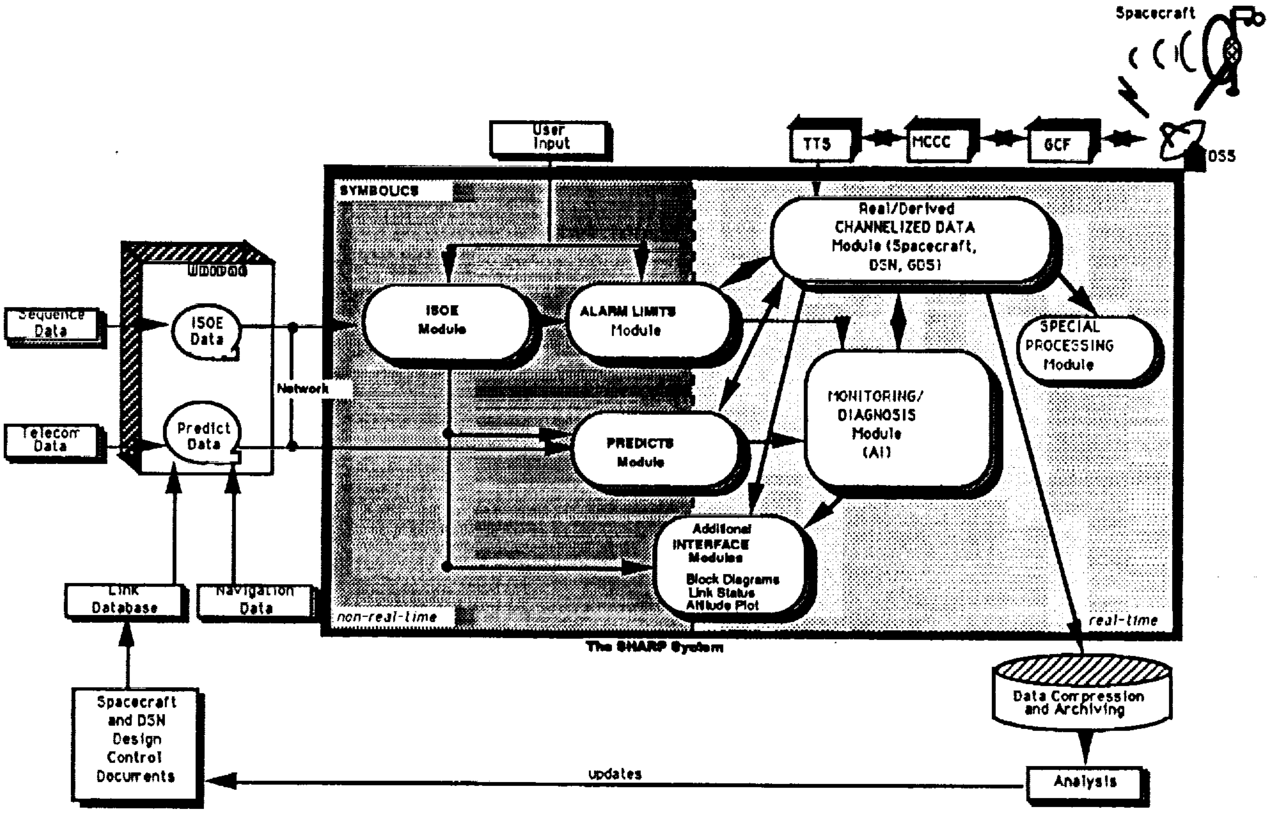

Figure 1: SHARP Telecom System Overview

top-level view of the SHARP system. Shown are the individual modules that comprise the system, as well as relevant components that are external to the Voyager application of SHARP. This section describes these components and related SHARP features in more detail.

The SHARP system provides numerous sophisticated graphical displays for spacecraft and station monitoring. A comprehensive user interface has been developed to facilitate rapid, easy access to all pertinent data and analysis. An interface exists for each major module of the SHARP system. Each interface provides customized functions that allow data specific to that module to be easily accessed, viewed, and manipulated. Each SHARP module can be accessed from any other module at any time, and all displays are in color with mouse sensitivity and menu-driven commands.

SHARP is implemented in Common LISP on a SYMBOLICS 3650 color LISP machine. Many components of the system utilize STAR*TOOL [1] (patent pending), a language and environment developed at JPL which provides a toolbox of state-of-the-art techniques commonly required for building AI systems. The SHARP system is a moderately large system, consisting of approximately 354,000 lines of Common LISP source code, including code specifically developed for SHARP, knowledge bases, STAR*TOOL functions, and Symbolics Genera^{TM} supporting code.

Predicts

To later evaluate spacecraft performance, predictions, or "predicts," of a set of critical parameters are usually obtained in advance for particular spacecraft subsystems. Numerical simulations are frequently employed in the generation of predicts. The SHARP system captures raw telecommunications predicts which are output from an existing computer program. Pass predicts are automatically generated at 15-second intervals, the shortest possible time interval between the arrival of any two spacecraft data points. Instantaneous predicts, which are pass predicts corrected in real-time for spacecraft pointing loss and Deep Space Station (DSS) system noise temperature, are also automatically calculated at 15-second intervals. Spacecraft and DSS residuals, difference measurements between the actual values and predicted values, are automatically derived in real-time. These functions alone can save up to two hours of operator time each shift.

The Predicts interface in SHARP allows tabular display of raw predicts, pass predicts, instantaneous predicts, and residuals for any specified time range. A color-coded DSS availability graph has also been provided which enables rapid identification of available stations for any given viewing period. Situations that mandate that another Deep Space Station be acquired can be addressed immediately as opposed to the more arduous current method, which requires the manual look-up of each station at the specified time period.

Integrated Sequence of Events

The SHARP system also acquires the Integrated Sequence of Events (ISOE), from an external computer. The system provides an ISOE interface which offers numerous capabilities to the operator. A generic capability to extract subsystem-specific information from the ISOE has been developed, e.g., Telecom-specific events may be stripped from the ISOE and displayed to enable rapid identification of significant Telecom activities to be monitored during any particular pass. Editing of the ISOE, e.g., to reflect real-time commanding of the spacecraft, may be performed with ease. This reduces the likelihood of referencing outdated material. Editing is accomplished via menu-driven commands that contain explanations of the complex ISOE data. For example, CC3A32330 means that the x-band modulation index is 32, the two drivers are on, the subcarrier frequency is high, and the data line rate is high. Translation of these spacecraft commands from their raw form into more understandable summaries of spacecraft activity may be performed, and the user can request status summaries of any activity. A history display is maintained as the ISOE is updated so that the user can verify modifications.

Channelized Data Plotting

SHARP includes very flexible plotting capabilities for channelized data. Using simple menu-driven commands and program parameters, an operator can construct an arbitrary number of data plots on an arbitary number of screens, a significant improvement over existing capabilities. The user dynamically customizes the display at any time by selecting which and how many channels to view, the time scale, the data range for each plot, and even the icon to use for graphing points on each channel. Each plot is color-coded by the user for easy visual distinction between displayed channels. When any channel is in alarm, its corresponding data points are plotted in red, facilitating rapid detection of an alarm condition. The channel's associated alarm limits may be optionally overlaid onto the channel's plot for further information. Each data point is mouse-sensitive to provide time and numerical value indicators, and an automatic counter continually indicates the number of data points per plot. Pan and zoom features augment this display, which can represent information as graphs of actual or derived data vs. time, xy plots, scatter plots, or logarithmic scales.

Dynamic Alarm Limits

SHARP automatically determines alarm limits in real-time. The selection of alarm limits by SHARP accurately reflects each spacecraft or DSS configuration change as a result of an automated analysis of the ISOE, Predicts, and the real-time channelized engineering data from the spacecraft. Dynamic alarm limit determination eliminates the cumbersome alarm change paperwork process as well as many occurrences of false alarms.

Alarm tables, representing spacecraft configurations, are also available to operators on-line within the SHARP system. SHARP provides a user interface which allows viewing and editing of established spacecraft engineering alarm limits, DSS performance limits, ground data system limits, and residual thresholds. Authorized users may permanently alter any of these limits, and specified values may be changed temporarily for the remainder of that particular spacecraft pass. The latter capability, manual override, enables alarm suppression or closer scrutiny for any particular event, with no intervening paperwork.

System Status Displays

SHARP provides several graphical displays which represent system-wide status, including system configurations, and on-going or predicted events. SHARP automatically highlights alarmed events as they occur, and indicates the location of problems as well as the probable causes of the alarms. These displays are possible in SHARP because of the centralization of a wide variety of data sources in the system as well as its highly integrated architecture.

One such display shows a comprehensive view of Telecom link status over a settable, multiple hour time period. The display is continuously updated in real-time. Actual station coverage is illustrated, along with spacecraft transmitter power status, data rate, data outages, and real-time recording of station uplink (signal transmission) and projected downlink a round-trip light time later. Detailed analysis is performed and information is subsequently color-coded to represent changes in status. The display provides the user with such valuable information as time ranges and explanations of data outages (e.g., no station coverage or ongoing spacecraft maneuver), and can warn the operator when to expect noisy data and why.

Telecommunications system status may also be monitored using displays of functional block diagram schematics. Schematic diagrams are provided for the end-to-end communications path from the spacecraft through the Deep Space Communications Complex and Ground Communications Facility (GCF) to the Mission Control and Computing Center (MCCC) at JPL and final destination of the Test and Telemetry System (TTS) computers. The diagrams are organized hierarchically, with increasingly detailed schematics at the lowest levels. An operator can shift among the diagram displays with mouse-clicks on elements of the diagram or simple menu commands.

The appearance of the schematic diagrams is dynamically driven by changes in the ISOE, channelized data, alarm conditions, and diagnostic conclusions. The status of spacecraft and DSS components (operational, off-line, or in alarm) is depicted by color, facilitating rapid status identification at a glance.

Special Analysis Modules

The SHARP system also contains special processing modules to perform subsystem-specific analyses. Such modules are easily integrated with SHARP and can make use of the system's alarming, plotting, and diagnostic capabilities. For the Telecommunications subsystem, a special processing module which performs a Fast Fourier Transform (FFT) on DSS receiver automatic gain control (AGC) data and analyzes the results was implemented. SHARP analyzes the conical scanning component of the FFT to determine whether the DSS antenna is going off point. This is a relatively common event, which currently may take hours to detect and correct. Spacecraft and scientific information can be permanently lost when this situation occurs. SHARP's FFT display illustrates the results of a FFT process performed on 64 data points of a particular channel, and provides instant information on conical scan error. The problem can be detected in a matter of minutes, and the station can be contacted to correct the antenna movement prior to the loss of contact with the spacecraft.

Artificial Intelligence

Many of the modules of SHARP which are based on artificial intelligence techniques are written in a knowledge-based system development language and environment called STAR*TOOL. STAR*TOOL is a Common LISP-based programming language designed at JPL to achieve high computational performance in a suite of tools and techniques commonly required by research, development, and delivery of knowledge-based systems. High computational performance is essential in the scale-up of laboratory pilot systems to operational artificial intelligence systems. Since real-time performance was a baseline requirement for SHARP, STAR*TOOL became one of the fundamental tools used in system development.

AI techniques are distributed throughout all components of the SHARP system. Intelligent programming methodologies such as heuristic adaptive parsing, truth maintenance, and expert system technology enable more effective automation and thorough analysis for most SHARP functions. Artificial intelligence techniques used throughout SHARP, even for "conventional" processes such as alarm limit selection or database maintenance, have proven to be essential to real-time fault detection and diagnosis as well as the high degree of accuracy and precision in these modules. While fault detection, diagnosis, and recovery recommendation functions are localized in the "AI Module" (the structure of which is illustrated in Figure 2 and discussed below), our experience in SHARP development is that artificial intelligence functions should not be simply "layered" on a system, but must be designed as an integral aspect of the complete application.

A blackboard architecture, provided by STAR*TOOL, serves as a uniform framework for communication within the heterogeneous multi-process environment in which SHARP operates. Generally, when two or more processes are cooperating, they must interact in a manner more complicated than simply setting global variables and passing information along such paths. The STAR*TOOL blackboard message system in SHARP provides a standardized method of communication and shared-control between multiple processes for data which can not be efficiently shared through SHARP's centralized database. SHARP's individual diagnostic procedures utilize the blackboard for requesting, retrieving, and combining dependent or ancillary diagnoses.

Heuristic adaptive parsing, a technique from natural language processing, is utilized in the analysis of the raw predicts database which SHARP takes as input. Periodically the format of this data source changes without notification. This has little effect on the human mission operators who read the raw data in tabular form, but for the automated SHARP system, it would require the raw predicts parser to be rewritten to incorporate the new format. To solve the problem, SHARP utilizes Augmented Transition Network (ATN) [2] techniques to accomplish adaptive parsing. The advantage of such an ATN lies in its ability to parse the database according to semantic content rather than syntactic structure. The raw predicts database can therefore be modified and yet remain successfully parsable by SHARP without programmer intervention. This heuristically controlled, format-insensitive parsing ensures continuity despite format modifications in the generation of the raw predicts database.

The centralized database of the SHARP system serves as a central repository of all real-time and non-real-time data, and functions as a local buffer to enable rapid data access for real-time processing. Numerous database manipulation functions have been implemented, and database daemons have been constructed to implement spon-

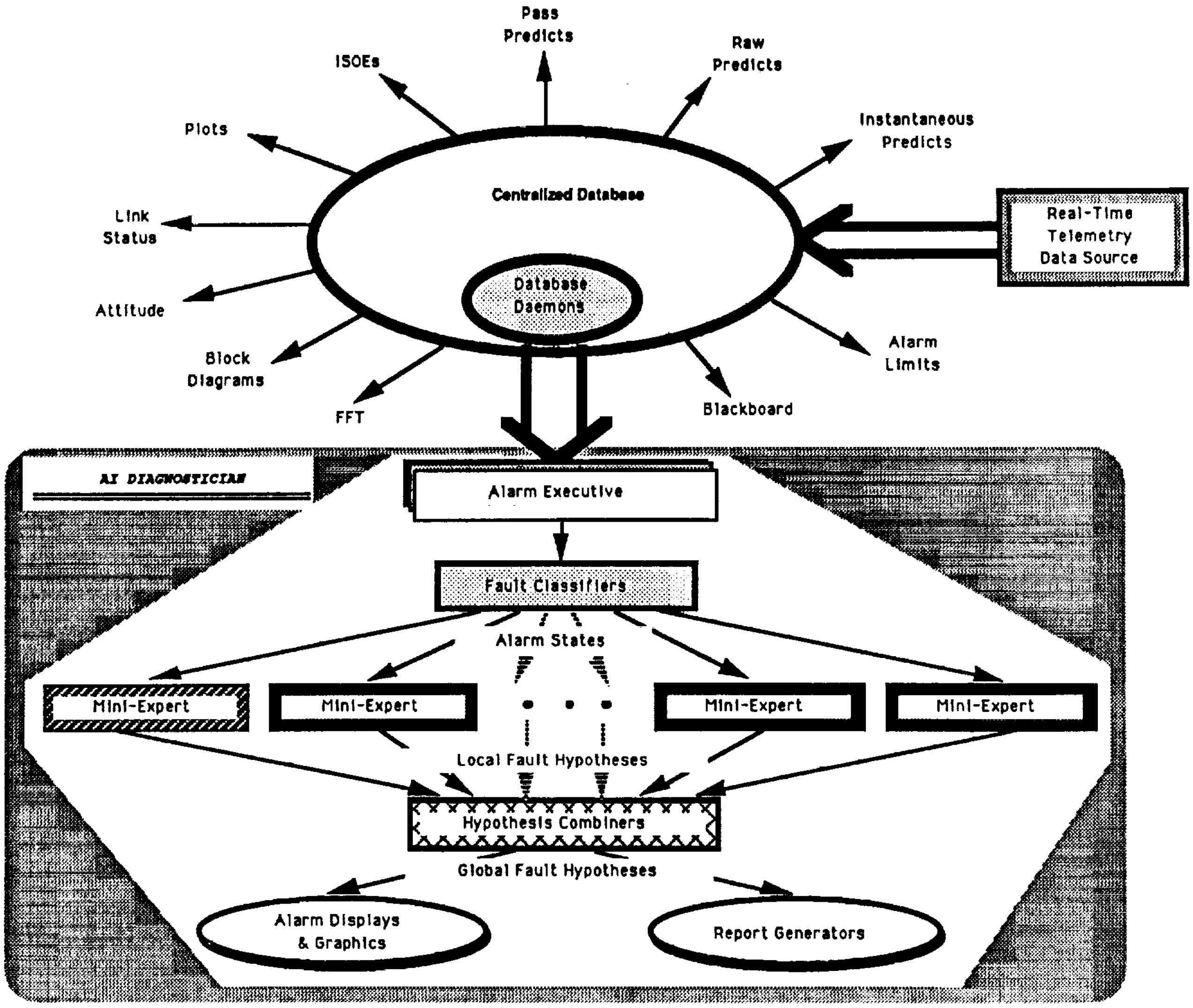

Figure 2: SHARP Artificial Intelligence Module

taneous computations [3]. Requests can be made to the database to trigger arbitrary computations when a complex combination of past, present, and future events occur. A wide selection of retrieval methods by time or value highlight the flexibility inherent in the database. Requests to the database can be made from both AI and non-AI modules of SHARP, and can be handled serially or in parallel.

Various SHARP modules represent and manipulate data symbolically rather than numerically so that particular numeric values can change without forcing the algorithms themselves to be modified. For example, to determine if a channel is in alarm, the rule interpreter manipulates one symbolic fact, "ChannelInAlarm", rather than the many numeric operations that are required to make an actual determination. This is a significant advantage as SHARP presently analyzes over 100 channels, and the alarm determination process varies from channel to channel. Symbolic representation and manipulation of data also simplifies the exchange of information between SHARP modules and reduces reliance on specific dimensionless numeric values. When needed, the specific numeric quantities underlying the symbolic data can be retrieved.

The diagnostic component of SHARP is composed of a hierarchical executive diagnostician coupled with specialized diagnostic systems, called "mini-experts". Each mini-expert is responsible for the local diagnosis of a specific fault or class of faults, such as particular channels in alarm, conical scan errors, or loss of telemetry. The decomposition of diagnostic expertise in SHARP has important and beneficial implications. The localized knowledge in each mini-expert can be maintained and modified independently of the other mini-experts. This has allowed efficient, incremental development of SHARP's knowledge bases and should improve the long-term maintainability and extendability of SHARP's knowledge. Localized knowledge also allows each mini-expert to exert more direct control over reasoning processes than would be possible in a "flat" knowledge base with awkward control constructs (a typical failing of pure production rule systems). Efficient control over reasoning processes is required for SHARP to attain real-time performance.

The specialized mini-experts can be either cooperating or non-cooperating. A non-cooperating mini-expert focuses only on its designated fault area, but a cooperating expert has the additional capability of searching beyond its local area to identify related faults that are likely to occur, and in the process thereby invoke other mini-experts to pursue the implications of these hypotheses. Cooperating experts are used in situations where the identification of a particular fault cannot be made by examining a single fault class alone.

The executive diagnostician reviews the overall situation and synthesizes hypotheses which propagate from each mini-expert. This has the effect of simplifying the ultimate diagnosis and recommendations for corrective actions which are presented to the human operator. When multiple, possibly conflicting fault hypotheses are generated, the system lists all possible causes of the anomaly and ranks each according to plausibility. In some cases, SHARP simply has insufficient data to resolve these conflicts. In other cases, conflicts may represent gaps in knowledge which must be resolved by the knowledge engineer through interaction with the domain expert. Bayesian inference processes are used for comparing multiple hypotheses and for prioritizing conflicting fault hypotheses. Bayesian inference procedures also perform uncertainty management to allow continued high performance in the presence of noisy, faulted, or missing data.

If one or more of the cooperating experts fails, the executive diagnostician will continue to operate with only a reduction in the area of local diagnosis that would have been derived from the failed mini-experts. Similarly, if the executive diagnostician fails, the cooperating mini-experts will locally diagnose the faults in isolation of multiple fault consideration, i.e., they degrade into non-cooperating mini-experts.

The executive diagnostician and mini-experts are implemented using rules that execute in pseudo-parallel pursuit of multiple hypotheses. Pseudo-parallelism (i.e., software-based, as opposed to true hardware parallelism) is implemented in SHARP using facilities provided by STAR*TOOL, which includes parallelism as a fundamental control structure. The various diagnostic rules operate in isolation of one another by executing in independent contexts [4] provided by the STAR*TOOL memory model, and communicate through the blackboard facility as described earlier. Contexts can be organized into a tree-like structure to represent contradictory information resulting from changes in facts or from the introduction of new or contradictory hypotheses.

Facilities in the STAR*TOOL truth maintenance system [5] used in SHARP handle data- and demand-driven diagnoses to ensure an appropriate balance between the persistence of hypotheses and sensitivity to new data. The truth maintenance system constantly monitors for violations of logical consistency. For example, it performs conflict checking to maintain consistency among multiple rule firings, hypotheses, and the knowledge base, and allows the context-sensitive management of alarms through a complex response system to combinations of alarm conditions. Truth maintenance techniques also provide a variety of functions for temporal reasoning in multiple fault diagnosis.

CONCLUSIONS

Spacecraft and ground data systems operations present a rigorous environment in the area of monitoring and anomaly detection and diagnosis. With a number of planetary missions scheduled for the near future, the effort to staff and support these operations will present significant challenges.

The SHARP system is an attempt to address the challenges of a multi-mission monitoring and troubleshooting environment by augmenting conventional automation technologies with state-of-the-art artificial intelligence. The artificial intelligence technology used in SHARP will endow mission operations with considerable benefits. Results of this effort to date have already begun to show significant improvements over current Voyager methodologies and have demonstrated potential enhancements to several aspects of Voyager operations. In as many areas as are automated, expert knowledge will be captured and permanently recorded, reducing the frenzied state that occurs when domain specialists announce their impending retirement. Cost reductions will occur as a result of automation and decreased requirement for 24-hour real-time operator coverage. Telecommunications domain experts have said that an application of SHARP could allow as much as a factor of five reduction in the real-time Telecom operations workforce, with comparable savings in other real-time operations areas. Automated fault detection and analysis, which present results to human operations in seconds, have greater accuracy than human manual analyses and will facilitate quicker response to mission anomalies. The time savings afforded by SHARP-like capabilities, especially during periods of unmanned operation or during emergencies, could mean the difference between the loss or retention of critical data, or possibly even of the spacecraft itself.

ACKNOWLEDGEMENTS

The research described in this publication was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. The authors wish to acknowledge the following people for their participation in the SHARP project: Harry J. Porta and Gaius Martin, software development; Boyd Madsen, expert knowledge for Voyager and Galileo spacecraft telecommunications; and Bruce Elgin and Erann Gat, software contributions. An earlier, and expanded, version of the paper was presented at the 1989 Goddard Conference on Space Applications of Artificial Intelligence, Greenbelt, Maryland, May 16-17 and is included in the preceedings of that conference.

REFERENCES

[1] James, Mark, and Atkinson, David, *STAR*TOOL — An Environment and Language for Expert System Implementation*, NTR C-17536, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California, August 19, 1988.

[2] Woods, W.A., "Transition Network Grammars for Natural Language Analysis," COMMUNICATIONS OF THE ACM, New York, NY, Vol. 13, No. 10, October, 1970, pp. 591–606.

[3] Reiger, Chuck, "Spontaneous Computation and Its Role in AI Modeling," in PATTERN-DIRECTED INFERENCE SYSTEMS, Ed., Waterman, D.A., and Hayes-Roth, Frederick, Academic Press, Inc., Orlando, Florida, 1978, pp. 69-99.

[4] McDermott, D. V., "Very large PLANNER-type data bases," Memo AIM-339, AI Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1975.

[5] Doyle, Jon, "A truth maintenance system," ARTIFICIAL INTELLIGENCE, Amsterdam, The Netherlands, Vol. 12, No.3, 1979.