N90-22306

990

AUTONOMOUS POWER EXPERT SYSTEM

Jerry L. Walters, Edward J. Petrik,
Mary Ellen Roth, and Long Van Truong
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135

and

Todd Quinn and Walter M. Krawczonek
Sverdrup Technology, Inc.
NASA Lewis Research Center Group
Cleveland, Ohio 44135

ABSTRACT

The Autonomous Power Expert (APEX) system has been designed to monitor and diagnose fault conditions that occur within the Space Station Freedom Electrical Power System (SSF/EPS) Testbed. The APEX system is being developed at the NASA Lewis Research Center by the Space Electronics Division (SED) in conjunction with the Space Station Directorate and Power Technology Division (PTD). APEX is designed to interface with SSF/EPS testbed power management controllers to provide enhanced autonomous operation and control capability.

The APEX architecture consists of three components: (1) a rule-based expert system, (2) a testbed data acquisition interface, and (3) a power scheduler interface. Fault detection, fault isolation, justification of probable causes, recommended actions, and incipient fault analysis are the main functions of the expert system component. The data acquisition component requests and receives pertinent parametric values from the EPS testbed and asserts the values into a knowledge base. Power load profile information is obtained from a remote scheduler through the power scheduler interface component.

This paper will discuss the current APEX design and development work. Operation and use of APEX by way of the user interface screens will also be covered.

INTRODUCTION

The APEX prototype system was designed as a high-level advisor for diagnosing faults in subsystems of the SSF/EPS testbed. A hierarchy of conventionally programmed controller computers reside between the symbolically programmed APEX system and the testbed subsystems (Wright, et al. (1989)). Prototype development work, for determining the design and requirements for APEX, was based on the Power Distribution Control Unit (PDCU) subsystem (Truong, et al. (1989)). APEX is currently interfaced to the PDCU subsystem controller and data communications has been established over a serial link. Ethernet communications is also available and future plans include obtaining parametric data values over the Ethernet remotely to development workstations and locally to a delivery workstation.

PRECEDING PAGE BLANK NOT FILMED

The APEX system consists of a rule based expert system and two interfaces that acquire data from the testbed and a remote power scheduler program. Domain knowledge from human experts has been acquired and coded into rules and stored in a knowledge base along with a model of the domain. Diagnostic rules for other SSF/EPS subsystems are to be added as Ethernet communication is established with other subsystem controllers.

Information required to diagnose faults is obtained through APEX testbed and scheduler interfaces. Parametric data values are requested from the PDCU subsystem controller by the testbed interface. Only pertinent parametric data values as determined by knowledge about expert troubleshooting techniques are requested from the PDCU subsystem controller.

Load profile information is read by the scheduler interface. Heuristics are applied to the load profile information to determine recommended actions when faults occur. Recommended actions are based on load profile information such as priorities of the loads, duration of each load, how much power the loads require for their durations, and the amount of available power from the sources subsystem.

Rule Based Expert System

Design of the expert system is based on a model which consists of objects organized into frames. This combination of objects and frames represents an integration of object oriented programming and frame based knowledge representation. The frames form a network which correspond to the PDCU subsystem. The frame representation of the subsystem contains connectivity information about the devices in the subsystem and how objects relate and have inheritance to other objects.

Fault identification is done in two phases: (1) fault detection and (2) fault isolation. Forward and backward chaining rules emulate expert reasoning necessary to detect and isolate faults. Fault detection monitors parametric values of the electrical power system to determine if the system is operating correctly. The parametric values are power, voltage, current, and status. The load profile from the remote power scheduler program contains information about expected operating conditions for each load. The APEX system determines expected analog values, for each PDCU analog test point, based on each loads scheduled operating condition. The expected test point values are compared to measured parametric values from the testbed to detect faults.

Three different types of faults that can be detected: (1) inconsistent, (2) active, and (3) incipient. Inconsistency faults occur when two or more data values give conflicting information. Active faults are detected when measured values are higher or lower than the expected values within a defined tolerance. Single and multiple active faults can be detected. Incipient faults are detected by monitoring a history of data values that identify trends toward tolerance limits. Trends are detected by statistical inference based on correlation and regression analysis of historical data. All faults are detected by forward chaining inference. Once a fault is detected, domain specific troubleshooting knowledge is referred to and backward chaining is initiated to isolate faults.

Probable causes are identified by the fault isolation phase. Rules based on knowledge acquired from domain experts are categorized by frames and associated with classes of faults. Backward chaining is initiated on the appropriate frame(s) of rules to identify probable causes. Organizing the rules with frames prevents unnecessary chaining on inappropriate rules. In some cases, more than one probable cause is displayed. When more than one probable cause is displayed the causes are shown to the operator in the order of highest to lowest probability. Justification is available to the operator to explain the reasoning process for each probable cause. Justification is obtained from the expert system from a trace back of the backward chaining rule firing. The trace back retrieves the premises of each rule that fired during backward chaining. Functions written in Lisp, translate the rule premises written in an expert system shell language, into English. The English is then displayed as a natural language explanation of the reasoning process leading to probable cause conclusions.

A recommended action feature suggests what should be done to correct the fault. The APEX system considers information such as the severity of the fault and priority of the loads in recommending the action that should be taken to correct, bypass or temporarily tolerate the fault.

Hardware and software being used for the development of APEX are Texas Instruments Explorer II LX workstations, the Knowledge Engineering Environment (KEE) expert system development shell (KEE User's Guide, 1989) and common Lisp (List Processing Language).

Testbed Data Acquisition and Scheduler Interfaces

The testbed data acquisition interface requests pertinent parametric data values from the PDCU controller and asserts new values received into the knowledge base. For incipient fault detection, the data acquisition interface stores the values in a First In First Out (FIFO) table that contains the last 200 values for each analog test point on the testbed. The scheduler writes the load profile to shared memory. A handshaking protocol indicates when the shared memory has been updated with new information. Upon sensing the update, the scheduler interface reads the load profile from memory and updates the knowledge base. Forward chaining fault detection is initiated whenever new values are received in the knowledge base.

User Interface

The APEX system is fully mouse activated for quick and easy operation. A combination of KEE active images and Lisp functions have been developed to provide user graphic screens that display information and menu pick options to the operator. The graphic screens also provide a verification method to assure the system is reasoning correctly. For verification, domain experts set up fault scenarios and review the expert systems diagnosis of the faults and recommended actions.

The operator reviews justification and recommended actions and performs recovery procedures to clear or bypass faults. A longer term goal is to communicate recommended actions as messages to subsystem controllers. The purpose of communicating messages to the subsystem controllers would be to initiate automatic fault correction. Currently, the operator is kept in the fault detection, isolation, and recovery loop as a measure of validation.

The main user interface screens provide three levels of access to the system. The three levels of access are: (1) a top level block diagram of the SSF/EPS testbed, (2) block diagrams of each subsystem, and (3) subsystem schematic diagrams. Each screen is mouse sensitive for displaying other screens. Visual flashing indications appear on areas of the displays when faults occur. In addition, the schematic display shows the latest voltage, power, phase angle, current, and status values at each device test point. Figures 1, 2, and 3, respectively, show the screens corresponding to the three levels of access.

Three other screens show explanations of fault detection, isolation, and justification. Examples of these three screens are shown in figures 4 to 6.

An example of a recommended action display is shown in figure 7.

There are two screens that correspond to the testbed and scheduler interfaces. An example of the scheduler interface screen appears in figure 8.

There are three screens to display graphical plots of incipient fault data. An example of a ratio plot of measured to expect values for one of the current test points is shown in figure 9. The other two plot types are tolerance and history.

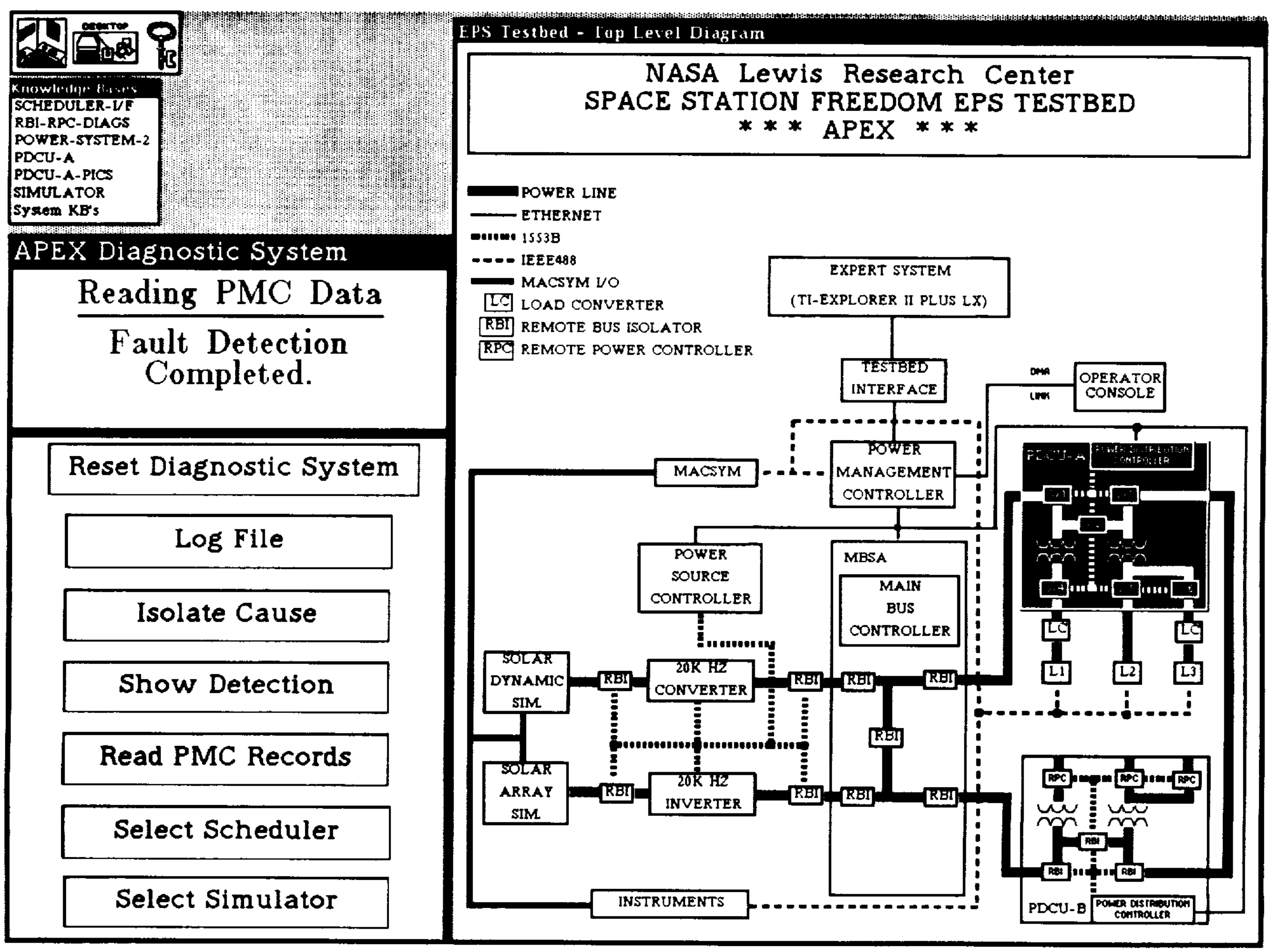

FIGURE 1. - TOP LEVEL BLOCK DIAGRAM.

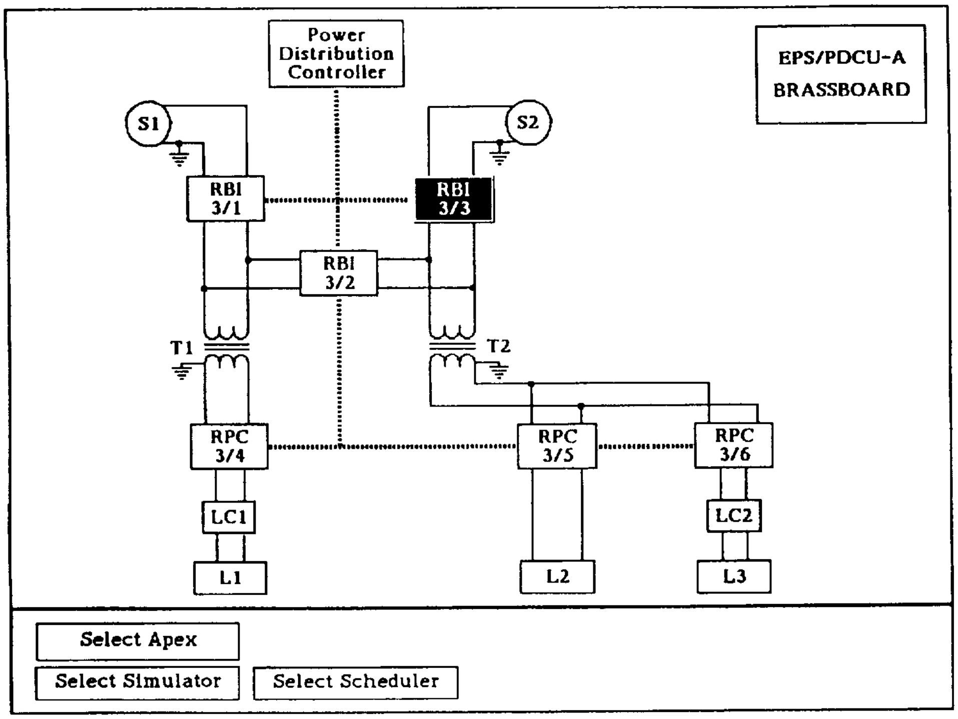

FIGURE 2. - PDCU SUBSYSTEM BLOCK DIAGRAM.

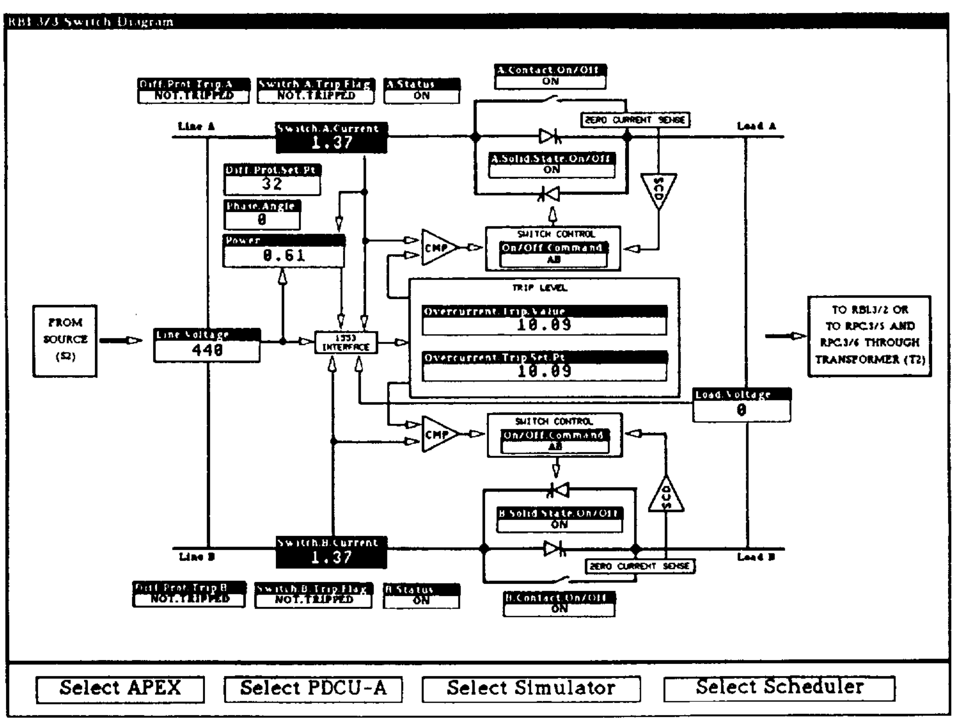

FIGURE 3. - SCHEMATIC DIAGRAM.

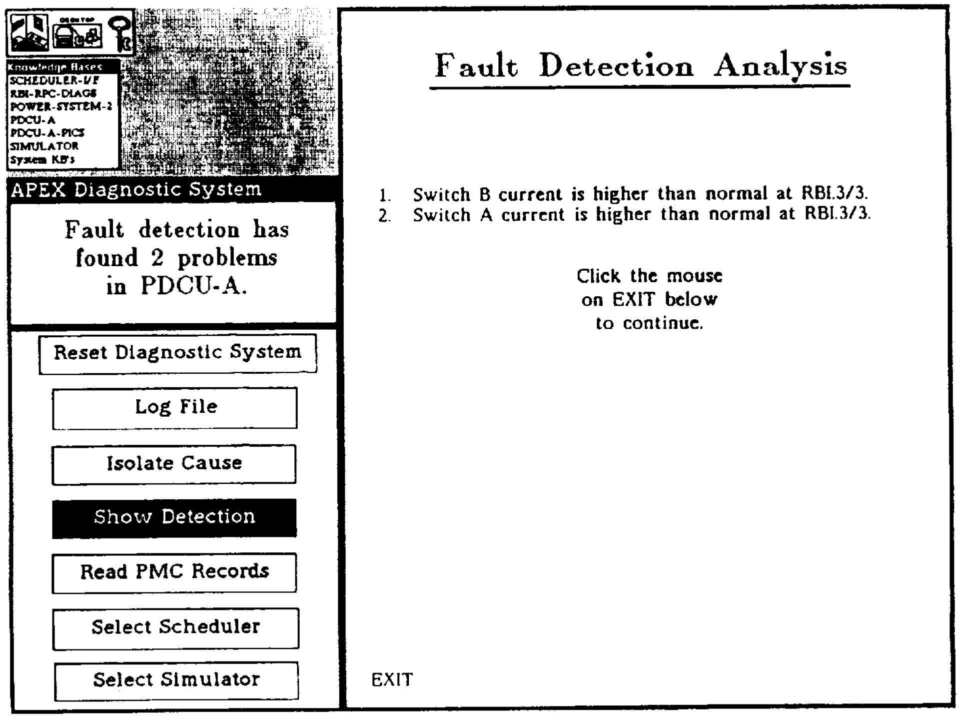

FIGURE 4. - FAULT DETECTION ANALYSIS SCREEN.

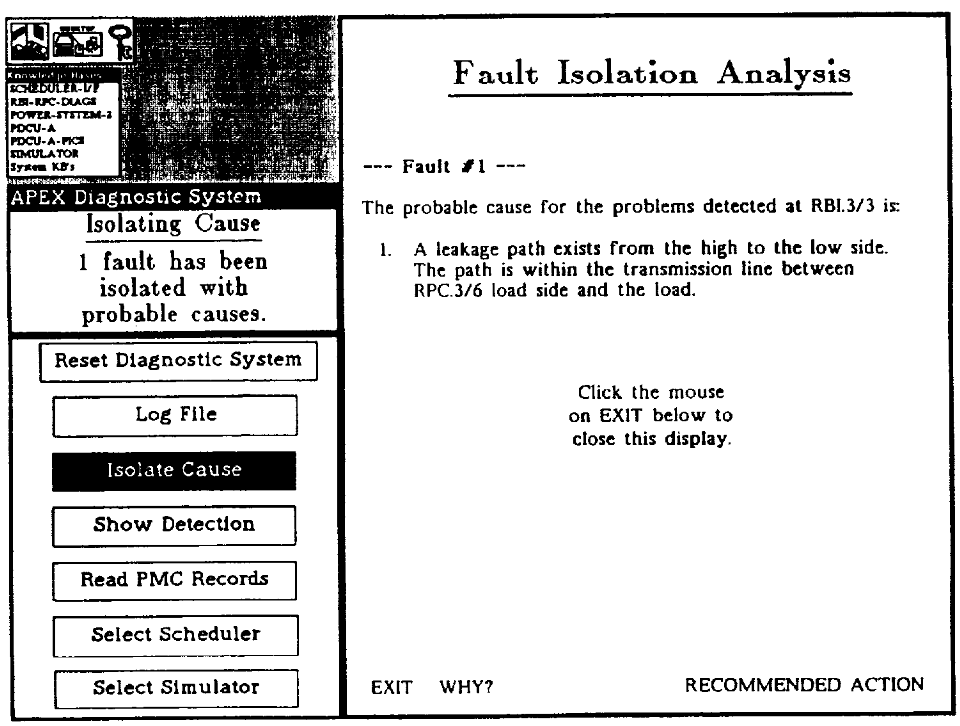

FIGURE 5. - FAULT ISOLATION ANALYSIS SCREEN.

ORIGINAL PAGE IS OF POOR QUALITY

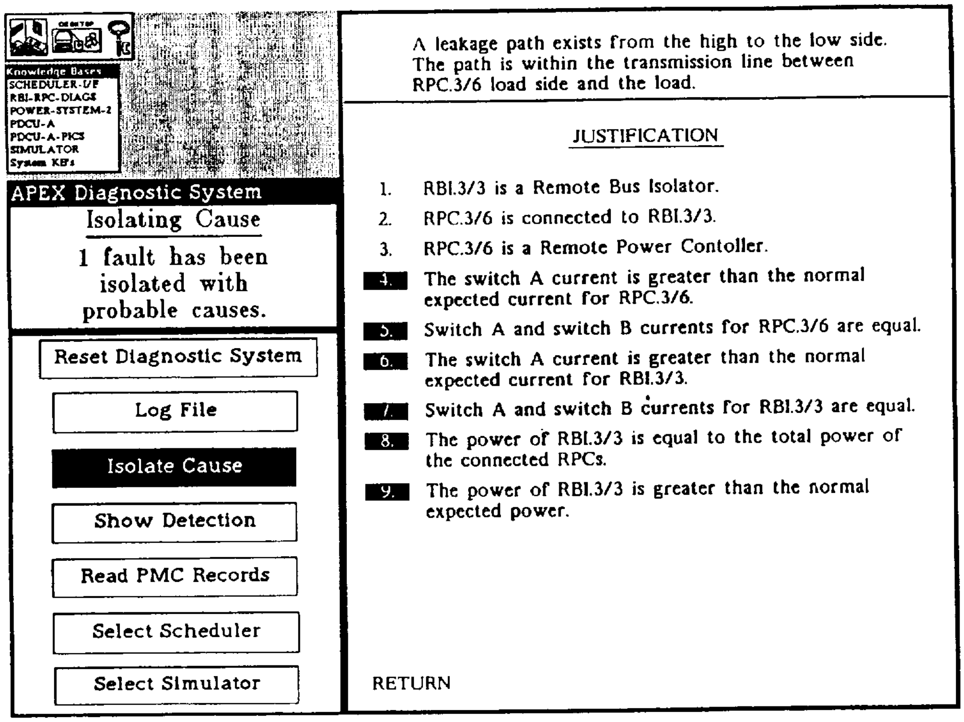

FIGURE 6. - FAULT JUSTIFICATION SCREEN.

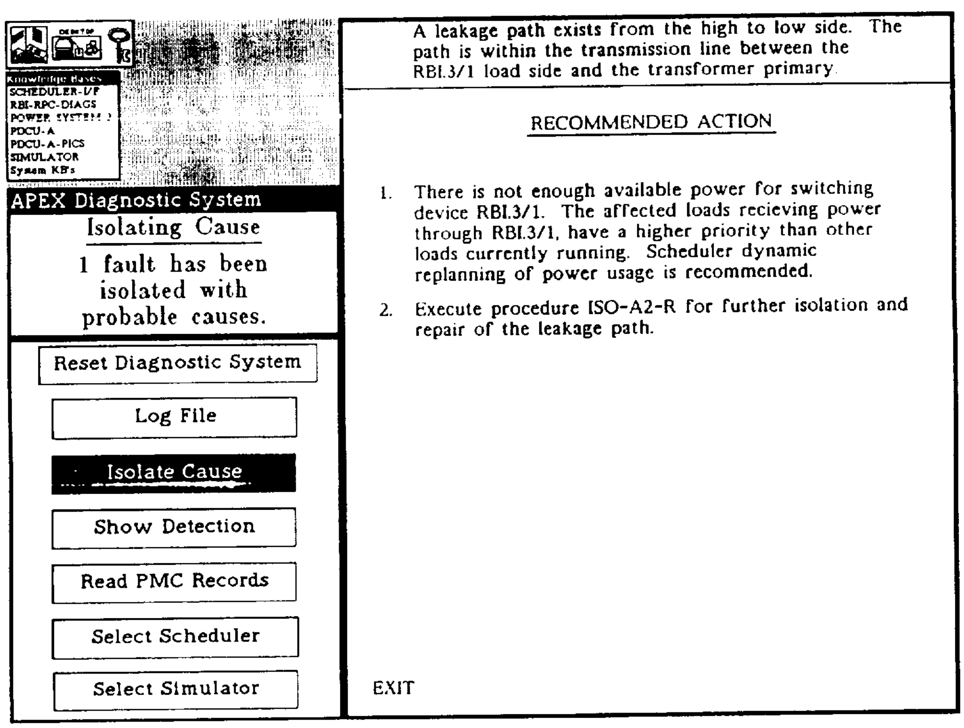

FIGURE 7. - RECOMMENDED ACTION DISPLAY.

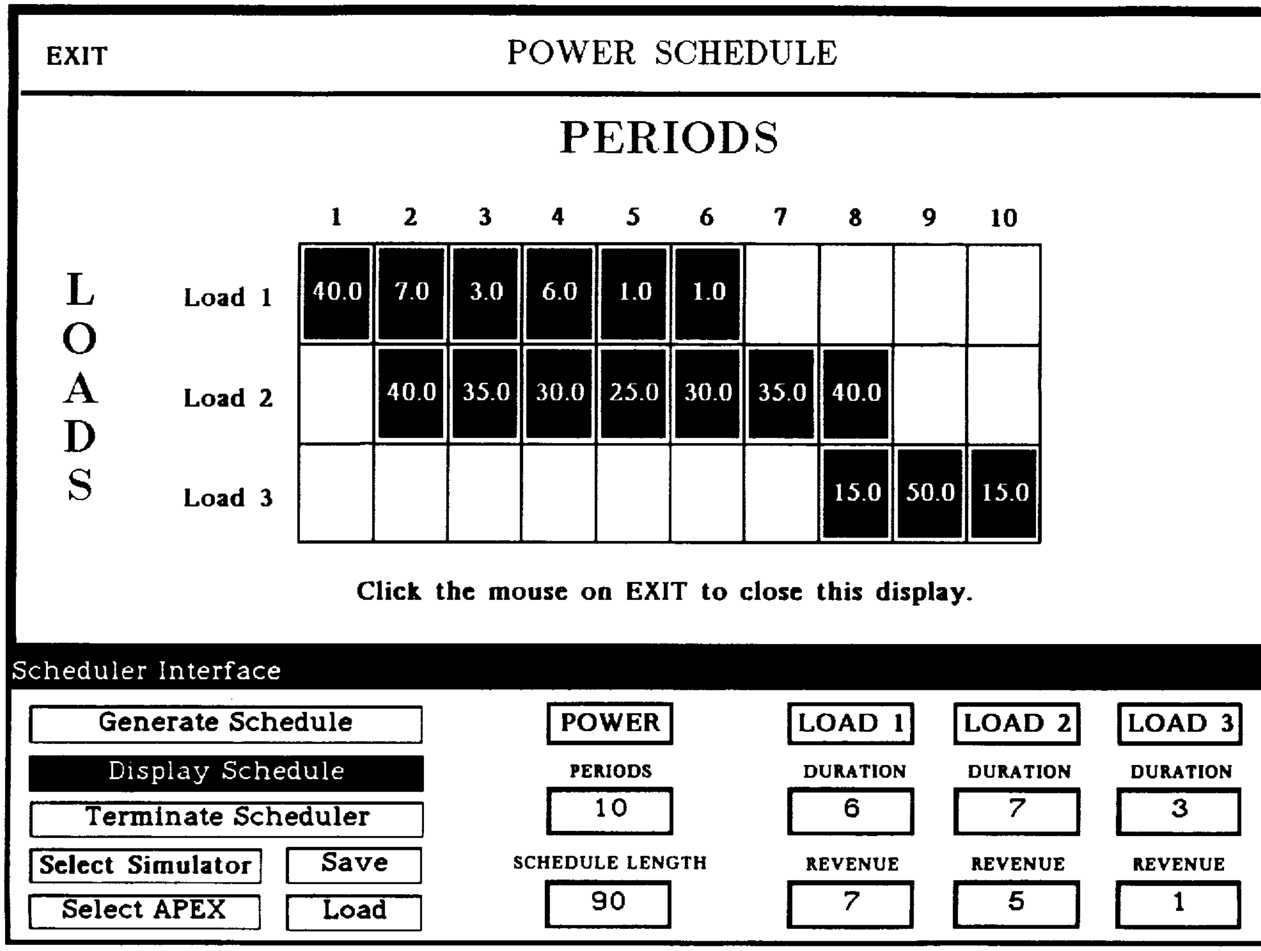

FIGURE 8. - SCHEDULER INTERFACE SCREEN.

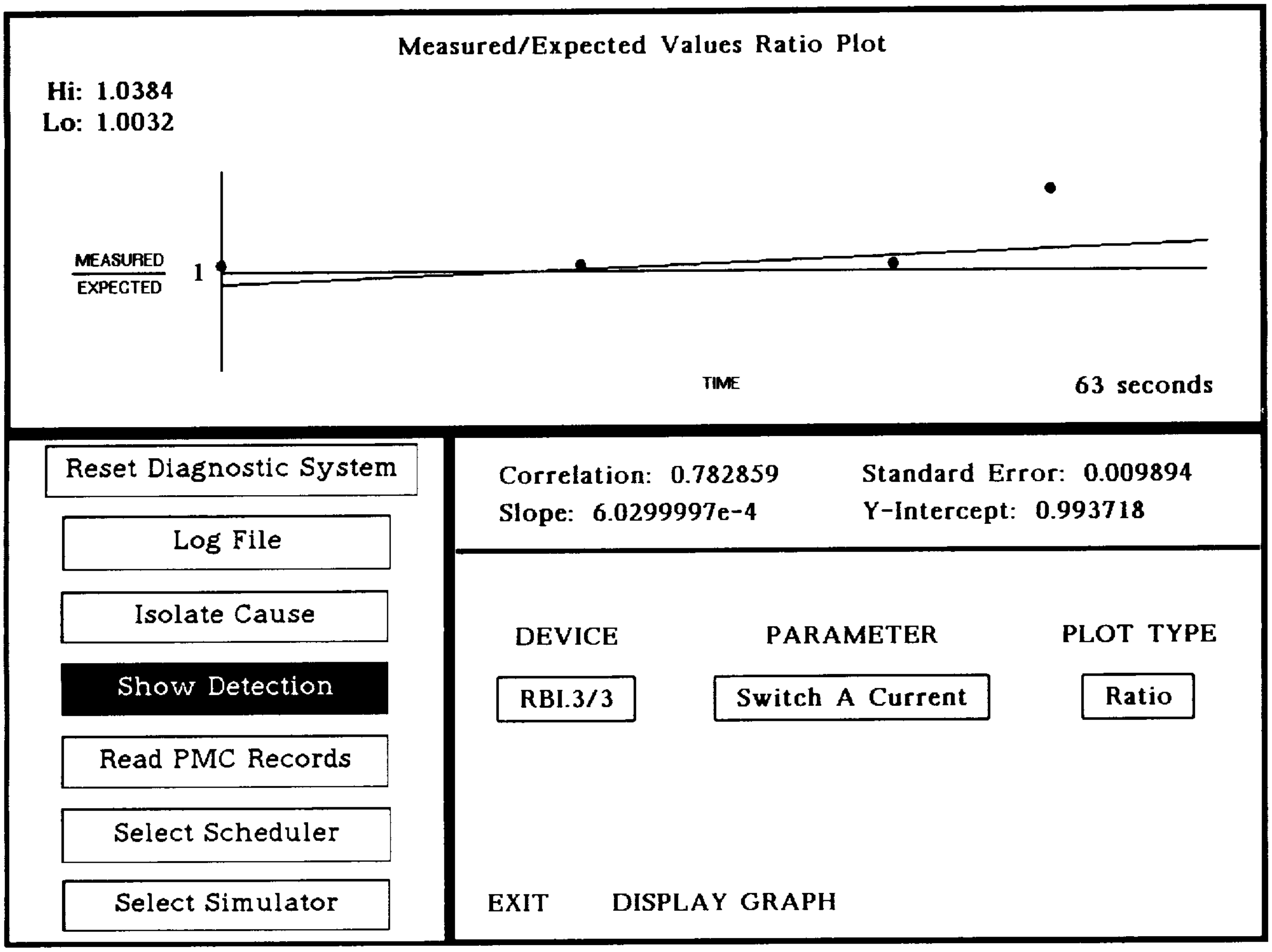

FIGURE 9. - INCIPIENT FAULT RATIO PLOT.

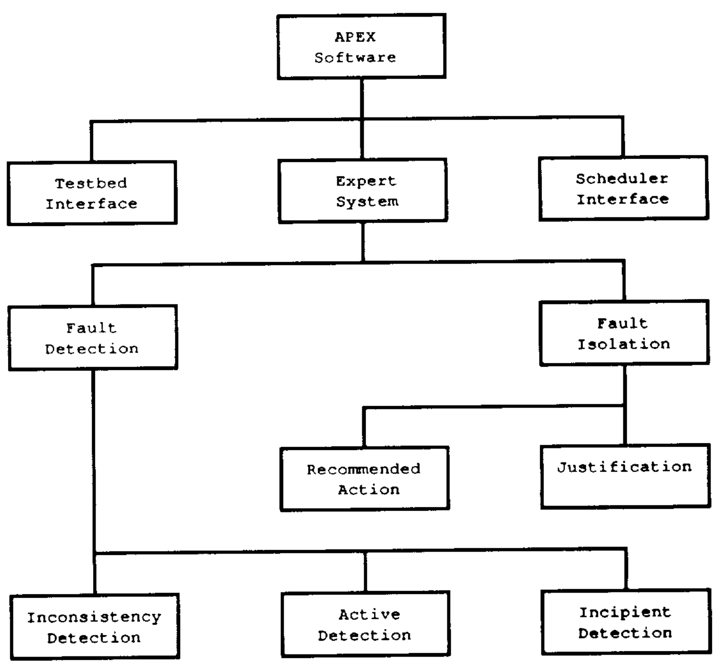

FIGURE 10. - APEX SYSTEM ARCHITECTURE.

SUMMARY AND CONCLUSIONS

APEX has been designed to emulate expert diagnostic fault detection, isolation, and recovery methods. Figure 10 shows the APEX system architecture.

The expert system has fault detection and fault isolation phases. Data values are monitored at each power system test point and compared to expected values derived from a remote power scheduler to detect faults. The testbed interface acquires parametric data values from the testbed. New data values in the knowledge base drive forward chaining for fault detection. Areas of fault detection include inconsistency checks, monitoring for single and multiple active faults and incipient fault analysis. Fault isolation includes justification of probable causes and recommended actions to clear faulty conditions.

The expert system can check more test points, more often than a human operator can, and do so without fatigue. Expert knowledge is continuously available to monitor and diagnose faults in the power system and appropriate recovery procedures are instantly displayed and available to lower level controllers that can command and control the power system. These are valuable benefits for a system such as a space station that will require continuous, long-term health monitoring and autonomous control. Much of the burden placed on human operators can be relieved with the type of expert system technology build into the APEX system.

ACKNOWLEDGMENTS

We would like to acknowledge Eric Bobinsky, Allen Richard, Mark Ringer, Amy Edelman, and Silvia Washington for their work in developing power scheduler software; Mark Ringer for his work in developing the hardware interface between the APEX system and the PDCU subsystem controller; James Dolce and Pamela Mellor for their support, and James Kish, our project manager, for guidance.

REFERENCES

Wright, T; Mackin, M.; and Gantose D.: Development of Ada Language Control Software for the NASA Power Management and Distribution Testbed. NASA Lewis Research Center, Cleveland, OH, 1989.

Truong, L., et al.: Autonomous Power Expert Fault Diagnostic System for Space Station Freedom Electrical Power System Testbed. Third Annual Workshop on Space Operations Automation and Robotics (SOAR 1989), Lyndon B. Johnson Space Center, Gilruth Recreation Center, Houston, TX, July 25-27, 1989 (NASA CP-   , to be published).

KEE User's Guide, Version 3.1, Intellicorp, Inc., Mountain View, CA, May 1989.