N89-26595

REDEX: The Ranging Equipment Diagnostic Expert System

Edward C. Luczak and K. Gopalakrishnan
Computer Sciences Corporation
4600 Powder Mill Rd.
Beltsville, MD 20705

David J. Zillig
Telecommunication Systems Branch, Code 531
NASA Goddard Space Flight Center
Greenbelt, MD 20771

ABSTRACT

This paper describes REDEX, an advanced prototype expert system that diagnoses hardware failures in the Ranging Equipment (RE) at NASA's Ground Network tracking stations. REDEX will help the RE technician identify faulty circuit cards or modules that must be replaced, and thereby reduce troubleshooting time. It features a highly graphical user interface that uses color block diagrams and layout diagrams to illustrate the location of a fault.

A semantic network knowledge representation technique was used to model the design structure of the RE. A catalog of generic troubleshooting rules was compiled to represent heuristics that are applied in diagnosing electronic equipment. Specific troubleshooting rules were identified to represent additional diagnostic knowledge that is unique to the RE. Over 50 generic and 250 specific troubleshooting rules have been derived.

REDEX is implemented in Prolog on an IBM PC AT-compatible workstation. Block diagram graphics displays are color-coded to identify signals that have been monitored or inferred to have nominal values, signals that are out of tolerance, and circuit cards and functions that are diagnosed as faulty. A hypertext-like scheme is used to allow the user to easily navigate through the space of diagrams and tables. Over 50 graphic and tabular displays have been implemented.

REDEX is currently being evaluated in a stand-alone mode using simulated RE fault scenarios. It will soon be interfaced to the RE and tested in an online environment. When completed and fielded, REDEX will be a concrete example of the application of expert systems technology to the problem of improving performance and reducing the lifecycle costs of operating NASA's communications networks in the 1990's.

1.0 INTRODUCTION

One of NASA's big challenges for the 1990's is to contain the operations and maintenance costs of its mission operations and ground support systems while at the same time sustaining the current high levels of mission performance and readiness. Expert systems technology is being evaluated by several NASA organizations because of its potential for capturing the specialized expertise of operations and maintenance personnel and making it available rapidly, consistently, around the clock, and at multiple locations.

One of the more promising areas for applying expert systems technology is in the area of problem and fault diagnosis. Prototype "diagnostic assistant" tools are being developed to help operators and engineers detect, isolate, diagnose, and in some cases recover from operational problems and system faults.

A number of prototype expert systems are being developed and evaluated by NASA to diagnose problems onboard orbiting spacecraft, including the Tracking and Data Relay Satellite (TDRS) [1], the Hubble Space Telescope [2], and the Gamma Ray Observatory (GRO) [3]. Another prototype is being developed to diagnose faults in communications links with the Cosmic Background Explorer (COBE) [4].

Expert systems are also being evaluated for use in detecting and diagnosing faults in NASA ground systems. Use of diagnostic assistant tools can reduce the amount of time required to diagnose complex ground systems, and allow them to be repaired and restored to mission support more rapidly. Diagnostic expert systems for NASA ground systems are being developed and evaluated for Shuttle launch preparation hardware [5], telemetry processing systems [6], and an institutional local area network (LAN) [7]. An advanced fault isolation expert system that diagnoses problems in the ground and space components of the NASA Space Network [8,9] is approaching operational use.

This paper is a case study description of the Ranging Equipment Diagnostic Expert System (REDEX). REDEX is an advanced prototype expert system that diagnoses hardware failures in the Ranging Equipment (RE) at NASA's Ground Network tracking stations. When an RE failure occurs, REDEX will help the RE technician identify the faulty circuit cards or modules that must be replaced. REDEX is designed to assist the technician by significantly reducing troubleshooting time. It features a highly graphical user interface that uses color block diagrams and layout diagrams to illustrate the location of a fault. When testing and evaluation are completed, REDEX is planned to be installed at several NASA tracking stations.

The following sections describe the characteristics of the RE diagnosis problem (Section 2), the approach used in developing REDEX (Section 3), key aspects of the design of the expert system (Section 4), the current status of the implementation (Section 5), key observations and conclusions that resulted from this work (Section 6), and directions for future work (Section 7).

2.0 THE PROBLEM

The NASA Ground Network (GN) consists of a set of tracking stations that are used to track Space Shuttle launches and various unmanned spacecraft. The GN and its affiliated facilities include tracking systems at Bermuda, Merritt Island, FL, and Wallops Island, VA, and engineering test beds at Goddard Space Flight Center (GSFC) and Johnson Space Center (JSC).

The RE [10] is one of the principal equipment items in the S-Band ranging system at the tracking stations (Figure 1). The ranging system uses a 9-meter dish antenna to send and receive tracking signals from the target spacecraft. The RE determines the range and Doppler (velocity) of the spacecraft, and transfers this data to a tracking data processor. The data is then transmitted via the NASA Communications Network (NASCOM) to GSFC.

The RE occupies one full equipment rack, and includes over 50 custom-designed circuit boards, seven power supplies, and an embedded PDP-11/73 computer and color display terminal. The equipment includes a variety of high and low frequency analog circuitry, digital electronics, and several embedded microprocessors. It is arguably the most diverse and complex single rack of electronics at a GN tracking station. When a hardware failure occurs in the RE, the technician must identify the faulty circuit board or module and replace it with an on-site spare as rapidly as possible to restore the RE to service for mission support.

The RE design includes a Ranging Internal Monitor (RIM) system consisting of 74 test points that are distributed throughout the RE. These RIM points continuously monitor key signal strengths, DC voltages, frequencies, and logic levels. The RIM points status is gathered and reported to the PDP-11 computer, which displays the data upon request to the RE operator. RIM points that have strayed from their nominal range are displayed as red, and those within tolerance are shown in green. A fault is indicated when one or more of the RIM points are red. The RE technician uses the array of RIM point status data to begin troubleshooting the failure.

The troubleshooting process may take an hour or more, requiring the technician to trace signals and status through a set of several hundred block diagrams and circuit schematics. A major difficulty is that a single fault often causes many RIM points on different circuit boards to become red, because of the propagation of bad signals throughout the RE. The diagnostic troubleshooting process involves reasoning about signal flow and cause-effect relationships in electronic systems. This reasoning process requires knowledge of the available RIM point values, knowledge of general electronic troubleshooting rules, and specific knowledge of the design and operation of the RE.

The REDEX diagnostic assistant was conceived to reduce RE troubleshooting time, which is especially critical in the event of an RE failure during a Shuttle launch countdown. This problem is appropriate for an expert system solution because it involves reasoning processes that can be well defined, and requires specific knowledge spanning only a narrow domain. As shown in Figure 1, REDEX resides on a workstation and obtains the RIM point status data from the RE via a communications link. When the RE internal monitoring system has detected a fault, REDEX can diagnose the situation and identify the circuit cards that are suspected to be faulty. REDEX can display the RIM point status data and the results of the diagnosis using displays of RE block diagrams, rack and drawer layout diagrams, and tabular data.

3.0 APPROACH

Compared to traditional software development efforts, the software industry has relatively little experience in developing expert systems. It is widely recognized that the successful development of expert systems requires a methodology that differs from those typically used for

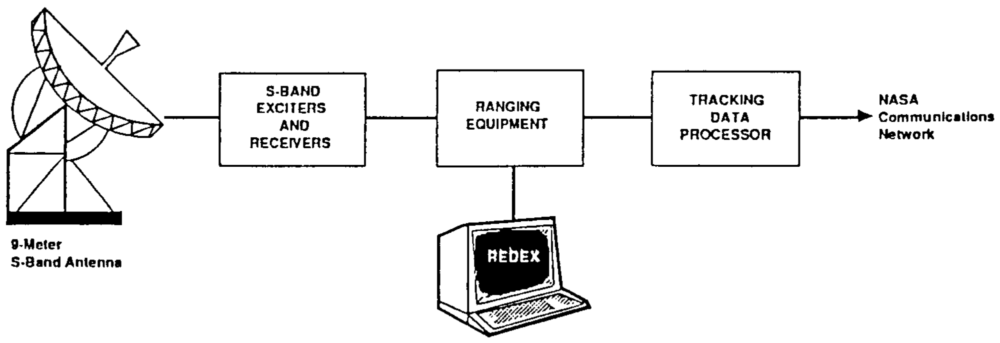


Figure 1. NASA Ground Network Tracking Station

traditional software. Unfortunately, insufficient data has been gathered to allow the evaluation and selection of such a methodology with confidence.

REDEX is being developed using a methodology that has several of the characteristics discussed in [11]. The approach has followed the overall framework of the standard software lifecycle "waterfall model", involving requirements definition, design, implementation, and testing. Requirements were defined in a knowledge acquisition and knowledge representation process, and then formalized in a requirements document [12]. The software design was developed and specified in a design document [13].

However, iteration is a key component of the REDEX development approach, and has been driven by a continuing evolutionary prototyping effort. (See Section 3.3.) The requirements were first informally defined, then prototyped, and then formally documented. The design followed the same sequence. This approach allowed the requirements and the design to be evaluated and validated via prototyping prior to formal documentation. Both the requirements definition and design were allowed to evolve during the prototyping. This iterative approach is characteristic of the "spiral model" of software development. REDEX is being developed using many elements of the spiral-based prototyping methodology described in [14] and [15].

The documentation steps were valuable because they provided (a) the discipline of clearly defined development milestones, (b) a mechanism for careful definition of the state of the requirements and design, and (c) a vehicle for external review and validation. Both the requirements and design documents will be periodically updated.

The following sections describe several aspects of the approach in more detail.

3.1 KNOWLEDGE ACQUISITION

The knowledge needed to develop REDEX was acquired from the three sources described below. The knowledge acquisition team included persons familiar with electronic systems, fault analysis, and knowledge representation.

(a) Technical documentation - An intensive study of RE design documentation, circuit diagrams, operations manuals, and maintenance documentation was conducted.

(b) Training class - REDEX knowledge engineers attended the NASA training class that teaches new operators and technicians how to operate, troubleshoot, and repair the RE. The class included hands-on operation of the RE.

(c) Interviews with expert - The principal instructor for the training class is an expert in troubleshooting the RE; he was interviewed to obtain additional detailed information.

3.2 KNOWLEDGE REPRESENTATION

The knowledge required to diagnose RE faults includes knowledge of the design of the RE and knowledge of the procedures or rules needed for troubleshooting. The techniques described below were used to represent this knowledge, and to document it in [12].

Knowledge of the RE design was represented using semantic networks. A model of the RE design structure was constructed representing RE design objects and their relationships. The objects represented include signals, RIM test points, circuit boards, functions, switches, power supply voltages, circuit drawers, and functional subsystems. A function is generally a portion of a circuit board.

Figure 2 illustrates the full set of object classes used in the RE model, and the relationships that can exist between instances of these classes. Both hierarchical and heterarchical relationships are included. An object instance is characterized by a set of values for attributes that are associated with its class.

Semantic network diagrams were drawn to capture the relationships "input-to", "output-from", "supplies", and "status-of" between signals, RIM points, power expressions, and functions. Diagramming conventions were defined to simplify the construction and interpretation of these diagrams. They have the appearance of structured block diagrams. Approximately 20 semantic network diagrams were constructed and included in [12]. They provide a readable, formal definition of the lowest level of the RE model that REDEX needs for diagnostic reasoning.

Troubleshooting knowledge was represented using rules in an "IF (conditions) THEN (conclusions)" format. A notation for representing these rules was developed that is independent of any specific implementation language. This independence is analogous to the way that Program Design Language (PDL) used in traditional software design is independent of the language to be used for coding. A simple example rule is:

```
IF red(RIM22) and green(RIM29)
   and green(RIM31)
THEN suspect(a6a13f1)
```

These rules are called "specific troubleshooting rules" because they represent specific knowledge about assessing the health of a function or a signal in the RE. Over 250 specific rules were derived.

The specific rules were derived from the semantic network diagrams by searching for certain patterns of monitored and unmonitored signals around function boxes that enable diagnostic inferences to be made. These patterns are called "generic troubleshooting rules". Figure 3 illustrates two simple generic rules. A catalog of over 50 generic rules was compiled. These generic rules represent general troubleshooting knowledge and are applicable to other equipment that is similar to the RE.

3.3 PROTOTYPING

REDEX has been developed using an evolutionary prototyping process, in which a series of five prototype stages were defined and built. This process could be viewed as a spiral, in which each stage involves definition, implementation, testing, and evaluation activities. Each prototype stage was only a few months in duration. Each stage was defined such that its successful completion would reduce development risks in an important area.

The first stage began early in the requirements definition activity; its objective was to demonstrate the feasibility of representing RE troubleshooting rules in the Prolog language. A small set of rules required to diagnose faults in one functional subsystem of the RE was implemented and demonstrated as a "proof-of-concept".

Successful completion of this first stage led to the second prototype stage, whose objective was to encode the knowledge needed to diagnose faults in the entire RE. This prototype was implemented incrementally, adding and testing the knowledge for one RE subsystem at a time until all ten subsystems were implemented. The requirements document was written when this stage was successfully completed and demonstrated.

The objective of the third prototype stage was to develop and evaluate the concepts of the user interface design. (The first two prototype stages involved only a primitive user interface.) Development and demonstration of this prototype led to many improvements in the design concept. The objective of the fourth prototype stage was to fully implement the user interface. The design document was written when this stage was successfully completed and demonstrated.

The fifth prototype stage is currently underway. In this stage, REDEX is being refined to correct diagnostic errors discovered during testing, made more robust with error handling capabilities, and enhanced to incorporate feedback received from user evaluations. This stage also adds the capability to communicate with the RE.

4.0 DESIGN

The design of REDEX defines a system environment, a control structure for implementing the knowledge described in Section 3.2, a user interface, and a data interface with the RE. The design evolved and was tested using the prototyping process discussed in Section 3.3. The following sections describe several key aspects of the design.

4.1 SYSTEM DESIGN

REDEX is designed to execute on an IBM PC AT-compatible workstation with a high resolution (640x480 pixels) color display. The software is written in the Turbo Prolog language, which runs under MS-DOS. An evaluation of the effectiveness of this language for implementing REDEX can be found in [16]. Turbo Prolog was selected for REDEX implementation because it provides: (a) a straightforward and efficient representation of the troubleshooting rules, (b) the facilities and flexibility needed for building the graphical interface, (c) a good software development environment, (d) excellent runtime

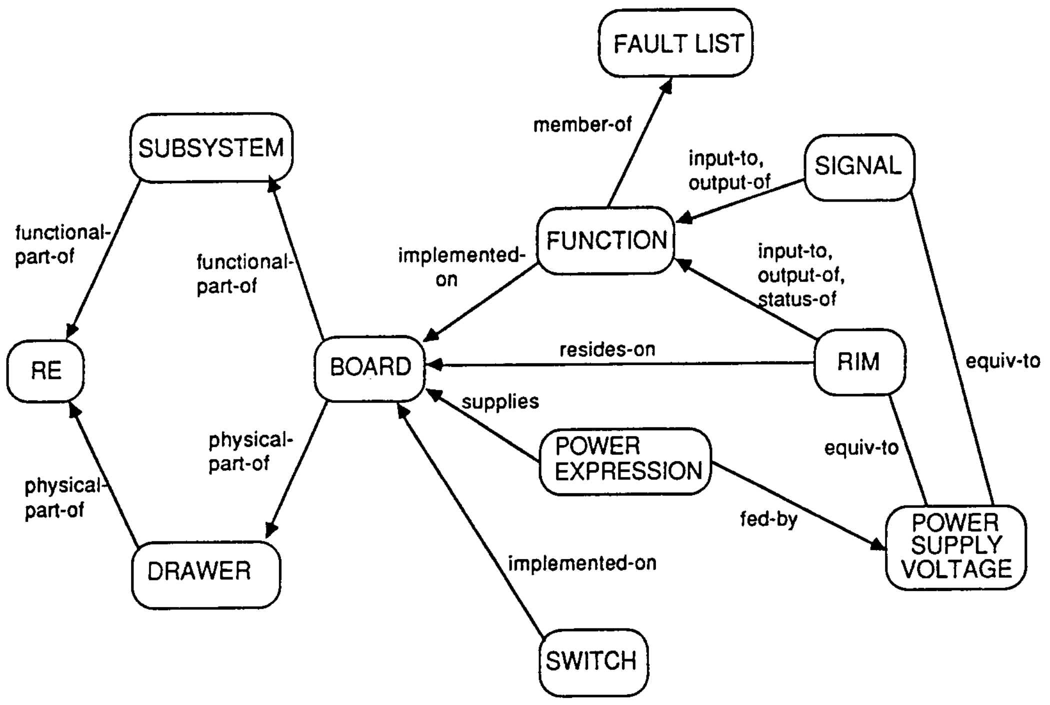
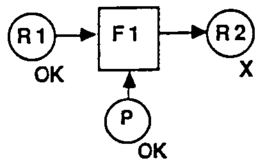


Figure 2. REDEX Object Classes and their Relationships

RULE S1-A "GOOD INPUT - BAD OUTPUT" RULE



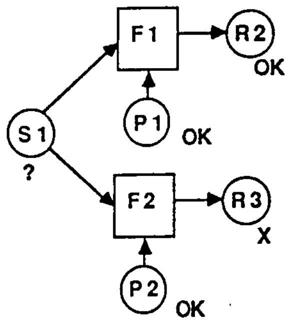
CONDITIONS:

GREEN(R1)
RED(R2)
GOOD(P)

CONCLUSION:

SUSPECT(F1)

RULE S3-A "COMMON UNKNOWN INPUT SIGNAL " RULE #1



CONDITIONS:

UNKNOWN(S1)
GREEN(R2)
RED(R3)
GOOD(P1)
GOOD(P2)

CONCLUSIONS:

GOOD(S1)
HEALTHY(F1)
SUSPECT(F2)

Figure 3. Example Generic Troubleshooting Rules

performance, and (e) it is significantly less costly than other expert system development tools..

Figure 4 shows the REDEX system block diagram. The knowledge base contains Prolog clauses that represent facts and rules. The inference engine is provided by the built-in control structure of Prolog. Procedures are implemented using Prolog predicates.

4.2 DIAGNOSTIC FUNCTIONS

REDEX uses a rule-based design approach to implement the RE troubleshooting rules. The design also incorporates a representation of the objects in the RE and their relationships. The overall control structure is goal-directed backward rule chaining. Figure 5 is an RE fault tree that illustrates the goal decomposition. The top node, "Diagnosis", represents the primary goal of determining the reason for an RE red health status (i.e. one or more RIM points are red). The second level of nodes represents three types of possible reasons: that an RE input signal is bad, that an RE function contains a fault, and that a reportable status condition that affects RE health status has occurred. The top level "Diagnosis" goal is satisfied if one of the second level goals is satisfied.

The third level nodes in Figure 5 represent specific reasons for red health status, e.g. specific functions that could contain a fault. The fourth level nodes are troubleshooting rules (R). When one of these rules fires, its conclusion may be that a signal is good or bad with a given confidence factor, that a function or set of functions is suspected to contain a fault, or that a reportable status condition exists. The conditions in these rules involve the color of RIMs, the positions of RE configuration switches, the values of power expressions, and the status of evaluated signals. In order to evaluate these conditions, many lower-level rules not shown in the figure may have to be fired.

Although it is unlikely that more than one fault will occur at a time, REDEX always searches the entire fault tree in order to detect multiple fault conditions if they occur.

4.3 USER INTERFACE

The REDEX user interface design is based on the use of menus for soliciting user command entries, and the use of color-coded graphic diagrams and tables for displaying RE status data and diagnostic conclusions. Menus are displayed in pop-up windows. One of the options on the main menu is an on-line help facility that provides assistance in the use of the menus and graphic displays.

The graphic displays include functional block diagrams of the RE design and physical layout diagrams of the equipment rack and drawers. This is a natural data display approach because the RE operator learns how the RE operates and how to troubleshoot it using the same type of diagrams. On these diagrams, signals that have been monitored or inferred to be good are displayed in green, while those out of tolerance are shown in red. The circuit functions, circuit boards, and rack drawers that are diagnosed as faulty are displayed flashing. Over 50 graphic and tabular displays have been implemented. Figure 6 shows a typical block diagram display and Figure 7 shows a typical rack layout diagram display.

A hypertext-like scheme is used to allow the REDEX user to easily navigate through the set of diagrams and tables. Link "buttons" are defined on each diagram and table. By moving the cursor to one of these link buttons and pressing a function key, a new diagram or table is displayed. These links allow the user to rapidly move up, down, across, and between the hierarchies of block diagrams, layout diagrams, and tabular data. Links between hierarchies are defined in such a way to support the user's thinking process. For example, when the user is viewing a table of RIM point values, he is likely to want to see the location of a given red RIM in the RE. A link button is therefore defined for each RIM point on the table to allow the appropriate subsystem block diagram to be quickly displayed.

5.0 IMPLEMENTATION AND TESTING STATUS

REDEX has reached the advanced prototype stage of implementation. All data interface, user interface, performance, and diagnostic functional requirements specified in the initial requirements document have been satisfied. Over a dozen demonstrations have been given to various review groups, and the feedback obtained has been factored into the evolving prototype. Reviewers have included RE operations personnel, RE design engineers, system engineers, hardware engineers, software engineers, human factors specialists, and artificial intelligence specialists.

REDEX is currently being tested in an offline mode, not directly interfaced with the RE. Testing has been performed using the following three techniques:

(a) Manually constructed test cases - Diagnostic test cases were manually constructed and executed. A test case consists of a set of red RIMs, a set of RE configuration switch settings, and an associated expected diagnosis. Sufficient test cases were designed to exercise each functional diagnostic rule in isolation or in a small set. This testing technique helped identify local coding errors in individual diagnostic rules. It is roughly analogous to structural unit testing of traditional software.

(b) Fault simulator - An RE fault simulator program was developed that simulates the propagation of fault effects through the RE. When the location of a hypothetical fault

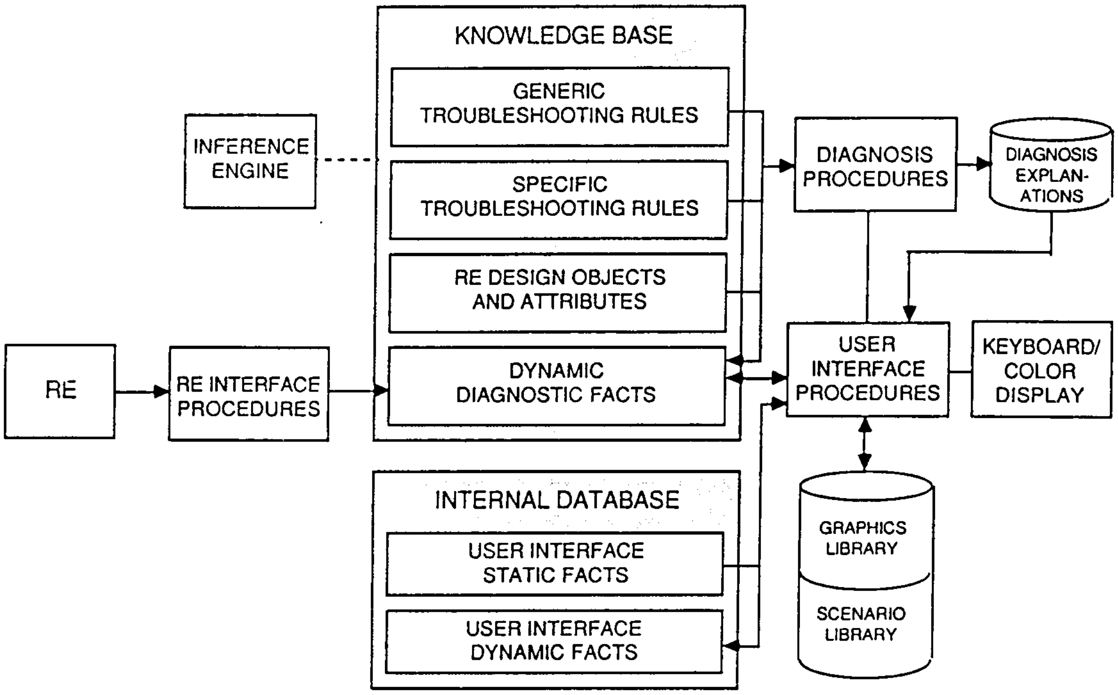


Figure 4. REDEX System Block Diagram

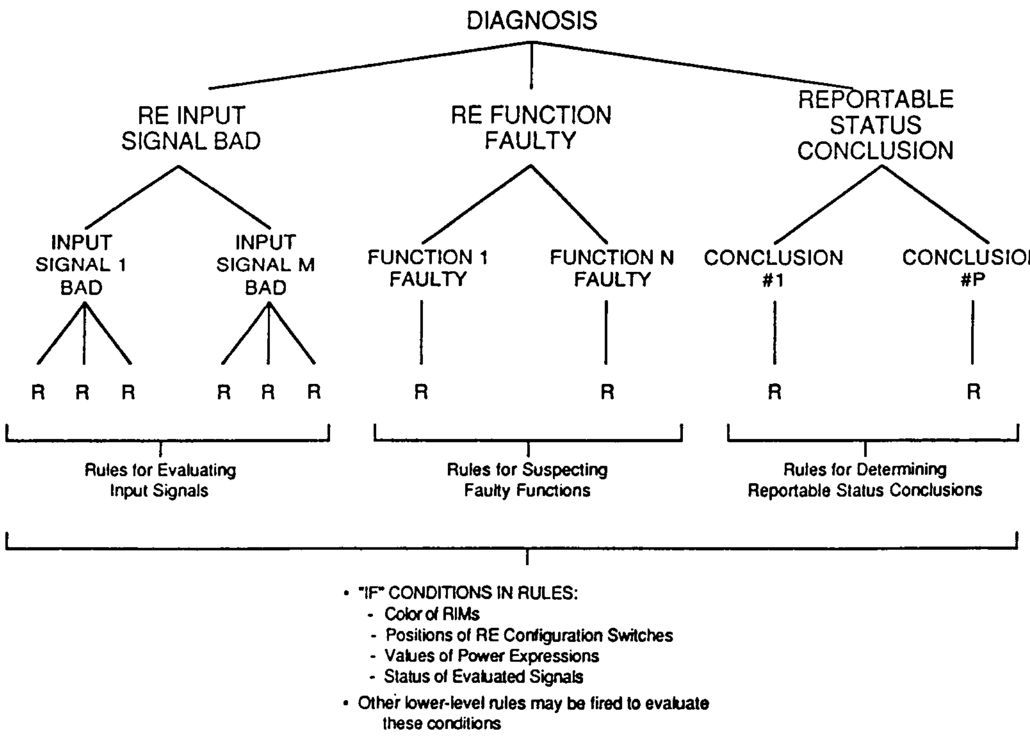

DIAGNOSIS
RE INPUT SIGNAL BAD
RE FUNCTION FAULTY
REPORTABLE STATUS CONCLUSION
INPUT SIGNAL 1 BAD
INPUT SIGNAL M BAD
FUNCTION 1 FAULTY
FUNCTION N FAULTY
CONCLUSION #1
CONCLUSION #P
R R R R R R R R R R
Rules for Evaluating Input Signals
Rules for Suspecting Faulty Functions
Rules for Determining Reportable Status Conclusions
• "IF" CONDITIONS IN RULES:
- Color of RIMs
- Positions of RE Configuration Switches
- Values of Power Expressions
- Status of Evaluated Signals
• Other lower-level rules may be fired to evaluate these conditions

Figure 5. RE Diagnosis Fault Tree

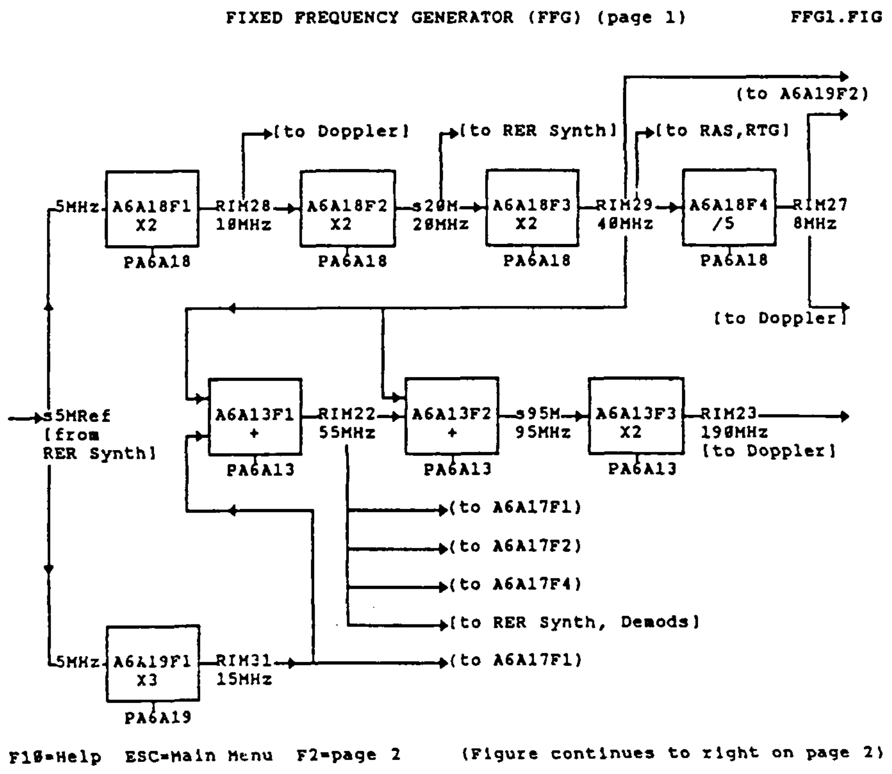


Figure 6. Example REDEX Block Diagram Display

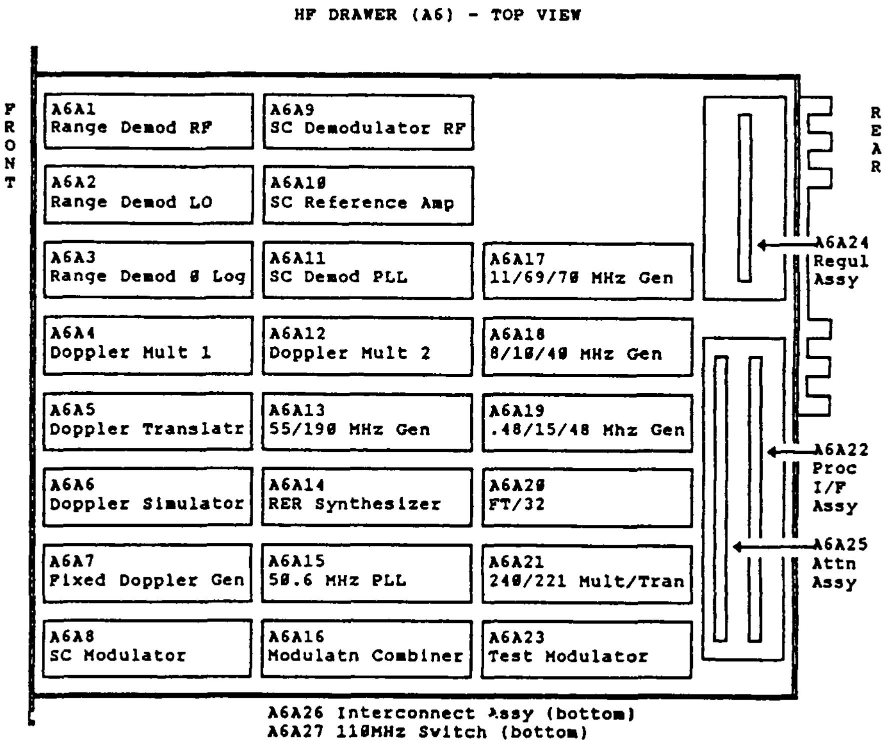


Figure 7. Example REDEX Physical Drawer Layout Display

is provided to this program, it generates the set of red RIM points that would result if the effects of the fault propagated fully. This program was used to automatically generate a large number of realistic test cases. An error was uncovered whenever a test case based on a hypothetical fault in function F was diagnosed by REDEX to indicate a fault in some location other than F. (In some cases, the error uncovered is in the fault simulator program, not in REDEX.) This testing technique helped identify errors in diagnostic strategy and in the RE design model in the knowledge base. The simulator should also be a valuable tool for training future RE operators.

(c) RE communications emulator - Because the RE has not yet been available for direct interface testing, an IBM PC was used to emulate the RIM point data transmission function of the RE. This PC was linked to the REDEX workstation using an RS232C communications cable, in the same way that the RE is planned to interface to REDEX. Actual RIM value reports obtained from the RE were entered into the PC, and transmitted over the cable to REDEX. This testing technique helped identify errors in the REDEX communications interface procedures.

REDEX will soon be interfaced with the RE for online testing. After online testing and evaluation is completed, REDEX is planned to be installed at four or more NASA tracking stations.

6.0 CONCLUSIONS

The development of REDEX has led to several observations and conclusions about the design and development methodology for ground system diagnostic expert systems:

(a) Equipment monitor points - The diagnostic strategy in REDEX depends on the availability of the set of status monitoring points within the RE. The internal monitoring system that made REDEX possible was specified as part of the RE design. Such monitoring points would be difficult to retrofit into a system like the RE after it is implemented. When a new equipment system is specified, it is paramount to consider including requirements for an internal monitoring system to provide status data for a diagnostic assistant. Monitor points should be placed at sufficient density to allow high confidence diagnosis to the desired system level (e.g. independent subsystem, or field replaceable circuit board) depending on the sparing and repair philosophy for the equipment. A mechanism for gathering the monitor data and delivering it to the diagnostic assistant, such as via a communications interface, should also be specified.

(b) Knowledge acquisition and requirements definition effort - The amount of time needed for these activities was six months, or about twice that amount originally planned. Despite undesirable schedule delays, it was recognized that careful performance of these activities was essential to the success of the project. Care should be taken when scheduling an expert system development effort to avoid underestimating time for these early activities.

(c) Requirements document - Detailed diagnostic functional requirements for REDEX were specified in a formal requirements document using semantic network diagrams and language-independent rules. The discipline of producing this document was found helpful in crystalizing, evaluating, and verifying requirements. The document was produced and will be periodically updated in parallel with the prototyping effort.

(d) Prototyping - Evolutionary development by iterative prototype enhancement was found to be an effective approach for developing REDEX. Frequent demonstrations of prototype status were important in fueling the evolution, particularly of the user interface.

(e) Testing by fault simulation - The development of the RE fault simulator diverted resources from the development of REDEX, but was assessed to be well worth the cost. The fault simulator was used to generate simulated monitor data for a large number of repeatable fault conditions, and enabled comprehensive testing of REDEX.

(f) Generic troubleshooting rules - The catalog of generic troubleshooting rules compiled for diagnosing RE faults is applicable to other equipment that is similar to the RE. Development of diagnostic assistants for other equipment may be facilitated by using this set of generic rules. The rule set should also facilitate the development of guidelines for effective selection of monitor points required by diagnostic assistants. These guidelines would be helpful in developing future hardware specifications.

(g) User interface - The color coded block diagrams used in REDEX to display monitor point status and diagnostic results have been well-received by system engineers and technicians. This approach has been adopted in another prototype diagnostic expert system for NASA ground equipment.

These observations and lessons learned will influence our continuing and future work.

7.0 FUTURE DIRECTIONS

The work on REDEX may be extended in a number of directions, as described below:

(a) Manual test points - REDEX currently renders a diagnosis based solely on the status of the internal monitor points and the current configuration of the RE. This data is usually sufficient to allow a fault to be diagnosed down to the circuit card level. In some cases, however, the fault can be isolated only down to a set of circuit boards. In these cases, manual observations and tests must be conducted to identify the faulty circuit board. REDEX may be extended to ask the technician for observation data and to recommend manual tests, and use this additional information to sharpen the diagnosis. Manual tests would be recommended considering their cost (in time required to perform), and the potential value of the test results in completing the diagnosis.

(b) Anomalous monitor data - REDEX currently assumes that status monitor data is correct. However, due to the possibility of mis-calibrated or faulty monitoring circuits, there are situations in which monitor data may be incorrect. REDEX may be extended to reason about anomalous monitoring situations, and develop alternate diagnoses based on various interpretations of the monitor data.

(c) Other tracking station equipment - REDEX may be extended to diagnose faults in other components of the RF Systems equipment at the tracking stations, such as the S-Band Exciters and Receivers. This direction would also motivate the extension of REDEX's reasoning with generic troubleshooting rules. REDEX currently uses generic rules to diagnose major portions of the RE, but needs specific rules to handle areas with unusual configurations, or where monitor points are very sparse. A goal would be to develop more powerful techniques for reasoning with generic rules, and thereby eliminate the need for specific rules. If this could be done for a class of equipment, then a diagnostic expert system "shell" could be built containing the full set of generic rules. A diagnostic assistant could be easily built for any equipment item in the class by just populating the knowledge base with a structural and functional description of the equipment item.

(d) Hypertext - The successful application of hypertext techniques in REDEX may be extended to other operator-driven information display functions at the tracking stations. This technique may be an effective approach for providing an integrated operator interface to the various information systems at the tracking stations.

It is felt that the current work on REDEX and these future directions are positive steps in the directions of increasing the levels of automation and performance while decreasing operations costs at the GN tracking stations.

ACKNOWLEDGEMENTS

The authors would like to thank Bikas Das for his assistance in defining the initial prototype, and Andy Rolinski for his review of project documentation. This work was done by Computer Sciences Corporation for Goddard Space Flight Center under contract NAS5-31500, Task Assignment 39-115.

REFERENCES

1. K. Howlin, J. Weissert, and K. Krantz, "MOORE: A Prototype Expert System for Diagnosing Spacecraft Problems," *1988 Goddard Conference on Space Applications of Artificial Intelligence*, May, 1988.

2. B. Cruse and C. Wende, "Expert System Support for HST Operations," *1987 Goddard Conference on Space Applications of Artificial Intelligence and Robotics*, May 1987.

3. J. Bush and S. Weaver, *Gamma Ray Observatory Backup Control Mode Analysis and Utility System (BCAUS) Prototype Requirements, Design, and Implementation*, Computer Sciences Corporation, CSC/TM-88/6069, September 1988.

4. L. G. Hull and P. M. Hughes, "CLEAR: Communications Link Expert Asssistance Resource," *1987 Goddard Conference on Space Applications of Artificial Intelligence and Robotics*, May 1987.

5. Kennedy Space Center, "Liquid-Oxygen Expert System," *NASA Tech Briefs Technical Support Package*, KSC-11332.

6. S. A. Mouneimne, "Mission Telemetry System Monitor: A Real-Time Knowledge-Based System," *1988 Goddard Conference on Space Applications of Artificial Intelligence*, May, 1988.

7. R. Dominy, "A Local Area Computer Network Expert System Framework," *1987 Goddard Conference on Space Applications of Artificial Intelligence and Robotics*, May 1987.

8. D. Lowe, et. al., "FIESTA: An Operational Decision Aid for Space Network Fault Isolation," *1987 Goddard Conference on Space Applications of Artificial Intelligence and Robotics*, May 1987.

9. W. Wilkinson, et. al., "Achieving Real-Time Performance in FIESTA," *1988 Goddard Conference on Space Applications of Artificial Intelligence*, May, 1988.

10. D. J. Zillig, "S-Band Ranging System for NASA Mission Support," *Proceedings of the 1987 International Telemetering Conference*, San Diego, CA, October 26-29, 1987.

11. J. Baumert, A. Critchfield, and K. Leavitt, "The Need for a Comprehensive Expert System Development Methodology," *1988 Goddard Conference on Space Applications of Artificial Intelligence*, May, 1988.

12. E. Luczak, *Diagnostic Expert System for the GN Ranging Equipment - Requirements Document*, Computer Sciences Corporation, July 1988.

13. E. Luczak, *Diagnostic Expert System for the GN Ranging Equipment - REDEX Design*, Computer Sciences Corporation, October 1988.

14. L. Gilstrap and J. Retter, *Expert System Development Methodology Users Guide*, Computer Sciences Corporation, CSC/TM-88/6146, September 1988.

15. L. Gilstrap and J. Retter, *Expert System Development Methodology Standard*, Computer Sciences Corporation, CSC/TM-88/6147, September 1988.

16. E. Luczak, *Diagnostic Expert System for the GN Ranging Equipment - Prototype Tool Evaluation Report*, Computer Sciences Corporation, September 1988.