

# **AN APPLICATION OF DISTRIBUTED PROCESSING (IN THE TDRSS GROUND COMMUNICATION FAULT ISOLATION AND MONITORING SYSTEM)**

**R. L. Waddell, Jr. and H. K. Utterback  
Space Analysis and Computation Group  
The Johns Hopkins University  
Applied Physics Laboratory  
Johns Hopkins Road  
Laurel, Maryland 20707**

## **ABSTRACT**

A Fault Isolation and Monitoring System (FIMS) for analysis of the quality of data streams transmitted between TDRSS and NASA's Network Control Center has been completed and installed (Version 1) in the NASA White Sands Ground Terminal.

Due to the demands on the system's central computer, it was decided that as much of the operator interaction as possible should be handled by the intelligent terminal serving as the operations control console. This paper describes the methods and protocol used to implement the operator support software required to configure the monitoring system and to display and report the results of the analysis.

## **INTRODUCTION**

The function of the Fault Isolation and Monitoring System (FIMS) is to provide qualitative and quantitative assessments of the communications between the Tracking and Data Relay Satellite System Ground Segment (TDRSS-GS) at White Sands, New Mexico and the Network Control Center (NCC) at the National Aeronautics and Space Administration/Goddard Space Flight Center (NASA/GSFC) in Greenbelt, Maryland. Monitoring is performed as specified by the NCC to insure channel integrity and as an aid in determining the sources of failures or malfunctioning equipment.

All data to be analyzed enters the FIMS through the Distribution and Switching System (DSS) via a "TEE" connection in the main data path between the TDRSS-GS and the

NCC. The DSS has a local command and display console for monitoring and rerouting the data “tapped” from the main data path. For FIMS monitoring, selected channels are routed to the Frame Analyzer (FA) System. Each of the thirty-two microprocessor based frame analyzers can accept a data/clock pair of telemetry or command data and measure many characteristics of the synchronization word. The characteristics to be measured and the channels to be monitored are selected by the NCC through scheduling messages sent to the FIMS. For more information on the FIMS hardware and interfaces see References 1 and 2.

The FIMS- operator receives the scheduling messages and configures the frame analyzers at the FIMS console, a Chromatics CG 1999 intelligent terminal (CGT) . The frame analyzers send summary information to the FIMS computer, a Sperry-Univac V77-600 (V77), which accumulates statistics based on this information. The V77 forms status report block messages and transmits them automatically to the NCC via the DSS. From the CGT console, the operator has the capability to:

- monitor dynamically accumulating or summary FA statistics maintained by the V77;
- create and store (on V77 mass storage) up to fifty FA configurations, assignable individually or in groups of thirty-two or fewer clustered under a single name;
- recall and modify stored configurations and groups;
- receive, display, and print text messages received from the NCC;
- create, modify, store, and transmit text messages;
- selectively enable/disable automatic sending of FA reports;
- selectively enable/disable automatic printing of text messages.

Due to the real-time demands on the V77 to accept asynchronous input from the FA’s and the NCC, it became apparent that as much of the operator interaction as possible should be handled by the CGT.

## **DESIGN PHILOSOPHY**

The options available to the operator for the control of the FIMS are divided into a number of groupings of related control and monitor functions. Each such grouping, called a service, is presented as a series of one or more pages, or screen images, which associate displayed data with related control functions.

The screen image is divided into fields as shown in Figure 1. The MAIN FIELD is used for data display and menu presentation. Prompts and brief instructions for use of the current page are presented in the PROMPT area, and the service-page title is presented in the TITLE field. Greenwich Mean Time, updated at nominal one-second intervals, is presented in the upper right corner. Just below the GMT is a region reserved for alerting the operator to situations requiring his attention. The LIMITED TEXT I/O area is used for display of interactive error and confirmation messages, and for the echoing of alphanumeric text input by the operator.

The small boxes along the bottom of the screen serve as labels for the eight function keys on the top row of the terminal's keyboard. The labels are changed with each change of service-page to reflect the function implemented by pressing the associated key. It is the pressing of these function keys, after entry of appropriate information, which initiates a request for action by the host computer (see Figure 2).

## **FUNCTIONAL REQUIREMENTS OF THE V77-CGT INTERFACE**

The services provided by the current version of the FIMS are: SELSER, or the select-service service which runs automatically upon system start-up and presents to the operator a menu of available services; FANSER, or frame analysis service, which provides the capability to setup and monitor the results of the frame analyzers and report those results to the NCC automatically; and COMSER, or communications service, which allows the creation of free text messages and their transmission to NCC, as well as the receipt display, logging and automatic acknowledgment of scheduling and free text messages from NCC.

The support of these services entails three distinct functional requirements of the V77-CGT interface. First, to minimize the operator's start-up procedures, the operator Services EXecutive (OSEX) program in the V77 must be capable of downloading the operator support programs in the terminal, collectively called the Terminal Intelligence EXecutive program (TIEX). The downloading method is also used for service changes, when, in response to an operator request, a different program is loaded into the terminal memory. Second, the OSEX program must be able to display the data it has accumulated and the control information given to it for storage. Most of the data may be posted to the screen as part of the presentation of the page image upon service-page change. Some items, however, require a dynamic interaction between the V77 and the CGT concurrent with ongoing operator support. These items are GMT updates, alert message posting, and, for one page of FANSER (Figure 3), a five-second update of active frame analyzer data. Finally, the V77-CGT interface is required to handle the transmission of operator input: data stream monitoring requests, menu selection, and editing of stored message drafts and frame analyzer setup files.

## THE INTERFACE

The TIEX programs are written in BASIC and execute under the CGT's BASIC operating subsystem. BASIC has the ability to call on any functions available under the so-called CRT operating system, including all graphics capabilities, cursor control commands, and logical device reassignment. The disk operating system is also accessible from BASIC, a feature used to facilitate the remote loading of TIEX programs without operator intervention.

Host communication with the terminal is via four serial input/output (SIO) ports. They are RS232 standard interfaces, and the status registers and data registers are directly accessible to TIEX through the intrinsic BASIC function INP. Other I/O devices available are the keyboard and the screen. To access these physical devices, ten logical input and ten logical output devices are independently assignable to the physical devices, up to two input and output physical devices per logical device. Host emulation of keyboard input can, therefore, be accomplished by assigning both the actual keyboard and an SIO as physical devices to the logical device from which the keyboard handling system expects input. Some caution is required in the use of this method, as input is taken on a character-by-character basis, allowing two input streams to be merged if they are sent simultaneously. The association of a logical with a physical device is implemented by a four-byte command given to the CRT operating system.

On the V77 side of the interface, standard FORTRAN writes are used for most output. A special routine, CCMIO, was developed to allow a FORTRAN program much greater control of the I/O process, including the choice of wait or no wait reads and writes, the queuing and dequeuing of read and write requests, and the ability to check the status of pending I/O. Using CCMIO, a double-buffered read queuing method was implemented, assuring that operator requests for service support would not be ignored by OSEX regardless of loading levels on the V77.

The service support requests are initiated by the operator by his entering required data, if any, and pressing a function key, or, in some cases, a carriage return. TIEX recognizes which function key was pressed, verifies that any required support data has been previously entered, and transmits the request (or posts an error message prompting for any lacking input). The request format consists of a minimum of four bytes, all ASCII characters. The first two bytes identify to OSEX the function key struck. Since all service-page changes are supported by both OSEX and TIEX, OSEX is always able to discern what interpretation the identified function key is to be given, despite the page-to-page variation in meaning of any given keystroke. The next two bytes tell OSEX how many bytes of required support data will follow, ranging from zero to ninety-nine. The required support data may contain the results of editing a screen full of text for message drafts or

frame analyzer setups, the name of a stored draft or setup, the number of a received message for display, a frame analyzer number, etc.

Two different methods are employed for dynamically updating screen data, depending on the volume of data to be displayed. For a large volume of data, TIEX tells OSEX when it is prepared to receive the data and then turns itself off. OSEX then takes over the keyboard and “types” the data into the CGT, restarting TIEX on completion. This method clearly involves locking out the operator. For small amounts of data, characters are sent one at a time to the TIEX program, and posted to the screen when an end-of-list character is received.

## **THE IMPLEMENTATION**

The FIMS is started from the V77 system console with a single keystroke. The CGT must be powered up, with the SIO ports setup for the proper baud rate, etc., and the logical-to-physical device assignments established by accessing a stored command file. The OSEX initialization routine emulates the CGT keyboard as described above, sending to BASIC the command to load the TIEX SELSER program from the CGT floppy disk drive. After a five-second pause to allow completion of the loading, OSEX issues the RUN command. Similarly, when the operator requests that a new service be loaded, TIEX suspends itself, allowing BASIC to return to command level, and OSEX loads the requested service and restarts TIEX.

Interrupt processing is used to catch and interpret operator keystrokes. BASIC allows one physical device to be enabled for interrupts. With keyboard interrupts enabled, the pressing of a key causes program control to be vectored to an Interrupt Service Routine (ISR). The first step taken by the ISR is to determine which service-page is currently active. Next, the keystroke is identified and interpreted as required for the page. Keys which have no significance for the current page result in error messages posted to the text I/O region. The response to a valid keystroke may be as simple as echoing the struck key to the current cursor position; it may involve a sequence of steps to test and adjust screen cursor position pointers, buffer pointers, and buffer contents; or it may require the transfer to OSEX of requests for support accompanied by appropriate data gathered from previous keystrokes.

When a key is recognized by the ISR as requiring a page change, the main field is cleared, OSEX is informed of the request, and a subroutine is executed. For each page of the service there is a subroutine containing the graphics and text characters which constitute the portions of the page unique to it. After these strings of symbols are displayed, preparations are made for handing control of the keyboard to OSEX. The physical keyboard is disabled to prevent the operator from merging keystrokes with those which

will be sent by OSEX. The screen is then disabled, so that system messages will not disturb the page image. Finally, TIEX sends OSEX a signal that it is ready to relinquish control, and executes a STOP statement. Upon receipt of the signal, called a main field request, OSEX sends to the CGT codes which cause the terminal to leave the BASIC subsystem and enter the CRT operating system. Thus, the subsequent characters will be displayed in the dumb terminal mode, rather than interpreted as commands. After posting all the required main field data, OSEX reactivates the BASIC system and restarts the suspended TIEX program. TIEX then enables the keyboard and the screen and posts the prompt to the operator for the new page, informing him that the keyboard is under his control again.

While waiting to receive operator keystrokes, TIEX polls continually for characters on a separate SIO port. The characters are sent by the Auto Analysis Management (AAMA) program in the V77, and are used to update the GMT, the frame analyzer dynamic reports, or the operator alert message. TIEX reads the status register for the SIO port and tests whether the character-ready flag is set. If it is, the character at the port is read and tested for validity as one of the three distinct flag characters. In case the character is not one of the flags, a character is sent to AAMA telling it to abort whatever update is in progress. If the flag is recognized as requesting an update of GMT or an alert, TIEX sets cursor positioning variables for posting the GMT or alert, clears an update buffer and a character counter, and sends an ok-to-send character. Then the status register is tested until another character is received. As long as the character count maximum is not exceeded and an end-of-list or abort character is not received, each character is loaded into the update buffer and an ok-to-send character is sent in response. When an end-of-list character is recognized or the character count maximum is reached, the update is posted to the screen.

If AAMA detects that too much time ( $>100$  msec) has elapsed between sending a character and receiving the ok-to-send acknowledgment, it sends TIEX an abort character and stops the sending process. TIEX abandons the character gathering procedure and returns to the flag searching loop. The purpose of the abort procedure is twofold. First, it prevents GMT from being posted if the delay between characters is excessive, so that old times will not appear. The more significant purpose lies in the assumption that an excessive delay implies a foul-up in the handshaking procedure. AAMA is able, with this method, to go back and try again with a flag character, and TIEX, wherever it is in its process, is informed that it should give up and wait for a new request flag.

The update polling procedure must allow keyboard interrupts at any point in its execution. operator keystrokes take priority over updates, and they cause control to be vectored to the ISR. Most keystrokes entail echoing the struck character to the screen or otherwise moving the cursor as part of the response. To ensure that the correct screen position is maintained, careful tracking of the cursor is required. If a keyboard interrupt is received

while the update software has the cursor in the GMT or alert fields, it must first be returned to its operational position for the keystroke response, then placed back at the update position for the completion of update posting, and finally replaced at the new main field operational position. To accomplish the cursor tracking, the cursor position is stored before servicing a keystroke interrupt and again just before returning from the ISR. By testing the location before the keystroke, TIEX can determine whether an update was in progress, and if so, return to the section of code which will complete the process. If no update writing was in progress, the ISR returns control to the point at which a new update flag is awaited.

For frame analyzer dynamic report updates a method like the page change main field request is used. Rather than gathering the characters one at a time, TIEX prepares as described above for a page change and hands control to OSEX. This locks out the operator for a brief period, but is not disturbing in practice.

## **CONCLUSIONS**

The flexibility gained in the design and implementation of a complex software system through the use of intelligent terminals is well worth the substantial efforts required to establish communications protocol within the system. Especially significant is the removal from the host computer of the human interface, with its demands for asynchronous, input-driven programming.

## **REFERENCES**

1. Zink, H. D. and Grunberger, P. J., NASA Ground Terminal at White Sands Engineering Segment Capabilities, JHU/APL Special Report S3C-1-0121 Laurel, MD, March 1980.
2. Space Analysis and Computation Group, NASA Ground Terminal at White Sands Engineering Segment Software, JHU/APL Report SDO-5679, Laurel, MD, June 1980.



**Figure 1. Screen Image Fields**



**Figure 2 Function Key Labels Refer to Right Side, Top Row.**

