THE MAN-MACHINE INTERFACE IN COMPUTERIZED TELEMETRY SYSTEMS

Tilo F. Reber
Systems Development Section
Physical Science Laboratory
Box 3548
New Mexico State University
Las Cruces, NM 88003

ABSTRACT

This paper concerns itself with the interface between men and computerized telemetry systems. This interface relates to the operation, planning, and implementation of setup and data processing functions.

One of the major problems in operating a telemetry system is the programming of equipment parameters. This programming is often done by skilled “real-time” software personnel. This is both a costly and restrictive approach. The author has developed a “friendly” menu-driven, operator interactive, approach to solving these problems. The man-machine interface consists of software developed to present telemetry system options to an operator for selection. These options are displayed via the operator’s CRT. These displays are menus and they are formatted to display operator options in telemetry language. The object is to allow normal telemetry operators to configure the equipment setup and the data processing parameters. Once the configuration has been defined, the system can be configured quickly and precisely by the computer software. Changes to a setup or data processing configuration can be made by telemetry operators without the help of full-time programmers.

INTRODUCTION

The process of turning an encoded telemetry data stream into useful engineering data and, when necessary, doing this in real-time, is still far from being a trivial task. Furthermore, the increased capabilities of our telemetry hardware and our computers have done little to simplify the job of the man trying to “get the data out”. If anything, the newer and greater capabilities mean more options and, consequently, more places to go wrong. More recently we have become aware that computers, beyond their abilities to crunch massive amounts of numbers, may also be used to simplify other aspects of the data gathering cycle. They may be used to ease the tedious task of setting up the ground stations for a variety of missions. They may be used to help in the definition of processing to be performed and they can help to define output or data distribution modes.

SOFTWARE CONCEPT

In developing a friendly computer interface for a telemetry ground station we first addressed ourselves to a number of questions which we hoped would be answered by the design of the software. We asked:

1. How do we tailor this particular computer to work with this particular telemetry station?

2. How do we tailor the software to be structurally and conceptually simple for the user?

3. How do we get the computer to do as much of our work as possible?

4. Can we ensure that the main computer interface is simple enough to allow a non-programmer to do the job?

5. Can the computer help us make fewer mistakes?

6. Can we add more hardware, new hardware, or other processing requirements without having to develop a new software system?

The software solution which follows reflects our attempt to answer most of these questions with a practical system. While keeping the concepts as general as possible we did need to apply the system to some real hardware. A PDP-11 series computer running under the RSX-11M real-time operating system was used in conjunction with a variety of Sangamo Weston 700 series and 2700 series telemetry hardware.

CRT DISPLAY

The CRT displays presented for user interaction are a number of tightly formatted pages. Each page is designed so that it fulfills one major conceptual step in the process of eliciting the required configuration information from the user. All pages have a background mode, displayed in a gray field and a foreground made, displayed in a white field. All characters in both fields are black. The gray area is protected, so that the cursor cannot be moved there. It contains information required to fill in the white fields and specifies an allowable range of answers. The white fields are unprotected, the cursor moves only through them.

Each white field is placed next to its information field in gray and contains only the maximum amount of spaces required to hold the setup data. The unprotected fields may be tabbed or spaced through, backward or forward. Corrections may be made to any field on the page in any order. Only when the user is completely satisfied will the page be transferred to the mass storage device.

PAGING FLOW

When the user has provided all the necessary information on one page and presses the return key the next page will automatically be displayed. The paging sequence is such that the user is lead from the general to the specific. The earlier information is used by the computer to decide which pages should be presented to the user to elicit more specialized information. The user may, however, change his mind and page backward, or he may at any time start at the beginning.

THE HUMAN INTERFACE STRUCTURE

The decision and information structure imposed by the software may easily be seen by walking through a sample ground station setup task.

> The program is started and the required conditions are initialized by simply typing @SETUP.
>
> The first formatted page is displayed. Here the user chooses either to use a pre-defined setup configuration or to edit a new or old configuration. This is done by placing a character in the box next to the desired option.
>
> Assuming a pre-defined setup is chosen, the computer next displays the names of all currently defined and stored setup configurations. The user then places any character next to the chosen configuration and hits the carriage return. The computer will now complete the configuration and setup the station. Note, the user’s total interaction has been to start the program and to place two characters in predefined CRT boxes. For a vast majority of the cases, this would be all that would be required. Only if a new or altered station setup were necessary would we do more.
>
> Had the user chosen the edit mode, the interaction sequence would have been more extensive. In this mode the user is first given a choice of configuration formation options. These are: add a new configuration, copy an old one, delete an old one, modify an existing configuration, rename a configuration, or produce a printed record of an existing configuration.

Once the computer knows the nature of the operation that is required it goes on to ask for more detailed information. The amount of information still needed depends upon the function to be performed. Figure 1 shows graphically which pages and path will be taken by the program for any given choice. The longest of these paths is taken when a new configuration is to be added. Taking this sample path from the editing process selection page would yield the following result.

First, a newly formatted page would allow the user to enter 24 characters as a name for the new configuration. Having entered the name, the user is presented with the next formatted page.

This page contains a list of every programmable hardware item in the telemetry station. The next step, then, is to place an A (for Add) next to each item the user wishes to use in the run-time configuration.

This done, the user is finally given a single formatted page, for each device, upon which he enters the setup parameters required. For example, he may have the frame synchronizer page where he fills in the fields for words-per-frame, bits-per-word, etc. He need only use decimal numbers. All conversions and bit shifting will be done by the program.

Upon completion he has a stored data base which he may reuse at any time in the future to program his hardware.

STORAGE AND REUSE

All setup information is recorded as part of a disk data base for permanent storage, portability, and access speed. Disk file storage and access are transparent to the user. A set of files can be created for any project simply by requesting them from another formatted page. Each project data base may contain up to 20 different setup configurations. The total number of project data bases allowed is dependent largely upon the capacity of the disks used. We are able to accommodate 20 separate data bases each with 20 complete setup configurations on a 10 megabyte disk.

ERROR AND LIMIT CHECKING

Although the computer cannot know what the user wants, it can point out inconsistencies in many cases. As was previously pointed out, our program will define acceptable limits for most data entry requirements. Should the user enter an out-of-limits value in a given field, the program will flag the field and issue an error message. The user may then enter the correct value and continue. To protect old files, the program requires a verification before deleting any files that have been marked for deletion. New names are checked for uniqueness. The failure of any hardware device to respond to the setup instructions is displayed to the CRT. The program also contains internal checks which are transparent to the user. This is to verify that it is running properly.

CONVENIENCES

A certain group of features were included which are not strictly necessary for the software to perform its task. These came about as a result of experience and a realization of their need.

For those who have long programmed their hardware from punch cards or front panels there is often some confusion and distrust about what the computer is doing. We have, therefore, included a verification mode. In this mode the final output buffers are listed in octal format. This allows the user to check the correlation between what the computer is doing and what he would expect from a card or front panel setup.

A print mode is provided for those that need hardcopy records. All the necessary pages for the selected configuration are printed exactly as they appear on the screen.

Because not all telemetry data streams are as clean or well defined as one would like, we have provided a “fiddle mode”, whereby the user may experiment with a single parameter on a single device while keeping all other parameters constant. This is akin to pushing the buttons on the front of a bit sync until it locks up. The difference is that when the right combination is found it is automatically stored as part of the permanent data base.

FLEXIBILITY

The particular grouping of hardware which goes to make up a telemetry station is not always constant. The problem then is to design the software so that it may easily be adapted to new hardware situations. When speaking of setup software flexibility, then, we are concerned with three categories.

1. Flexibility of present hardware use,

2. The ability to add more hardware of the type already in the telemetry station, and

3. The ability to add completely new hardware.

The flexibility to use certain pieces of the telemetry station and leave out others in order to form a particular run time configuration is an inherent part of the software and the setup procedure. Adding more hardware of the type already present must be done by a programmer familiar with the software. But it is a simple task which can be done in approximately three hours per device. Completely new devices may also be added. The time this takes is dependent upon the complexity of the device. A new device with the complexity of an EMR 710 PCM Decom takes about three days to integrate into the system. A simpler device such as an EMR 720 Bit Sync or an analog tape recorder may be integrated in one day. Some non-standard devices have also been fitted nicely into the software. We have, for example, added programmable multiplexers to relieve the burden of signal routing and patching. The important thing here is that the modular design of the software allows changes to be made in the form of additions rather than in the rewriting of the existing code.

PROCESSING SOFTWARE

To this point we have concentrated mainly upon the telemetry station setup aspects of our software design. We have also found that the design could easily be adapted to real-time data processing. The only change in concept comes at the lowest level of the software structure. Rather than dealing with individual hardware devices and the definition of how they are to be setup, we deal with real-time processes and the definition of how they are to run. In this way we achieve consistency and simplicity in the use of the software. The following real-time modules have been developed for our systems.

1. Real-time data path definition (how the data is formatted and moved from the hardware into the computer)

2. Digital tape logging and playback

3. Data output to printer

4. Formatted display to CRT

5. Normalization of data

6. Limit checking of data

7. Polynomial conversion of data

8. Data output to digital devices to analog conversion

Data to be displayed or logged may be either raw or converted. The important point here is that complete real-time data processing configurations may be defined and stored before a mission and reused whenever they are needed.

CONCLUSION

This software design has been implemented and tested at the Telemetry Data Center of White Sands Missile Range and seems to be performing according to our expectations. With a few key strokes the user can configure the entire station, and with a few more he can perform extensive real-time data processing. The system has been used successfully by non-programmers and in a variety of hardware configurations. The major problems we have encountered have been with the limited speed capabilities of doing data conversion in real-time. Future plans, then, are to move this task from software to hardware. In our design, the function will move from the processing software to the setup software. Work is currently being done to allow the system to do real-time graphic displays.

ACKNOWLEDGMENT

Thanks to Donald Cisney for developing the concept of a menu-driven, data-base telemetry system.

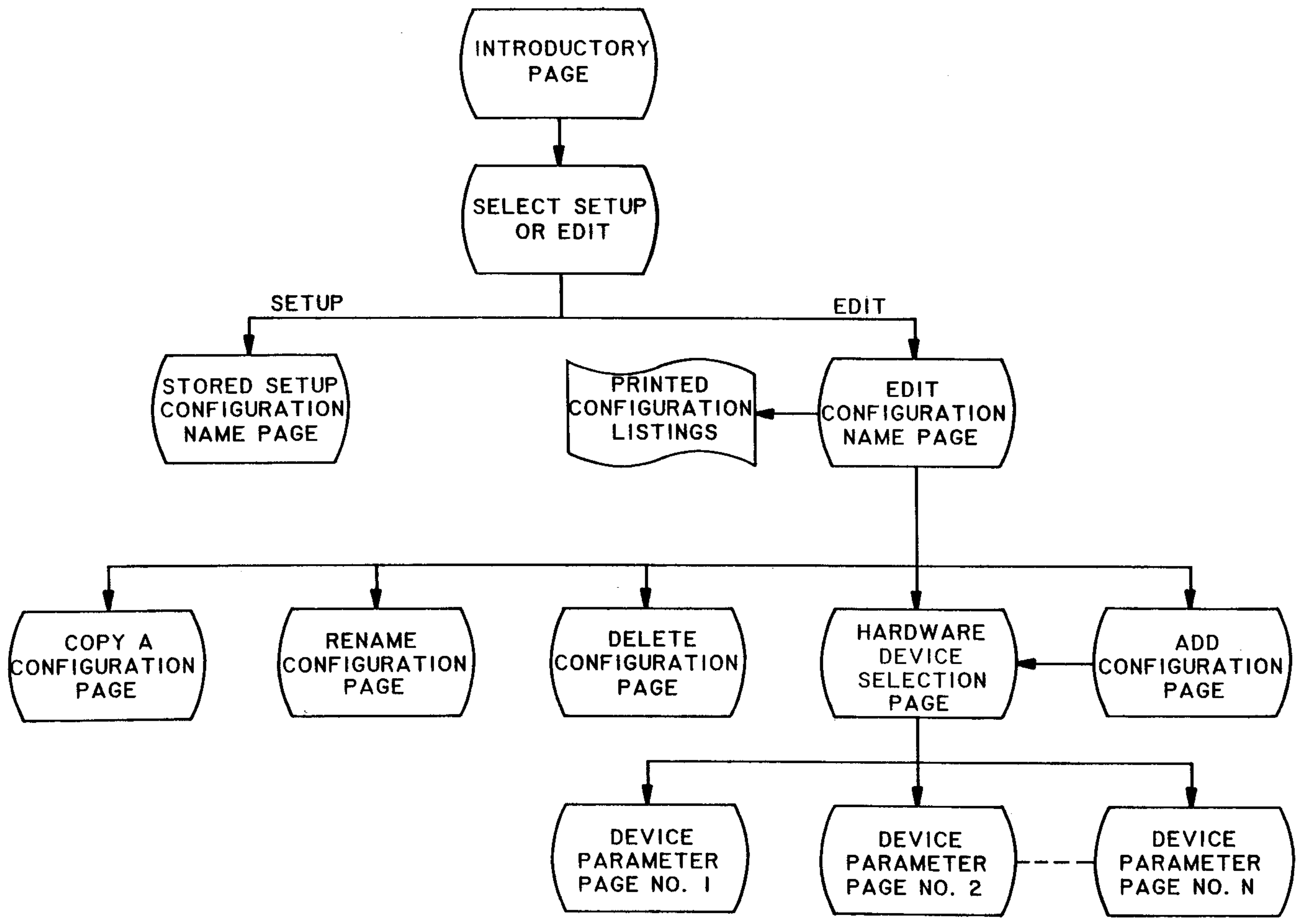
INTRODUCTORY PAGE
SELECT SETUP OR EDIT
SETUP
EDIT
STORED SETUP CONFIGURATION NAME PAGE
PRINTED CONFIGURATION LISTINGS
EDIT CONFIGURATION NAME PAGE
COPY A CONFIGURATION PAGE
RENAME CONFIGURATION PAGE
DELETE CONFIGURATION PAGE
HARDWARE DEVICE SELECTION PAGE
ADD CONFIGURATION PAGE
DEVICE PARAMETER PAGE NO. 1
DEVICE PARAMETER PAGE NO. 2
DEVICE PARAMETER PAGE NO. N

Figure 1   Setup Program Menu Pages