MIS30G

A HIERARCHICALLY DISTRIBUTED ARCHITECTURE FOR FAULT ISOLATION EXPERT SYSTEMS ON THE SPACE STATION+

STEVE MIKSELL
SUE COFFER*
STANFORD TELECOMMUNICATIONS, INC.

Abstract

The SAFTIES (Space Station Axiomatic Fault Isolating Expert Systems) System deals with the hierarchical distribution of control and knowledge among independent expert systems doing fault isolation and scheduling of Space Station subsystems. On its lower level, fault isolation is performed on individual subsystems. These fault isolation expert systems contain knowledge about the performance requirements of their particular subsystem and corrective procedures which may be involved in response to certain performance errors. They can control the functions of equipment in their system and coordinate system task schedules. On a higher level, the Executive contains knowledge of all resources, task schedules for all subsystems, and the relative priority of all resources and tasks. The executive can override any subsystem task schedule in order to resolve use conflicts or resolve errors that require resources from multiple subsystems. Interprocessor communication is implemented using the SAFTIES Communications Interface (SCI). SCI is an application layer protocol which supports the SAFTIES distributed multi-level architecture.

1.0 INTRODUCTION

The SAFTIES project included the following goals:

- Design a hierarchical system where one expert system (executive) coordinates many expert systems to isolate faults and do related processing.

+ Work supported under NASA contract NAS5-29280.

* Sue Cofer participated in this project while employed by STI. She is currently associated with Digital Equipment Corp.

- Define a communications protocol (applications layer) to support expert system interaction
- Build a demonstration system which will form the basis for a test bed for further definition and evaluation of distributed expert system capabilities

This paper provides an overview of the demonstration system which was developed to provide a proof-of-concept testbed for achievability of these goals.

2.0 THE DISTRIBUTED HIERARCHICAL CONFIGURATION

Many of the feasibility questions associated with distributed processing and hierarchical control can be most readily addressed in a testbed configuration. Fault isolation was selected as the base-level element in the hierarchy. In order to develop requirements which would support the development of a meaningful test bed demonstration, it was necessary to identify specific areas where the fault isolation would be performed. Selection was influenced by an awareness of the potential long-term space-based applications. The general domain areas chosen for this processing were:

1) Robotics system (FIRES)
2) Communications system (FIESTA)
3) Environmental factors reporting system (FISHES)

To support project objectives, the architecture for a Space station Axiomatic FaulT Isolating Expert System (SAFTIES) (shown in Figure 1) was developed.

As the figure indicates, two levels of control are present. Within the lower level, the separate expert systems are operating on their defined domains of responsibility. Communication between levels is provided by the SCI (SAFTIES Communications Interface). The DMS (Data Management System) which would support the SCI is presented symbolically in the overview.

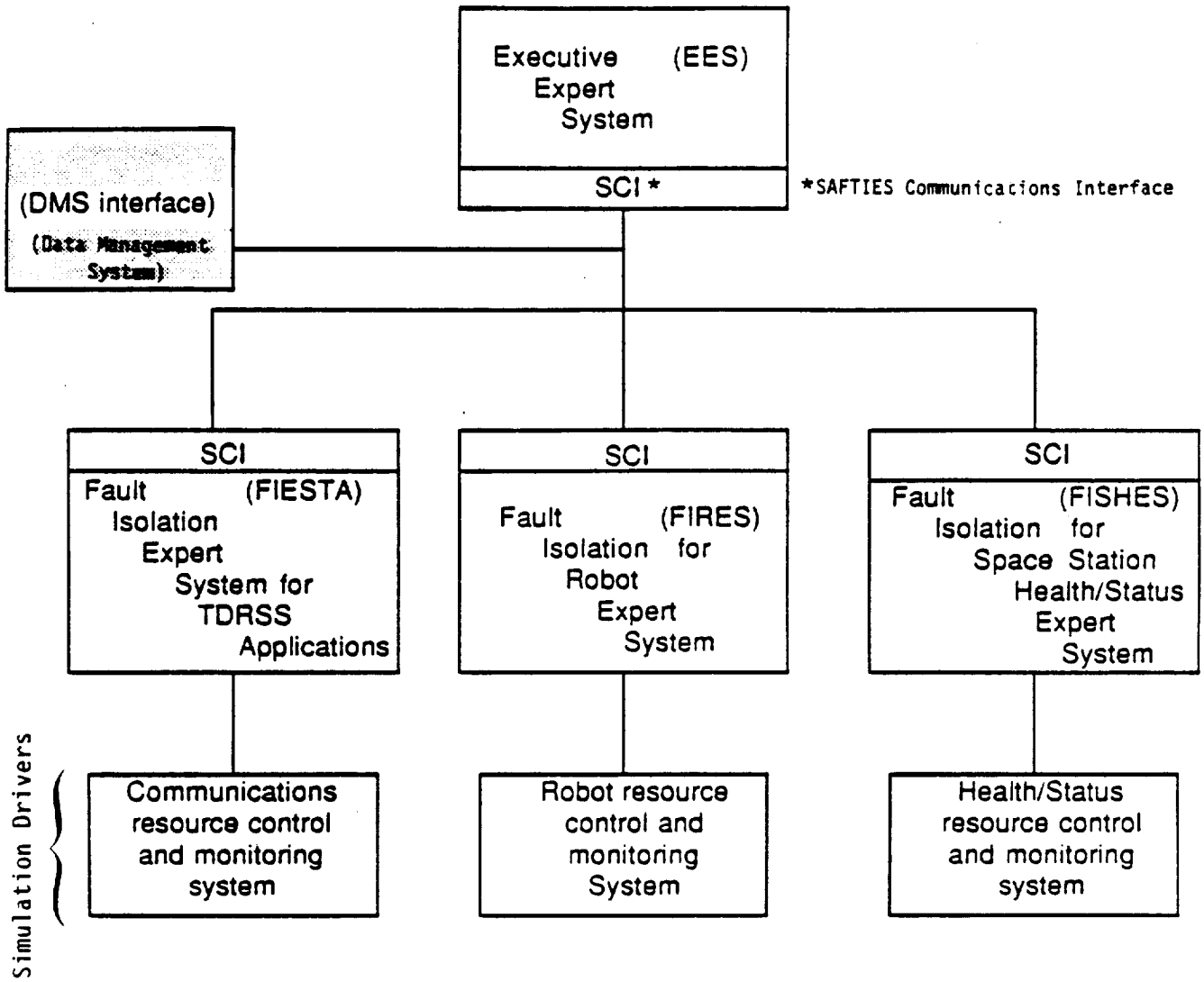

FIGURE 1: SAFTIES SYSTEM OVERVIEW

On the upper level, the executive function provides conflict resolution between the various subsystems as well as allowing for the optimal satisfaction of mission objectives by enforcing a prioritization of subsystem objectives.

A hardware configuration which would provide a distributed environment was identified. Functional allocation over the hardware configuration was performed with establishment of hierarchical control as a major driver. This resulted in the functional allocation shown in Figure 2.

3.0 THE SAFETIES COMMUNICATION INTERFACE (SCI)

This implementation concerned itself primarily with the application layer of the ISO Basic Reference Model of Open Systems Interconnection. A summary of the desired capabilities are given in Table 1. Table 2 gives a list of the different kinds of message types used in the SAFTIES demonstration. There are many other types of messages that could be used in a full-fledged distributed system. The goal of the SCI protocol is to allow expert systems to "plug in" to SAFTIES with minimal alterations.

The Art Implementation

The lower level expert systems, running in ART on the Symbolics 3640, receive SCI data as follows: between every rule that fires, the serial line is read to see if data is present; if so, it reads the data and asserts it as a fact; if not, processing continues as normal.

The Pascal Implementation

The Executive, running in Pascal on the Sperry IT/PC, receives SCI data as follows: whenever the screen is idle (waiting for input from the

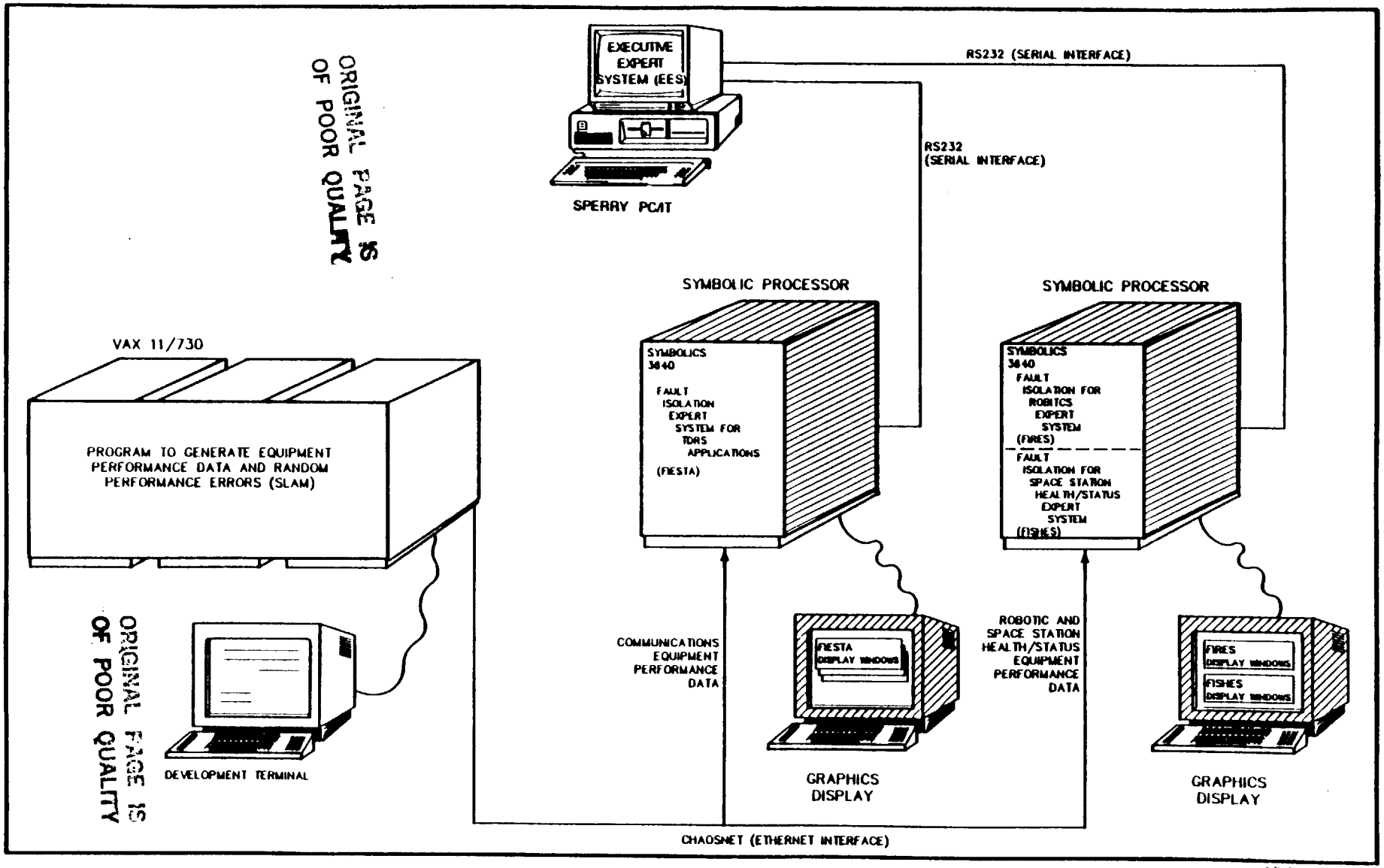

FIGURE 2: FUNCTIONAL OVERVIEW OF SAFTIES

TABLE 1

SUMMARY OF SAFTIES COMMUNICATIONS INTERFACE (SCI)

- A COMMUNICATIONS PROTOCOL THAT ALLOWS EXPERT SYSTEMS TO COMMUNICATE WITH THE EXECUTIVE
- ALL SCI MESSAGES HAVE A HEADER IN THE FOLLOWING FORMAT:
  - PRIORITY
  - MESSAGE TYPE CODE
  - SENDING SYSTEM
  - RECEIVING SYSTEM
  - ACTIVITY/EVENT CODE
  - SUB-CODE (RESOURCE ID, REQUESTED DATA PARAMETER ID)
- FOLLOWING THE HEADER IS A PARAMETER LIST GIVING INFORMATION SPECIFIC TO THE MESSAGE.

TABLE 2

SCI MESSAGE TYPES

MESSAGE TYPE	DEFINITION	PARAMETERS
101	REQUEST FOR DATA	NONE
102	SCHEDULES	BEGIN TIME, [END TIME], [REPETITIONS]
103	COMMAND	COMMAND CODE, COMMAND PARAMETERS
201	PERFORMANCE DATA	VALUE
202	CONFIGURATION DATA	CONFIGURATION LIST
203	FAULT ISOLATION DATA	CURRENT VALUE, ESTIMATED CAUSE OF FAULT
204	EMERGENCY ALARM	CURRENT VALUE, [ESTIMATED CAUSE OF FAULT]
205	EVENT STATUS	EVENT CODE [IN SERVICE, PRESERVICE, ETC.]
900	MESSAGE IN ERROR	[RETURN ENTIRE MESSAGE IN ERROR]

user) and before the keyboard is checked to see if an input is received, the serial "com" ports 1 and 2 are checked to see if data is present; if so, it reads it. This is done on the DOS BIOS level using interrupts and status registers. If it reads the serial line and data is present, it analyzes the data read to make sure an entire SCI record is read; if not, it issues consecutive reads until the entire SCI record is received. If no data is present on either serial line, it checks the keyboard for activity.

4.0 TESTBED CONFIGURATION COMPONENTS

This section presents each of the testbed configuration components. In Section 4.1 the simulation approach to data generation is described. The three independent expert systems are presented in Section 4.2. Section 4.3 then describes the executive expert system.

4.1 SIMULATION DRIVERS

Drivers are used to generate performance data and simulate faults. The simulation driver for each fault isolation expert system is written in SLAM (Simulation Language for Alternative Modeling); a summary of the drivers is given in Table 3. There are separate drivers for each expert system. The variety and quantity of equipment for which data is generated within each simulator is limited only by physical resources of the host hardware. A nominal value, high and low tolerance values, equipment code, and station (location of the equipment) are assigned to each data generating equipment. Every 5 seconds a "reading" is taken; under normal circumstances, the "reading" is a random number with a triangular distribution (mean as the normal value, normal tolerance as low value, high tolerance as high value). At random time periods, however, these values will fall out of range. How the numbers fall out of range is determined by the type of error.

TABLE 3

SUMMARY OF SIMULATION DRIVERS

- WRITTEN IN SLAM (SIMULATION LANGUAGE FOR ALTERNATIVE MODELING)
- A NOMINAL VALUE, HIGH TOLERANCE AND LOW TOLERANCE IS INITIALLY ASSIGNED TO EACH RESOURCE
- EVERY 5 SECONDS DATA IS GENERATED FROM ALL RESOURCES SUPPORTING SERVICES AND EVENTS; FOR EACH RESOURCE, IT IS A RANDOM NUMBER TRIANGULARLY DISTRIBUTED BASED ON THAT RESOURCE'S NOMINAL, HIGH TOLERANCE AND LOW TOLERANCE VALUES
- ERRORS INVOLVING RESOURCE PERFORMANCE ARE GENERATED BOTH RANDOMLY AND AT PRESET TIMES; WHEN AN ERROR HAS OCCURRED, THAT RESOURCE VALUE WILL:
  - EXCEED HIGH TOLERANCE
    - -- SLOWLY
    - -- QUICKLY
    - -- IMMEDIATELY
    - -- DELAYED
  - EXCEED LOW TOLERANCE
    - -- SLOWLY
    - -- QUICKLY
    - -- IMMEDIATELY
    - -- DELAYED
  - START AND REMAIN AT AN OUT-OF-RANGE VALUE

ONCE IN ERROR, THE ERROR REMAINS FOR A RANDOM TIME PERIOD.

4.2 THE INDEPENDENT EXPERT SYSTEMS

In this section the expert system nodes which occur in the SAFTIES hierarchy are presented. These consist of the three base level expert systems, FIRES (robotics), FISHES (environment), FIESTA (communications) and the top level EXECUTIVE EXPERT SYSTEM (EES).

4.2.1 FIRES (Fault Isolation for Robot Expert System)

4.2.1.1 The FIRES System. An overview of FIRES requirements is presented in Table 4. The FIRES expert system node receives performance/status data from a robotics system. For purposes of the testbed prototyping a Scara-type robot arm (4 degrees of freedom) was selected for simulation. Readings on various aspects of the arm operations are reported to FIRES every five seconds. All readings by FIRES are assumed to have a nominal value and an accepted range of operation as shown in Table 5.

The FIRES system employs a data-driven approach to diagnosis. When performance values which fall outside the acceptable range are encountered the relevant detection rules are activated. The detection rules then report the anomaly and summarize the information in a manner to support further diagnostic processing.

For the FIRES system, Fault Diagnosis has been implemented using a pattern-matching technique in a production system paradigm. This implementation illustrates the technique of inferring fault causes directly from specific patterns of anomalies.

For the demonstration prototype, fault isolation was followed by sending notification to the executive.

FIRES is implemented on a SYMBOLICS 3640 using the Automated Reasoning Tool (ART) from INFERENCE Corporation. ART is an Expert System (ES)

TABLE 4

FIRES REQUIREMENTS SUMMARY

FAULT ISOLATION FOR ROBOTICS
EXPERT SYSTEMS (FIRES)

- FUNCTIONS
  - ISOLATE FAULTS IN THE ROBOT OPERATING SYSTEM
  - INFORM EXECUTIVE OF MISSION CRITICAL FAULTS (ALARMS)
- INPUTS
  - EVENT SCHEDULE (CONTAINING ROBOT RESOURCE CONFIGURATION DATA)
  - COMMAND (FROM EXECUTIVE)
  - PERFORMANCE DATA (FROM ROBOT RESOURCES)
    - -- JOINT POSITION AND/OR TORQUE
    - -- DRIVE TORQUE
    - -- GRIPPER (TACTILE SENSOR, POSITION, FORCE)
    - -- SENSOR DATA (VISION, FIBER OPTICS, THERMAL, PRESSURE, STRAIN GAUGE)
- OUTPUTS
  - FI DIAGNOSTIC DATA
  - EVENT STATUS, AS REQUESTED
  - RESOURCE PERFORMANCE AND CONFIGURATION DATA, AS REQUESTED
- HARDWARE
  - SYMBOLICS 3640 (WITH OPTION TO MOVE TO MICROVAX II AI STATION)
  - COAXIAL AND SERIAL CABLES
- COMMUNICATIONS
  - RS232 SERIAL INTERFACE TO SPERRY IT
  - ETHERNET (CHAOSNET) CONNECTION TO VAX
- SOFTWARE
  - ART
  - LISP

TABLE 5

FIRES MONITORING PARAMETERS
(NOMINAL VALUES AND ACCEPTABLE RANGES)

	NOMINAL VALUE	LOW TOLERANCE	HIGH TOLERANCE	EQUIPMENT CODE	STATION/ LOCATION CODE
ROBOT-BATTERY	18.0	12.0	24.0	E01	01
SERVO TORQUE	10.0	0.0	20.0	E02	01
ARM VELOCITY	0.0	-18.0	18.0	E03	02
GRIP-FORCE	4.0	3.0	5.0	E04	03
GRIP-SENSE	1.0	0.0	1.0	E05	03

Development Tool which supports rapid prototyping. By providing basic ES constructs, it allows expert system development to concentrate on the knowledge engineering aspects of the task.

4.2.1.2 FIRES Simulation Driver. The FIRES driver simulates 5 performance monitoring pieces of equipment, obtaining readings of battery power, servo motor torque, arm velocity, gripper force and gripper presence sensor.

There are 6 possible errors which may occur:

1) Arm locked in place (collision with obstacle)
2) Incorrect gripper (possibly incorrect configuration in schedule)
3) Dropped object
4) Arm broken (collision with obstacle (wall) at high velocity)
5) Arm payload exceeded (picking up an object too heavy for the robot)
6) Battery low on power

These errors occur randomly, approximately one error per minute. At the end of the error, which lasts approximately 30 seconds, the readings return to acceptable range, indicating a correction has occurred.

4.2.2 FISHES (Fault Isolation for Space Station Health/Status Expert System)

4.2.2.1 The FISHES System. The FISHES expert system node receives performance data which is associated with health and status for an assigned area of the spacecraft. Requirements and monitored parameters similar to those presented for the FIRES system were developed for FISHES.

The data-driven approach employed in FISHES was also applied for detection of anomalies in the FIRES domain. The same form of pattern-

matching strategy employed in FISHES was adopted for FIRES. This logical replication allowed for the timely availability of the FIRES node during development and implementation of the testbed.

There was also physical replication since FISHES has also been implemented on a SYMBOLICS 3640 using ART.

4.2.2.2 FISHES Simulation Driver. The FISHES driver simulates 6 performance monitoring pieces of equipment, obtaining readings of cabin temperature, cabin pressure, percent oxygen, percent particulates, and the spacecraft power consumption and power generation.

There are four possible errors which can occur:

1) Hole in spacecraft (broken seal, puncture due to poor docking or meteor, etc.)
2) Fire in cabin
3) Faulty air filter
4) Faulty pressure reading (indicating a malfunction in the monitoring equipment itself)

These errors occur randomly, approximately one every minute, and last for approximately 30 seconds. When the error is completed, the readings return to within the acceptable range.

4.2.3 FIESTA (Fault Isolation Expert System for TDRSS Applications)

4.2.3.1 The FIESTA System. The specific requirements which are relevant to the SAFTIES project are summarized in Table 6.

FIESTA is an evolving prototype expert system which has been developed under the auspices of NASA/GSFC Code 532-1.* FIESTA is designed to isolate faults in a communication network. The requirements guiding the

* Work was performed under the direction of Bendix Field Engineering Corporation, NASA contract NAS5-27600.

TABLE 6

FIESTA REQUIREMENTS SUMMARY

(SAFTIES RELATED)

FAULT ISOLATION EXPERT SYSTEM FOR
TDRSS APPLICATIONS (FIESTA)

- FUNCTIONS
  - RECEIVE AND MONITOR SPACE NETWORK STATUS INFORMATION
  - ISOLATE FAULTS WHICH OCCUR ON THE NETWORK
  - INFORM EES OF ISOLATED FAULTS
- INPUTS
  - STATUS INFORMATION IN THE FORM OF NASA'S CURRENT HIGH SPEED MESSAGES
- OUTPUTS
  - HIGH LEVEL FAULT ISOLATION DIAGNOSTIC DATA
  - ALARMS
- HARDWARE
  - SYMBOLICS 3640
- COMMUNICATIONS
  - RS232 SERIAL INTERFACE TO SPERRY IT
  - ETHERNET (CHAOSNET) CONNECTION TO VAX
- SOFTWARE
  - ART
  - LISP

FIESTA development are contained in "FIESTA Project Development Folder/ Volume II: Prototype Requirements Specification" (STI/E-25190A, 17 December 1985). These requirements define FIESTA operating as a standalone testbed. FIESTA employs a highly developed Axiomatic/ Hypothetical approach to fault isolation. Inclusion of FIESTA within the distributed hierarchy allows an extensive demonstration of this methodology.

FIESTA was incorporated in the distributed hierarchy via the SCI interface described earlier. This served to validate the expansion capabilities afforded by the SCI protocol.

4.2.3.2 FIESTA Simulation Driver. The FIESTA driver simulates 12 performance monitoring pieces of equipment. Errors in any service (KSAR, SSAR or SSAF) occur as follows:

- The signal strength starts to degrade
- When the signal strength falls below 3, then the locks go to Ø
- The number of frames in lock starts to fall
- When the frames in lock equal Ø, then the data present goes to Ø.

Again, the errors occur randomly for a random amount of time. At the end of the error, the readings return to acceptable ranges.

4.3 THE EXECUTIVE EXPERT SYSTEM (EES)

An overview of the requirements established for the Executive Expert System (EES) are presented in Table 7.

4.3.1 The Domain of EES

EES receives high-level fault isolation data from lower-level expert systems. In addition, EES can request operational parameters as needed. All data received is in the SCI format. Upon receipt of notification of a fault, the Executive determines a corrective action

TABLE 7

EXECUTIVE EXPERT SYSTEM (EES)
REQUIREMENTS SUMMARY

- DESCRIPTION
  - EXPERT SYSTEM SHELL DESIGNED TO AID ASTRONAUT/USER IN SCHEDULING, COORDINATING, AND CONTROLLING OF DISTRIBUTED EXPERT SYSTEMS IN THE SPACE STATION
- FUNCTIONS
  - PROVIDE HUMAN INTERFACE/WORKSTATION TO
    - -- RETRIEVE INFORMATION FROM EXPERT SYSTEMS
    - -- COMMAND
    - -- EFFECT FAULT ISOLATION PROCESSING
  - RECEIVE FAULT ISOLATION DIAGNOSIS FROM DISTRIBUTED FAULT ISOLATION EXPERT SYSTEMS AND SCHEDULE CORRECTIVE EVENT(S)
  - ALLOCATE ASSIGNED RESOURCES, RESOLVING CONFLICTS AS THEY ARISE
  - SCHEDULE EVENTS BASED ON PRIORITIES, NEED, AND REQUESTS
- INPUTS
  - SPACE STATION EVENT SCHEDULES (FROM NASA OR SCHEDULER EXPERT SYSTEM)
  - HIGH LEVEL PERFORMANCE DATA
  - FAULT ISOLATION DIAGNOSTIC DATA
  - EVENT STATUS
  - HUMAN EXPERT INPUT AND OVERRIDES
- OUTPUTS
  - COMMAND (TO LOWER EXPERT SYSTEMS)
  - REQUEST FOR INFORMATION
  - EVENT SCHEDULES AND PRIORITIES
  - NOTIFICATION TO GROUND OF NONCORRECTABLE ERRORS
  - HISTORY FILE
- HARDWARE
  - SPERRY IT
    - -- 7.14 MHz CLOCK
    - -- COLOR GRAPHICS MONITOR
  - SERIAL CABLES
- COMMUNICATIONS
  - 2RS232 SERIAL INTERFACE TO SYMBOLICS 3640
- SOFTWARE
  - EXPERT SYSTEM WRITTEN IN TURBO PASCAL
  - SYNCHRONOUS COMMUNICATIONS ON BOTH SERIAL LINES ACHIEVED

and schedules it, based on the current operating state of the system using a "least cost" algorithm.

4.3.2 EES User's Operational Description

The console normally displays the main menu (shown in Figure 3).

When this and most all other screens are displayed, EES is ready to accept input from the user or receive data from lower level expert systems.

For example, if item 2 (FIRES details) is selected from the main menu, the FIRES menu is displayed (see Figure 4) which allows FIRES specific options to be selected.

Whenever a fault is isolated by one of the lower level expert systems and sent to EES, the operator is alerted to an anomolous situation. Notification is accomplished via an alarm window which pops up (non-destructively) in whatever window is currently active. Figure 5 provides an illustration of this feature.

5.0 SUMMARY AND CONCLUSIONS

The major thrust of this investigation was the establishment of the framework for a test bed capable of supporting an investigation of distributed expert system processing with hierarchically organized domains of responsibility and control. As described in this paper, the framework has been established.

The current configuration consists of an executive system which coordinates the activities of three individual expert systems at the next lower hierarchical level. The secondary level expert systems have separate domains and provide status summaries on their individual areas of responsibility to the executive. Although the domains are separate, the functions are similar, namely performance monitoring, fault detection and fault isolation.

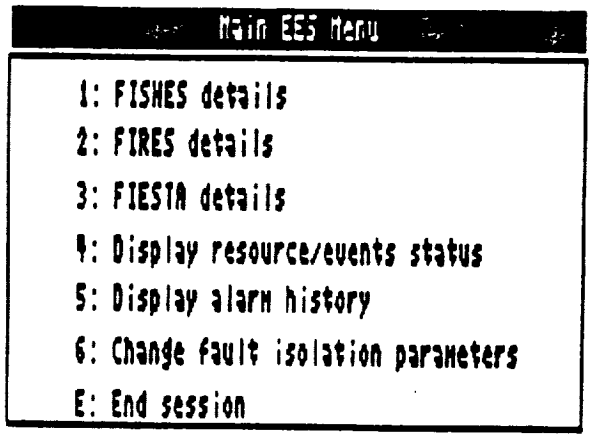

FIGURE 3: MAIN MENU

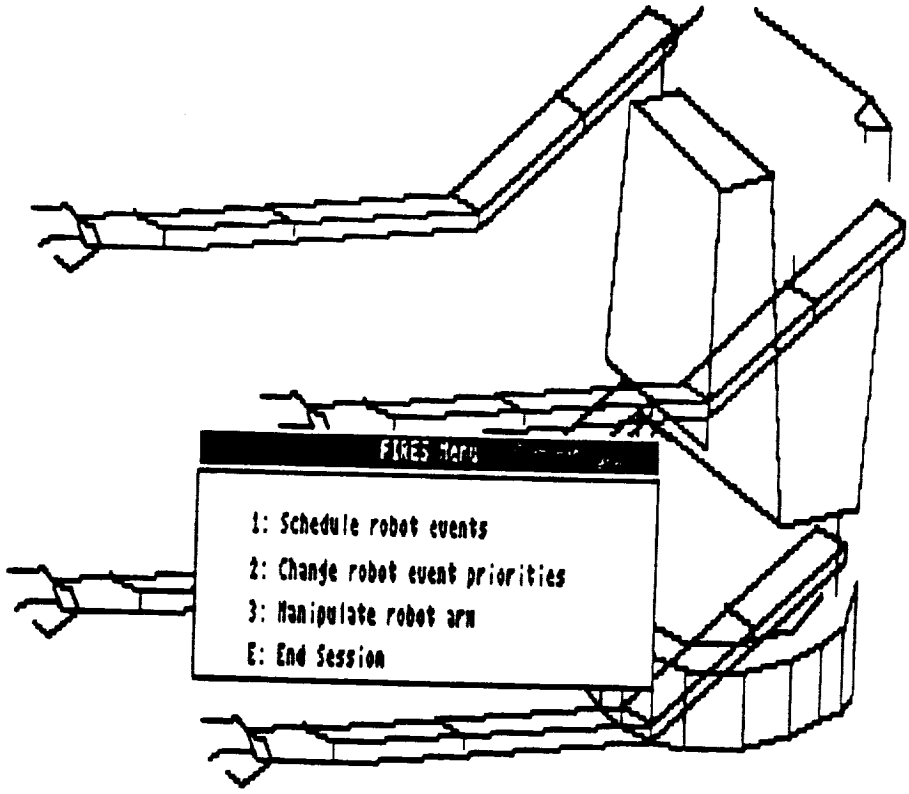

FIGURE 4: FIRES MENU

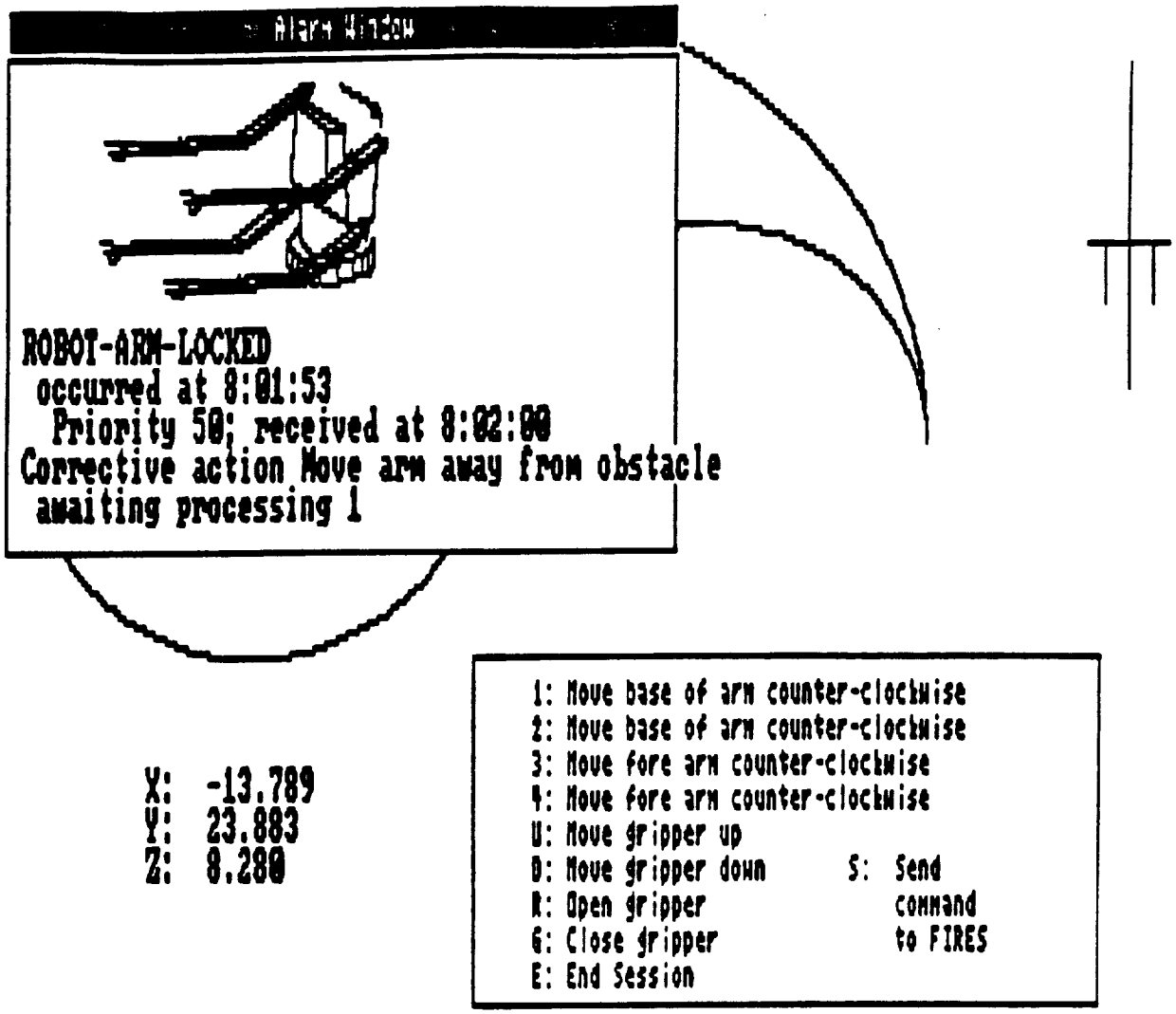

FIGURE 5: ALARM WINDOW NOTIFICATION (ROBOT ARM MANIPULATION WINDOW ACTIVE)

A major conclusion reached during this investigation is that a network consisting of multiple expert systems with hierarchically distributed control can be readily established. The feasibility of such a configuration was verified by establishing a operational test bed exhibiting these characteristics.

The simulation data was correctly monitored by the separate fault isolation system; nominal conditions being (implicitly) noted and non-nominal being detected. A review of the data indicated that the anomalies were being correctly identified. The executive demonstrated its ability to co-ordinate the resource of independent systems and correctly assign available resources to achieve problem resolution.

Another major conclusion (also implemented in the demonstration test bed environment) involved techniques for implementing the hierarchical control. It was shown that conventional software engineering techniques in the area of communication protocol, integrated with an expert system executive process, was capable of supporting the candidate architecture. The SCI was also shown to be capable of supporting integration of an existing expert system (FIESTA) into the hierarchical structure.

Many open questions remain in the area of di_tributed processing and control. The initial test bed structure provides an environment to support investigations is this area.

ACKNOWLEDGMENTS

The results reported in this papaer are drawn from "FINAL Report Phase I Study: Fault Processing using Axiomatic/Hypothetical Methods in a Multi/Level Expert System Environment", TR860155, 26 September 1986 by Steve Miksell, Sue Cofer and Edwin Zakrzewski.

That work was supported under an SBIR (Small Business Innovative Research) contract from NASA; Contract No. NAS5-29280. Research was conducted under the guidance of NASA/GSFC code 735. Their support and encouragement were appreciated and are acknowledged.