N94-23900

Combining Real-time Monitoring and Knowledge-Based Analysis in MARVEL

U. M. Schwuttke, A. G. Quan, R. Angelino, and J. R. Veregge

Jet Propulsion Laboratory, California Institute of Technology,
MS 301-270, 4800 Oak Grove Drive, Pasadena, CA 91109
(ums@puente.jpl.nasa.gov)

ABSTRACT

Real-time artificial intelligence is gaining increasing attention for applications in which conventional software methods are unable to meet technology needs. One such application area is the monitoring and analysis of complex systems. MARVEL, a distributed monitoring and analysis tool with multiple expert systems, has been developed and successfully applied to the automation of interplanetary spacecraft operations at NASA's Jet Propulsion Laboratory. In this paper, we describe MARVEL implementation and verification approaches, the MARVEL architecture, and the specific benefits that have been realized by using MARVEL in operations.

1. INTRODUCTION

MARVEL (Multimission Automation for Real-time Verification of spacecraft Engineering Link) is an automated system for telemetry monitoring and analysis at NASA's Jet Propulsion Laboratory (JPL). MARVEL has been actively used for mission operations since 1989. It was first deployed for the Voyager spacecraft's encounter with Neptune and has remained under incremental development since that time, with new deliveries occurring every six to ten months. MARVEL combines standard automation techniques with embedded knowledge-based systems to simultaneously provide real-time monitoring of spacecraft data, real-time analysis of anomaly conditions, and a variety of productivity enhancements. The primary goal of MARVEL is to combine conventional automation and knowledge-based techniques to provide improved accuracy and efficiency and reduced need for constant availability of human expertise. A second goal is to demonstrate the benefit of incorporating knowledge-based techniques into complex real-time applications.

The traditional spacecraft operations environment at JPL has not relied heavily on automation. This is because until fairly recently, software technology was insufficient for meeting the complex needs of this application. The traditional approach has involved large teams of highly-trained specialists and support personnel for each spacecraft subsystem and each mission. The total operations staff for the two Voyager spacecraft during peak activity periods (such as planetary encounters) has consisted of over 100 individuals. This traditional approach has been used successfully for the Voyager mission, resulting in enormous volumes of scientific data from brief fly-by encounters.

Despite the past successes, the increasing number and complexity of missions will cause this operations approach to become less feasible for two reasons. First, the workforce costs for supporting this style of operations for multiple simultaneous missions are too great to be sustained by current NASA budgets. Secondly, with the exception of Voyager, missions will be returning significantly higher volumes of engineering and science data on a more continuous basis than in the past.

MARVEL provides user-interface functions, data access, data manipulation, data display, and data archiving within an X-windows/Motif environment. The detailed expertise for anomaly analysis is implemented with embedded knowledge-based systems. In the event of anomalies, the appropriate knowledge bases provide an analysis and recommendations for corrective action. MARVEL makes it possible for an analyst to effectively handle significantly more demanding real-time situations than in the past, because it automatically performs numerous tasks that previously required human effort. As a result of MARVEL, it has become possible for individual analysts to be responsible for several spacecraft subsystems during periods of low and moderate spacecraft activity. This is because MARVEL reduces both the level of training and the cognitive load that are required to perform routine operations.

MARVEL has demonstrated that automation enhances mission operations. Individual spacecraft analysts are no longer burdened with routine monitoring, information gathering, or preliminary analysis. Benefits include less workforce dedicated to routine tasks, earlier anomaly detection and diagnosis, leverage of scarce and valuable expertise, and reduced impact from personnel turnover. As a result, a MARVEL system for the Galileo mission (to Jupiter) is now under development.

2. REAL-TIME PERFORMANCE WITH EMBEDDED EXPERT SYSTEMS

Knowledge-based systems have not yet been sufficiently demonstrated for complex real-time applications because in such applications the amount of computation is nondeterministic, even in the presence of constant input data rates. This is being recognized as a limitation of AI systems, making it difficult to apply AI approaches where they might otherwise be useful.

While future approaches may make it possible for intelligent systems to adapt more flexibly and dynamically to real-time situations [Horvitz 1989], [Hayes-Roth 1990], [Schwuttke 1992], it is unlikely that any single new method will be able to handle *all* real-time situations. However, judicious use of existing AI methods can make it possible to obtain improved performance, both in current systems and in more dynamic systems of the future. The following paragraphs describe some of the methods used in MARVEL that enable knowledge-based techniques to enhance the capabilities of a real-time system without causing negative impact on performance. An example of knowledge based analysis in MARVEL is provided elsewhere [Schwuttke 1992b].

2.1. Knowledge-based Methods Used Only Where Essential

For certain functions, such as diagnostics and anomaly correction, expert systems provide better implementational paradigms than more efficient conventional approaches. However, expert systems usually employ interpreters to perform inferencing on the knowledge base rather than compiling the knowledge base into native code. This tends to compromise performance and can pose difficulties in applications where the fastest possible response time is a critical factor in meeting real-time constraints [Barachini 1988], [Bahr 1990].

MARVEL achieves adequate response time by placing as much of the computing burden as possible into conventional algorithmic functions written in the C language. For example, C processes handle the initial tasks of allocating telemetry to a monitoring module and detecting anomalies. If a potential anomaly is found, the corresponding telemetry is passed to the appropriate expert system for verification. If the expert system concurs that the telemetry appears anomalous, the subsystem monitor then performs an algorithmic check to determine if the anomalous telemetry is merely the result of data noise or corruption. After these preliminary tests are done and a probability of anomaly occurrence has been established, the subsystem monitor invokes knowledge-based processing for diagnosis of the anomaly and for recommendation of corrective action. This technique contributes to an overall response time that is sufficient for real-time monitoring.

2.2. Hybrid Reasoning For Improved Performance in Knowledge-Based Methods

MARVEL augments several types of reasoning with conventional software methods. For example, MARVEL uses hybrid reasoning for detecting data that is uncertain, corrupted, or of decaying validity. In the MARVEL system there are two mechanisms for detecting data integrity problems. The first mechanism uses algorithmic calculations to check the validity of quantities such as telemetry values and data modes so that obviously noisy data can be eliminated from further processing. This technique is implemented at the level of the data management process and is used to monitor simple data types. The second mechanism is knowledge-based in nature and is implemented in rules. This mechanism employs the method of expectation-based data validation [Chandrasekaran 1984]. Data of questionable integrity is verified by cross-checking it with other data sources for correlation and corroboration. If an anomaly is indicated by a new incoming telemetry word, one can validate this hypothesis by examining known related data to see if it has values that one would expect if the hypothesis was true. If the related data corroborates the initial indication, then the knowledge-based system can conclude that the new data is valid and the anomaly hypothesis is confirmed. Conversely, if the related data does not appear to be consistent with the new data then the anomaly hypothesis is not proven. MARVEL's expert systems have been explicitly designed so that they do not disregard the new data, which may provide the first evidence of a true anomaly that will eventually be confirmed by subsequent telemetry. Thus, whenever possible, the conclusions of the expert systems are based upon patterns of con-

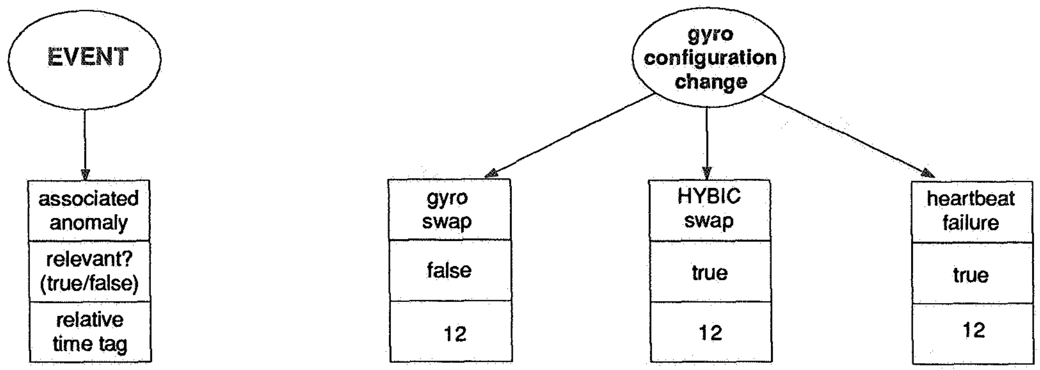

Figure 1. The event structure. The figure on the left depicts the general form of the structure; the figure on the right shows a specific instance of an event associated with three different anomalies.

sistent data rather than on a single piece of data in isolation.

2.3. Temporal Reasoning with Minimal Impact on Real-Time Processing

Real-time systems often need to reason about past events and about the order in which they occurred. The MARVEL expert systems respond to events (symptoms) indicated in the spacecraft telemetry by attempting to identify and diagnose specific subsystem anomalies that caused an event. In order to do this, the expert system may need to know about other spacecraft events that have occurred in the past and the sequence of their occurrence. This involves temporal reasoning, which is implemented in MARVEL using dynamically updated structures, as shown in Figure 1.

The structures contain the name of the event, the name of an anomaly that may have caused the event, a Boolean flag indicating whether the event has occurred and is currently relevant, and an integer specifying the sequence in which the event occurred relative to other events. The anomaly identifier is necessary since a particular event may have bearing on the diagnosis of more than one anomaly (that may or may not have occurred). Thus, a single event may point to multiple structures that are each associated with a different anomaly. The Boolean flag is set when the event associated with the structure is detected from telemetry. When this occurs, the relative time of the event is recorded in the structure. The validity of a Boolean flag expires after its corresponding anomaly is resolved, causing the flag to be reset so that it cannot contribute to the detection or diagnosis of the same anomaly unless the associated event occurs again.

These structures are intended to have minimal impact on performance. Once an event is detected, a structure is created for each anomaly whose diagnosis may depend on that event. Thus the multiple pieces of evidence that confirm the occurrence of an event need only be evaluated once, regardless of how many anomalies may be related to that event. Also, event structures are not retained indefinitely. There is a time limit beyond which an event structure is considered no longer useful for identifying and diagnosing new anomalies. After this time limit has expired, a structure's Boolean flag is reset to false regardless of whether its associated anomaly has been diagnosed. This minimizes the number of event structures that are active or relevant at any one time, which in turn reduces the number of event structure comparisons that must be performed during a rule evaluation cycle.

2.4. Multiple Knowledge Bases For Improved Focus of Attention

When significant events occur, real-time knowledge-based systems must focus their attention and resources on relevant parts of the search space in order to achieve adequate performance. Many expert system environments do not have an efficient method for doing this. One standard way to enable focus of attention is to apply different subsets of the domain rules within different contexts. MARVEL accomplishes this with separate knowledge bases for each spacecraft subsystem, and with rule contexts (mini-experts) within the individual knowledge bases.

A top-level data management process identifies incoming telemetry and determines which subsystem monitoring module to invoke for anomaly detection. When an anomaly is found, the subsystem monitor then invokes its corresponding expert system to perform the necessary analysis. This logical partitioning of input data among reasoning modules enables more rapid traversal of the search space and helps to ensure

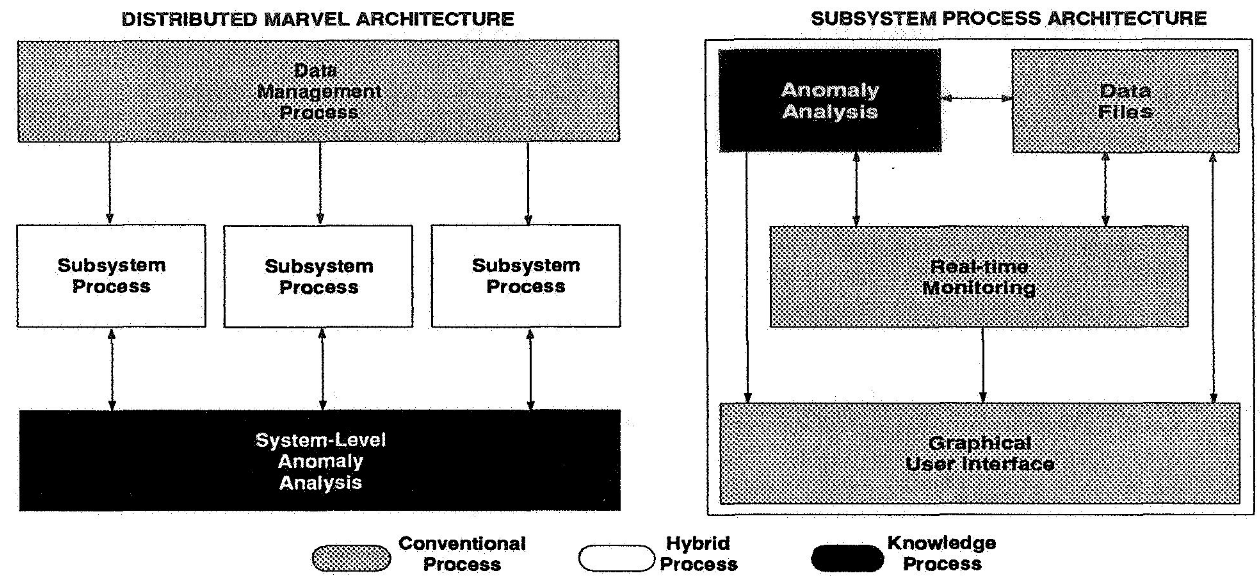

Figure 2. The MARVEL architecture, shown at the left, consists of knowledge processes, conventional processes, and hybrid processes. It can be configured to run on one to four workstations, depending on operations needs. The subsystem architecture shown at the right provides a more detailed look at the structure of MARVEL's hybrid subsytem processes.

that conclusions and responses that are not relevant to the current analysis are not pursued. This approach has also contributed to maintainability of the knowledge bases, in that several smaller knowledge bases are easier to maintain than a single large one.

3. THE DISTRIBUTED ARCHITECTURE

There are many reasons for distributed problem solving [Bond 1988]. For example, distributed systems are often characterized by greater computational speed because of concurrent operation. Also, a distributed system can be significantly more cost-effective, since it can include a number of simple modules of low unit cost. Further, distributed systems may offer a more natural way to represent certain classes of problems which contain inherently partitionable sub-processes. Each of these reasons is considered important in the mission operations environment, and as a result, a distributed MARVEL environment has been implemented.

3.1 Implementation

The distributed MARVEL architecture shown in Figure 2 is based on a central message routing scheme. The various software modules are allocated among a configuration of UNIX workstations. The data management module receives telemetry data from JPL's ground system. Each of the three subsystem monitors provide functions such as validation of telemetry, detection of anomalies, diagnosis of causes, and recommendation of corrective actions. The latter two functions are provided through intelligent reasoning modules that are embedded within each of the individual subsystem monitors. The remaining modules include the display processes for each of the three subsystem monitors, and the system-level reasoning module for diagnosing anomalies that manifest themselves in multiple subsystems (and therefore cannot be completely analyzed by any one subsystem alone).

The interconnectivity of the distributed system is provided by a TCP/IP central router program and a set of messaging routines that are linked into the subsystem processes. All MARVEL processes are connected to the central router by UNIX sockets.

3.2 Performance

For realistic systems with non-negligible communication overhead, the critical measure curve is related to the speedup S(N) [Fox 1988] defined as

$$S(N) = T_{seq} / T_{conc}(N).$$

In this equation, N denotes the number of processors, and T_{seq} and $T_{conc}(N)$ refer to the execution times of the sequential program and the distributed program on N processes, respectively. Distributed systems with a speed-up S(N)= 0.8N are considered to be very efficient [Fox 1988]; the minimum desired speedup for a distributed MARVEL system was 0.6N.

The basic measurement of performance for the distributed MARVEL is S(N). However, it has not been possible to measure a unique value S(N) because of the heterogeneous nature of the MARVEL modules. This heterogeneity arises because the processing load of the four basic components (the data management module and the three subsystem modules) are not identical. Our alternative to this measurement is the lowest speedup of the individual subsystems. With a four-processor implementation, a speedup of 3.6, or 0.9N was observed. This result indicates that MARVELis a highly efficient distributed system. Two factors contribute to the success of these results. The first of these is the modularity inherent in the application (as is common in many other complex applications). The second factor is a distributed design that effectively minimizes the need for interprocess communication.

4. DISCUSSION

The development of MARVEL has shown the value of a rapid development approach that emphasizes top-down design and bottom-up implementation. The implementation has been modular and incremental, with frequent deliveries (every 5 to 10 months) of new or enhanced capabilities. The result has been an automated tool that began as a simple software module for automating straightforward tasks and that has evolved over a period of five years into a sophisticated system for automated monitoring and analysis. The initial modular design enabled MARVEL to be developed incrementally, with each subsequent delivery providing greater breadth to the application. This approach has been instrumental to the success of the effort, because it was compatible with available budgets and encouraged user and sponsor confidence with frequent demonstration of results. In addition, the approach has influenced the validation and use of MARVEL as described below.

4.1 Verification of Expert Systems

MARVEL verification has been ad hoc, largely because of a lack of formal procedures. Two methods have been used: carefully engineered test cases and on-line verification (involving parallel operations with human analysts). Most problems were detected with the use of test-cases, but some were not detected until the software was used in an on-line mode. Newly delivered modules were subject to an on-line verification period, typically on the order of one month. On-line verification allowed continual comparison with the results of manual approaches so that reasonable confidence in the automated system could be obtained.

The primary advantage of this approach has been its minimal impact on development costs. However, there have also been several disadvantages. On a few occasions, minor bugs in MARVEL went undetected until the end of the parallel-operations phase. This temporarily undermined user-confidence, particularly with users who were not enthusiastic about automation. A second disadvantage is that without formal verification procedures there have been occasional questions about whether MARVEL should be formally accepted as "official"ground software for mission operations. The current lack of solid answers in this area would prevent the use of MARVEL's AI modules for certain tasks that are considered mission critical, but has not prevented the use of these modules in an advisory mode.

4.2 Use and Benefit of MARVEL

MARVEL has been in active, daily use since it was first deployed in 1989. The current version performs real-time monitoring for three spacecraft subsystems. These functions previously required the presence of human analysts for a minimum of eight hours per subsystem per day. During planetary encounters, human presence was required on a twenty-four hour basis. MARVEL also performs non-real-time functions that were previously unautomated. These functions save from 30 minutes per week (for clock drift analysis) to 2 hours per day (for daily report generation). During MARVEL's on-line tenure, it has detected all the anomalies that occurred within its domain. During parallel operations, most anomalies were first detected by MARVEL. On two occasions MARVEL detected anomalies that operations personnel believe may have been overlooked by them because the quantities of data transmitted at those times were larger than could reasonably be handled without automated assistance.

Initial emphasis on productivity enhancement resulted in an early version of MARVEL that (according to the responsible operations supervisor) would have made real-time CCS subsystem workforce reductions of 60% (3 out of 5 analysts) possible during the Neptune encounter, had MARVEL been approved for stand-alone rather than parallel operations. Subsequent to the Neptune encounter, significant workforce reductions have been implemented for all spacecraft subsystems because of post-encounter budget cuts. MARVEL played a substantial role in simplifying the transition to reduced workforce for the subsystems for which it was available.

The initial emphasis on productivity enhancement temporarily curtailed the development of MARVEL's expert systems, because it was perceived that diagnostic systems did not improve efficiency of operations. This perception stemmed from the observation that anomaly analysis was only required in the presence of spacecraft anomalies, which did not occur with sufficient frequency to warrant an automated approach, particularly since human confirmation of the expert system analysis would still be required.

However, the post-encounter workforce reductions caused renewed interest in expert systems. However, the goal is no longer workforce reduction, but the preservation of mission expertise. The current analysts are new to the mission and, for the most part, do not have the experience of the previous staff. The new personnel will have fewer opportunities to gain such experience: although the Voyager interstellar mission is scheduled to continue until approximately 2018, spacecraft activity is at a low level. This means that there are far fewer opportunities for learning about the spacecraft and its operation. There is concern that analysts with the experience to handle future anomalies will be less readily available, or that they will have retired. As a result, MARVEL's expert systems are being expanded to provide information that is based on the expertise of former analysts.

6. SUMMARY

This paper has presented methods for combining conventional software with AI techniques for use in real-time problem solving systems. The methods described have been presented in the context of the MARVEL system which has provided a continuous and evolving demonstration of the success of the approach since Voyager's Neptune encounter in August 1989. These techniques have been implemented in a distributed environment that will accommodate the real-time demands of NASA's more recently launched interplanetary missions.

7. ACKNOWLEDGMENT

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology under a contract with the National Aeronautics and Space Administration. The authors wish to acknowledge support from JPL's Voyager Project, and Director's Discretionary Fund.

8. REFERENCES

Bahr, E. and Barachini, F. 1990. Parallel PAMELA on PRE. In *Parallel Processing of Engineering Applications*, ed. R. A. Adey, 209-219. New York: Springer-Verlag.

Barachini, F. and Theuretzbacher, N. 1988. The Challenge of Real-time Process Control for Production Systems. Proceedings of AAAI-88, 705-709.

Bond, A. H., and Gasser, L. 1988. *Readings in Distributed Artificial Intelligence*. San Mateo: Morgan Kaufmann.

Chandrasekaran, B. and Punch, W. 1987. Data Validation During Diagnosis: A Step Beyond Traditional Sensor Validation. In Proceedings of AAAI-87, 778-782.

Fox, G., et al. 1988. *Solving Problems on Concurrent Processors*, Volume I. New Jersey: Prentice Hall.

Hayes-Roth, B. 1990. Architectural Foundations for Real-time Performance. *Artificial Intelligence Journal*, 26: 251-232.

Horvitz, E. J.; Cooper, G. F.; and Heckerman, D. E. 1989. Reflection and Action Under Scarce Resources: Theoretical Principles and Empirical Study. Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, 1121-1127.

Schwuttke, U. M. and Gasser, L. 1992. Real-time Metareasoning with Dynamic Tradeoff Evaluation. *Proceedings of AAAI-92*.

Schwuttke, U. M.; Quan, A. G.; Angelino, R; et al 1992b. MARVEL: A Distributed Real-time Monitoring and Analysis Application. In *Innovative Applications of Artificial Intelligence*, Volume 4, ed. C. Scott and P. Klahr, Boston: MIT Press. 1992.