A REAL TIME MULTIPROGRAMMED TELEMETRY SYSTEM

G. B. MUSE

Summary A study of several Telemetry Data Reduction Systems has revealed some common problems which reduce the effectiveness of these systems. The problems include: The high cost of real-time computer analysis programs, lengthy turnaround for data product modification, lack of testing flexibility, and reduced hardware system utilization. A computer system, 11TELFILE11, has been implemented to improve upon these problem areas. This system is operational at the Space and Missile Test Center (SAMTEC), Vandenberg Air Force Base, for support of the Minuteman III Weapons Systems.

Introduction A partial solution to these problems would be the development of a "Telemetry Computer Language". If a Test Engineer could command his telemetry data reduction system in a language he understands, flexibility in testing and relative independence from computer programmers could be gained. This independence would allow the engineer to derive exactly the data product he wishes and to make modifications to the testing program himself in a timely manner.

Telemetry languages have been developed, but many have not been successful; one reason has been the cost of language development. Computer manufacturers have spent millions of dollars to develop, refine and maintain non-telemetry computer languages (e.g. FORTRAN - COBOL) and the costs of these developments can be applied across many system users. But, when a single organization has endeavored to develop a "Telemetry Language", its costs are generally supported by a single user whose funds are more limited; consequently much of the needed capabilities and basic maintenance cannot be afforded--and the system really never performs as was planned.

Another problem in telemetry language development is complexity. Languages have been designed but never completely implemented because of the size and complexity of the task undertaken. These languages would allow tremendous flexibility in defining almost every type of telemetry format or mathematical calculation; but the complexity of the systems delayed implementation, increased costs, and in the end, only a small portion of the options were implemented.

Discussion The 6595th Aerospace Test Wing at Vandenberg Air Force Base had need to implement a telemetry language to aid its contractors in the checkout of the Second Source Minuteman III on-board guidance system.

At Vandenberg, a Control Data 3300 computer system was available to Minuteman for Telemetry Data Reduction. The Minuteman engineers and the Control Data Analysts at Vandenberg defined and implemented a system which provides the flexibility and convenience of a complex telemetry language, and it was developed with relatively straight-forward concepts.

The philosophy of this system (called TELFILE) was to apply to telemetry use a higher level computer language, which has already been well defined and developed--in this case, FORTRAN. It was thought that if telemetry data could somehow be made available to FORTRAN programs, the test engineers would have no trouble in learning the FORTRAN language and in writing programs to reduce their data in the manner they wanted.

In the TELFILE System, a computer program decommutates, reformats, and writes selected parameters of the Minuteman PCM data stream onto a Disk File. The data contained in the file (called a DATAFTLE) is in a format which is easily accessed by FORTRAN programs. Hence, the Minuteman engineers from Autonetics, TRW, Honeywell and Boeing have written dozens of FORTRAN programs which read their telemetry data from the DATAFILE and run through various calculations, producing their desired outputs.

Multiprogramming The TELFILE System has been programmed to operate under the 3300's Mass Storage Operating System (MSOS). One of the features of MSOS is its multiprogramming capability.

This multiprogramming feature allows two programs, one priority and one batch, to be resident in the 3300 simultaneously. The priority program, operating under MSOS, is interrupt driven: That is, the program is called into execution only when it must respond to external stimuli such as decommutator frame interrupts, disk end of operation signals, and operator commands.

Under MSOS, the batch program is normally executing, except when the 3300 receives one of the priority program”s interrupts. At this time, MSOS interrupts the batch execution and transfers control to the proper priority program.

In the case of the TELFILE system, the TELFILE Program operates as the priority program, and the FORTRAN processing programs operate as the batch programs.

By utilizing the priority/batch processing capability of MSOS, it is possible to simultaneously acquire telemetry data on a priority basis while processing previously acquired data in the batch mode, using FORTRAN programs. In the priority/batch environment, the priority program retains control and thus responds to the various stimuli initiating or terminating the acquisition of real-time telemetry data.

System Operation Figure 1 reviews the SAMTEC 3300 Telemetry Processing System while Figure 2 depicts the flow of data, status and control throughout the system.

With TELFILE as the priority program and a FORTRAN processing program as the batch, the system operates as follows: A frame interrupt is received from 1700 decommutation computer by the 3300 computer, and control is transferred to TELFILE. TELFILE inputs and reformats telemetry data from the "front-end" decommutation equipment. The reformatted data is written on the disk DATA FILE, and then control is given back to the batch program. The batch program can now access the telemetry data from the DATAFILE through normal FORTRAN read statements.

Under nominal conditions, TELFILE has been using about 12 per cent of the 3300's central processor's power. The system is limited, however, by the disk system. Under high data rate conditions, TELFILE can demand sole use of the disk for recording purposes and, of course, processing of data in the batch program falls behind until the recording rate is reduced or stopped.

A feature of the TELFILE Program is its ability to examine the telemetry data and determine if it should be recorded on the disk file. This capability is called event detection, and allows the test engineer to define to TELFILE a description of the types of data (EVENTS) he wishes recorded on the file and subsequently be processed by his FORTRAN programs.

Operational Results The results achieved with the system have been impressive. TELFILE has been used in a number of different modes, which include:

1. Evaluation of Minuteman III pre-launch guidance system calibration data.

2. Evaluation of missile configuration anamolies when checkout of the Second Source Honeywell on-board improved digital computational unit (IDCU) was being performed. The telemetry system via TELFILE and FORTRAN programs was a tremendous help in locating system problems by providing IIDCU core dumps, easy display of computer word data, selectable computer word search capability, and many FORTRAN written special purpose checkout routines.

3. Evaluation of flight data for First and Second Source flights. This has included such things as complete discrete tabulation, engineering unit time history lists, terminal count down program, and a complete flight evaluation program.

Results of theisystem have been significant from several standpoints, all related to the fact that the engineers have been able to construct their own test programs. The cost of implementing new functions with TELFILE has been reduced by an order of magnitude. For example, a launch terminal count down program was implemented in less than one man month. Similar capabilities have been implemented on other systems at costs of several man years. The key to the TELFILE System is, of course, that once the real-time portion of the system is implemented, it remains unchanged even though new capabilities are added through new FORTRAN programs.

The system has also permitted engineers to take on tasks not previously performed by computers; or, in fact, not previously performed at all. There has been a great deal of experimentation on the part of test engineers to try methods or to design new ways to display their data and approach their problems.

Conclusions The multiprogramming concept used in the design of the system has proven to be extremely useful, by helping to increase computer utilization. There have been many instances when a vehicle test was being performed, TELFILE was acquiring data in the priority mode and engineers were compiling new test programs in the batch. This is a significant improvement over previous systems, where a real-time vehicle test required a dedicated computer system for large blocks of time and program development has virtually stopped.

Utilizing the capabilities of the existing MSOS operating system also proved to be useful in the final system operation. There are dozens of features already programmed into MSOS that would have cost a great deal to reproduce; such features as file manipulation capabilities, interrupt handling capabilities, and input/output peripheral equipment subroutines.

Using FORTRAN as a basis for a telemetry language has also worked well. Many of the engineers who had never programmed were able to learn the system concepts very quickly and produce useful data products within a month.

The concept of using a computer manufacturer’s operating system and compilers as a basis for a real-time system has been successful. The majority of the “monitor” type of functions normally needed in a special purpose realtime system were already developed in MSOS, and thus great savings were realized in the TELFILE development effort. This same concept can and has been applied to other real-time application areas and to other

computers. For example, a SAMTEC real-time telemetry simulation system is currently being considered at Vandenberg which would also utilize a 3300 computer and MSOS.

Several CDC 6000 series computer system programs have been developed in Real Time]Radar and Telemetry applications at Edwards Air Force Base, MIT, and Grumman, where similar concepts of operating systems have been used; and with the application computer programs written in FORTRAN.

In reflecting on the overall performance of the system, it is safe to say that it has improved many problem areas that were hindering the Minuteman Test Program. Indeed, TELFILE is much more flexible; response time to testing requirements changes is less; and the utilization of the computer equipment at Vandenberg is much higher because it is more useful.

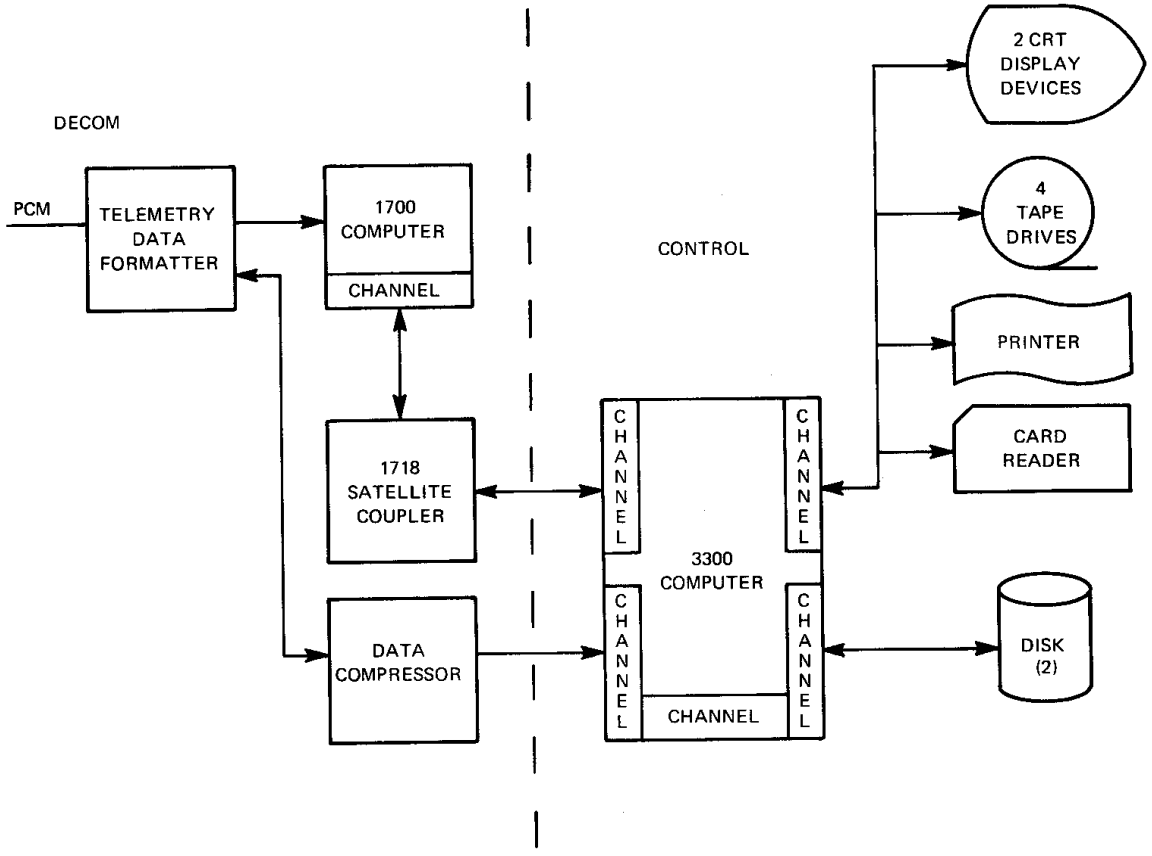

FIG. 1 - TELFILE HARDWARE CONFIGURATION AT VANDENBERG

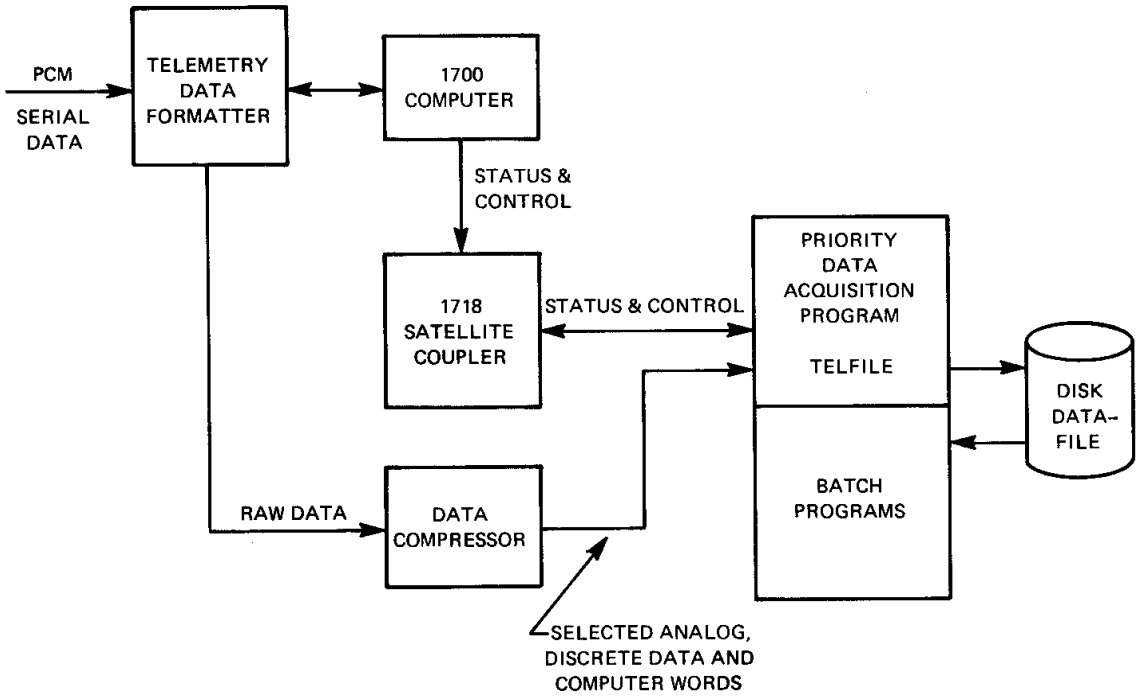

FIG. 2 - TELFILE SYSTEM DATA FLOW