THE USE OF OPEN ARCHITECTURE SYSTEMS IN COST REDUCED SATELLITE TELEMETRY & CONTROL STATIONS

David R. Spielman
(david@sd.aplabs.com)
AP Labs, Inc.
6215 Ferris Square
San Diego, CA 92121

KEYWORDS

Open Architecture, Telemetry, Command & Control, CCSDS, Real-Time, Satellite

ABSTRACT

A comprehensive examination of the market demands for cost reduced satellite telemetry & control stations will be presented. These systems are implemented using flexible, open architecture-based high performance real-time systems. The trend for combining telemetry monitoring of satellite data with closed-loop satellite command and control functions will be presented. This combined functionality opens up the possibilities for completely integrated, reduced cost satellite control systems. The market forces driving the demand for this integrated functionality include the broadening of non-military satellite applications, the widening international deployment of commercial satellites and the accompanying drive toward decentralized satellite control.

The major requirements for the telemetry processing and command & control functionality of the integrated, reduced cost satellite control system will be presented. These requirements include: full real-time performance for processing telemetry data; flexible architecture for the incorporation of a wide range of I/O devices; capability of performing real-time, closed-loop control based on conditions in the telemetry data; user friendly development environments for application-specific customization of the system; and low system costs with the capability of indigenous support. The divergent requirements of performance, flexibility and price of these integrated, reduced cost satellite control systems is made possible via the use of open architecture building blocks that include standard VME boards combined with specialized real-time software drivers and user oriented, flexible Graphical User Interface (GUI) software.

INTRODUCTION

The concept of a, "Low Cost" satellite telemetry & control station has historically been a contradiction in terms. Such a system, by the very nature of its components (large autotracking dish antenna, wideband receivers, diversity combiners, microwave power amplifiers, telemetry processing subsystems) will probably remain very expensive for the foreseeable future. However, because of the proliferation of non-military satellite applications in the United States and the widening international deployment of commercial communications satellites, the requirement for lower cost telemetry and control equipment for satellite applications continues to grow. Traditional ground station equipment, with its custom components and proprietary architectures are no longer cost effective in today's highly competitive, cost and schedule sensitive international market place. In today's environment, satellite programs are undergoing ever shortening development and deployment cycles. These facts mandate that the command, control and telemetry ground equipment used to support these missions must be highly flexible, easily reconfigureable, supportable on an international level and above all, be based around a totally open system architecture whose hardware and software components adhere to published international standards. The use of equipment and components that are available from only one vendor, or that are specific and limited in their focused application are becoming unacceptable to the majority of users.

COMBINING TELEMETRY MONITORING WITH COMMANDING

Ground systems that integrate the functions of telemetry processing with command & control are becoming more and more popular because the combination reduces program costs and development time for the end user. An example of a system that combines telemetry processing and monitoring with satellite command & control capabilities is the AP Labs VMEstation Telemetry System (VTS). Figure 1 presents a block diagram of the system. This system utilizes a real-time telemetry acquisition and analysis front end that is tightly coupled (via memory-mapped SBus-to-VME interface) with a SPARC 10 Unix workstation. The workstation acts as the display server for other X-Windows displays on the network. The real-time subsystem consists of a control CPU, the telemetry bit synchronizers, decommutators, PCM simulator, Viterbi decoder, an IRIG time code generator/reader and a high performance SCSI-2 controller and real-time archival disk. The hardware in this system is housed in a 20 slot 6U VME chassis. The telemetry decommutator boards are totally open architecture from a hardware standpoint. The decommutator boards have a 6U VME form factor and use the standard VME and VSB interfaces for the movement of information. Data is decommutated into a double buffered, dual ported memory that is accessible across the VSBbus, rather than a private/proprietary bus.

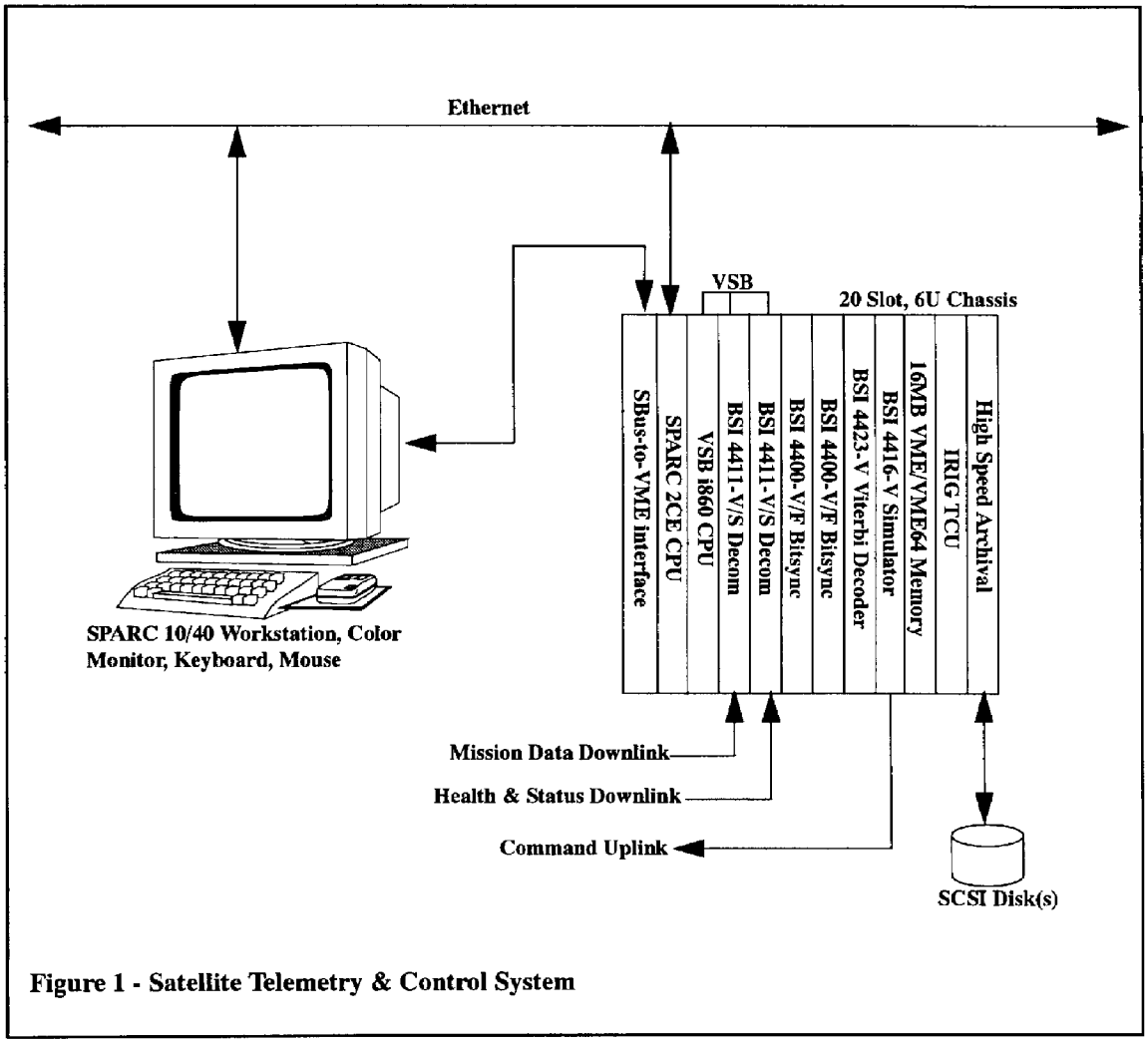

Figure 1 - Satellite Telemetry & Control System

When a buffer (frame or group of frames of data - up to 64KB) of data is available, a VME interrupt is generated and the decommutator board switches buffers. The processor board in the system responds to the interrupt, reads the decommutator buffer into local memory and performs the requested parameter processing and/or initiates the data archival to disk. Data transfers over the VSB bus occur at data rates exceeding 10 MBytes per second and are independent of traffic on the VMEbus.

The approach used by this decommutator, where a large buffer of data is acquired before passing it into the system, has several advantages over the conventional tag and data bus architecture historical to telemetry processing. First and foremost, a "true" open architecture is achieved that gives the end user much more flexibility in configuring CPU and I/O capabilities. This means that any board (there are many) that has a VME/VSB interface may be used to interface with the telemetry data stream. The user of this example system is not tied to any one vendor for system components and support. To upgrade capability, the addition of newer, faster CPUs and other interface and processing boards is straightforward. Standard VME boards from a

multitude of vendors are available for most required I/O interfaces. A second advantage of this decommutator is that frames of data from it are processed in a sequential, deterministic manner which removes the need to provide special time tagging to determine the order of parameter arrival. The third advantage is that all processing for a given telemetry stream is handled internal to the CPU board without the need to generate secondary tags and the accompanying rebroadcast of data. In the example system, there are no tags output from the decommutator; they all reside within the memory of the CPU board that handles the stream processing. This method of decommutation lowers the real-time bus loading by a factor of two. Processing and I/O overhead are kept to a minimum, allowing a much more efficient use of system hardware. Finally, this solution provides a modular architecture where additional I/O or processing capabilities may be added. By utilizing strictly VMEbus-compliant boards, the flexibility and growth potential for the system is significantly increased with minimal cost impact.

To generate the command uplink channel, a PCM simulator board is used. The simulator used in the example system, a BSI 4416-V, can be configured to output constant or dynamic data. It utilizes a double buffered output scheme, allowing data to be written to one of the buffers while the other is being output. Memory size up to 2MB per buffer allows for a significant amount of non-varying data to be output at rates up to 10 MBytes/second. The 4416-V can also be used to input data to the 4400-V/F bit synchronizer, or directly into the 4411-V/S decommutator for system testing when commanding is not required. When system self-test is invoked, the 4416-V works in conjunction with a dynamic data generation tool which allows the user to simulate waveforms and send them to database-directed parameter time slots within the telemetry frame. In the satellite command uplink mode, the 4416-V works in conjunction with AP Labs satellite command icon software and receives buffers of commands for transmission.

One major trend in current satellite programs is the adherence to the Consultative Committee for Space Data Systems (CCSDS) recommendations for telecommanding. CCSDS uses data transport protocols modeled on computer network protocols and enables interfacing with computers and commercial telephone service providers. The CCSDS requirement, with its packets of data embedded within a fixed length frame is well suited to the packet-based decommutator and PCM simulator hardware used in the example system.

The satellite command icon in the example system allows the user to issue command sequences from an on-line, dynamic command database. The command database is created, viewed and edited using either the setup screen of the command icon, or an ASCII text editor. Satellite commands are output from the command icon as

sequences of one or more Command Link Transmission Units (CLTUs). The CLTU in turn is made up of an optional CCSDS header and zero or more commands. The CCSDS header, if present, defines the fields used to encapsulate the CLTU's commands into a CCSDS transfer frame. Each command in the CLTU is then simply a sequence of bytes and is transmitted by the BSI 4416-V. Figure 2 illustrates how commands are built into sequences. Figure 3 shows how this sequence is formatted for output by the 4416-V.

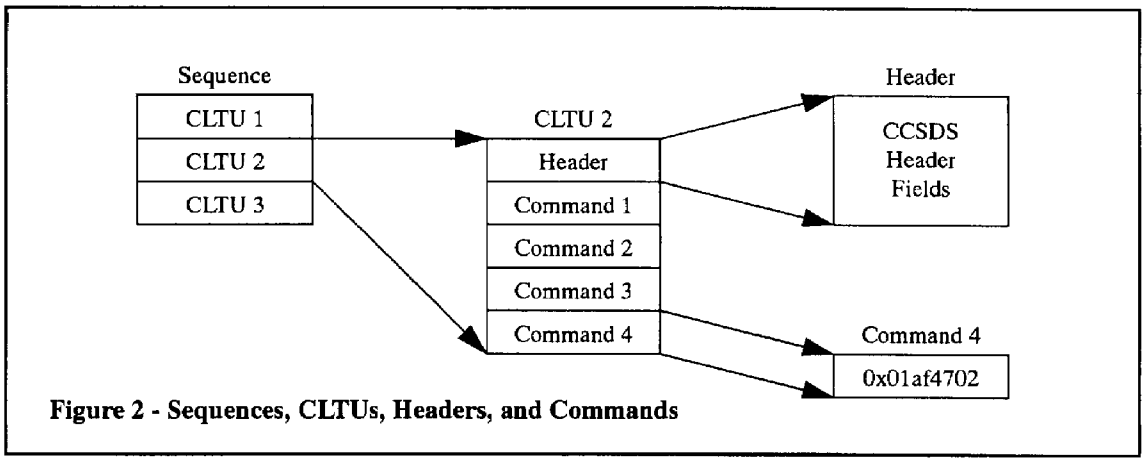

Figure 2 - Sequences, CLTUs, Headers, and Commands

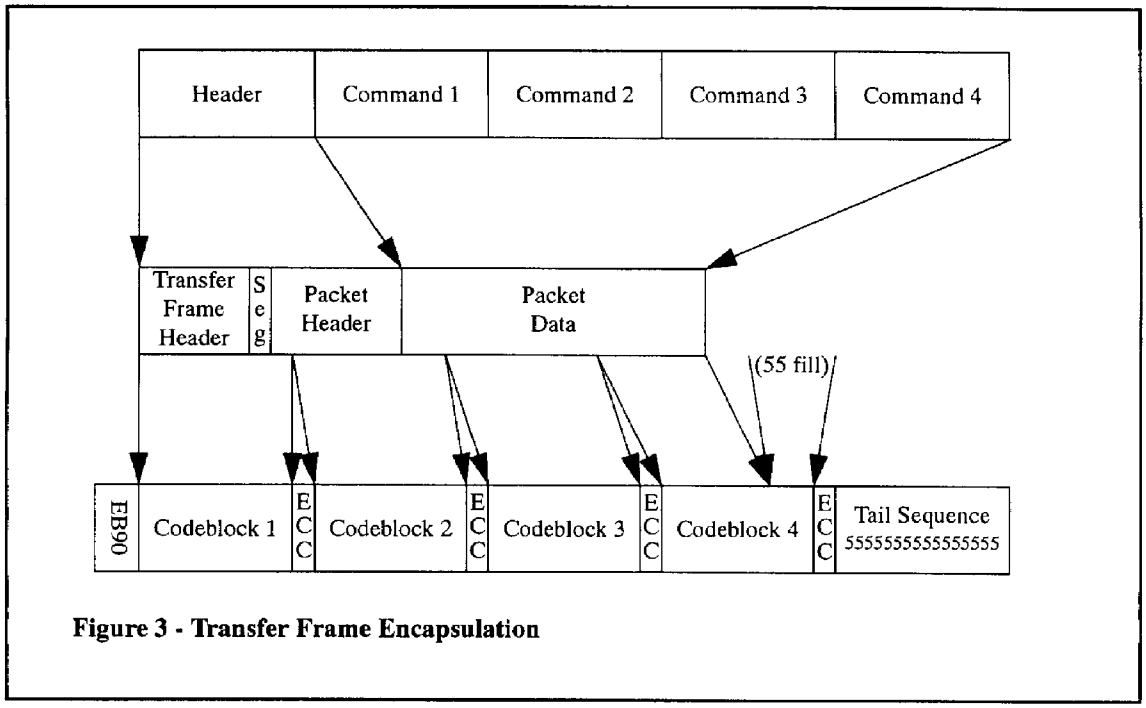

Figure 3 - Transfer Frame Encapsulation

If a CCSDS header is not included in the CLTU, then no transfer frame is created and the commands that are transmitted are processed without transfer frame encapsulation. This property is used to output arbitrary transfer frames, particularly ones with errors in the header fields, by embedding the binary equivalent of the transfer frame header

in the command data. After each command sequence is processed by the command icon, an "idle pattern" is transmitted that will be repeated continuously until the next command is sent.

A toggle switch function is included in the Command icon's setup screen to enable or disable the generation of CCSDS codeblocks. Normally, after a transfer frame is built from a CCSDS Header and one for more commands, it is broken up into CCSDS codeblocks for transmission. The segmentation of the message into codeblocks may be disabled if raw transfer frames are to be transmitted, or if erroneous codeblocks are to be output for system error testing.

The CCSDS commanding functionality described above is part of AP Labs' VTS system and represents one of its core capabilities. However, due to the open architecture of the VTS system, the user is not limited in any way to this commanding functionality. The user can develope his/her own command and control software, or may purchase one of the Commercial-Off-The-Shelf (COTS) products to support satellite operations. An example of one of the more powerful COTS commanding packages is Storm Integration's Intelligent Mission Toolkit (IMT). This product provides pass planning and command transmission capabilities that augment the telemetry processing capabilities provided by COTS vendors such as AP Labs. The IMT blends other commercially available software products into an open solution which solves the complex commanding requirements of today's satellite programs. Recognizing its functionality and flexibility, the USAF and the Ballistic Missile Defense Organization (formerly, Strategic Defense Initiative Organization) and NASA Goddard have selected the IMT as the commanding component of their new, state-of-the-art, workstation-based command and control systems.

The IMT is an open, modular COTS package which provides the ability to support multiple satellite processing applications with a complete closed-loop commanding and telemetry processing capability. Processes are distributed across an Ethernet or FDDI Local Area Network (LAN) with functionality resident on UNIX-based workstations and VME-based industry standard front-end equipment. The user interface is provided via multiple graphical Human Computer Interfaces (HCI) COTS software packages. The system is configured and driven by a relational database, making it a generic, high fidelity commanding capability.

INTEGRATING RADIO RECEIVERS WITH TELEMETRY & COMMANDING SYSTEMS

The combination of telemetry processing and monitoring with command & control capabilities within the same VME chassis opens up the possibilities for completely

integrated, reduced cost satellite control systems. A further step towards this goal is migrating the radio reception of the satellite downlink channels into the VME chassis where the digital baseband processing takes place. This step allows truly portable, small footprint systems capable of receiving and processing telemetry data to be fielded. These systems are principally used in the ground testing of satellites, both during assembly and later during pre-launch testing where the radio range between satellite and test system is limited. This is a factor because the nature and size of high power microwave transmitters used in the command uplink, do not lend themselves well to integration with other components in a small footprint VMEbus package .

The reception of the microwave downlink channel from a satellite has been successfully implemented within the VMEbus environment by several vendors of telemetry equipment. Further refinements in receiver and diversity combiner design and packaging are currently underway. These efforts will ultimately result in VMEbus-based telemetry receiver/diversity combiner products that will be used to receive and process telemetry data from satellites that have been placed into orbit. The challenge of adding high-power microwave command uplink capabilities to small footprint, reduced cost satellite telemetry and control stations still remains.

SYSTEM REQUIREMENT AND IMPLEMENTATION

The major requirement involving a system that combines telemetry processing with satellite command & control is that the system hardware and software must support in real-time, the generation of commands that are derived in whole or in part from conditions that are revealed in the satellites telemetry data. This implies that the processing of the satellite's telemetry downlink and the transmission of the command uplink is tightly coupled in a closed-loop system where real-time performance is critical (the example system presented in this paper is such a system).

The system in this example is implemented using a totally open hardware and software architecture. This ensures a flexible system that can incorporate a wide range of processor and I/O devices that are available from a large number of sources. With an open architecture, the user can grow and adapt the system to a variety of different mission requirements. By adhering to industry standards for software and network technology, the user can change, add or enhance the systems functionality by creating additional software programs to support new satellite launches.

For the example system presented in this paper, the AP Labs VTS software consists of low level drivers that control the boards in the system and higher level application software which allow setup and control of the telemetry, commanding and I/O boards. AP Labs' Asynchronous I/O (AIO) utilities are utilized in the example system to

provide low overhead access to the telemetry and I/O hardware. The VTS application software consists of real-time VME-based software running under the VxWorks operating system. This software controls the acquisition and processing of the PCM telemetry stream, generation of CCSDS commands and the Unix workstation-based setup and control graphical displays. The setup and control software running on the Unix host communicates with the real-time processor over an industry standard Ethernet link.

AP Labs' VTS software provides the high level control required to operate the system as a fully integrated product. The graphical setup and display portion of the software allows the user to configure and control each real-time task graphically and intuitively from the host workstation, using a series of icons, setup menus and display devices (referred to as, "widgets"). The software architecture for the VTS system is seen in figure 4.

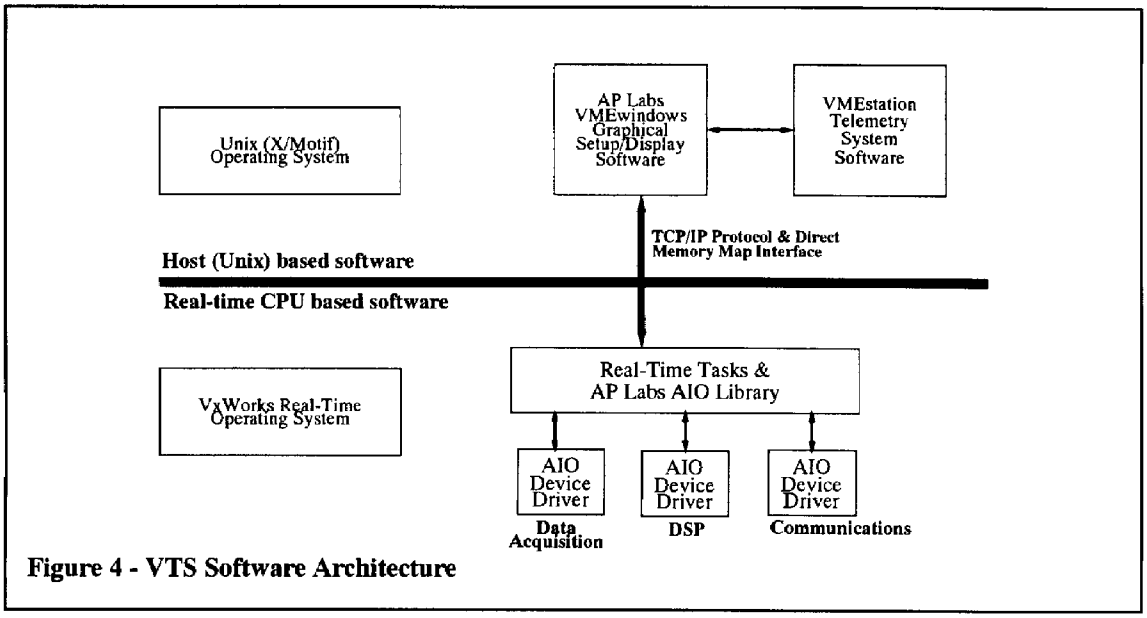

Figure 4 - VTS Software Architecture

The industry standard X-Windows/Motif-based graphical user interface software provides system setup, telemetry and command parameter definition, display definition and real-time display capabilities. It performs error checking on user-input parameters and signals the user when an invalid parameter is entered. Hard copy output of the systems configuration is supported. System configuration allows the user to graphically setup each of the telemetry I/O interfaces, the other I/O interfaces and the network. The parameter definition allows the user to enter/edit parameter processing definitions. The parameter ID and input information as well as the various algorithms to be applied to it are defined here. The display definition allows the user to select from a number of widgets (numeric display, gauge, bar graph, strip chart, X-Y plot, scrolling tabular display) and define the colors, ranges, positioning, size and

format of each output window. Parameters (with their respective out-of-limit values) are associated with the widget to define a display. Displays are grouped to windows and windows are grouped to a configuration. Save, restore and edit capabilities are available for each user-configuration item.

The goal of the VTS graphical user interface in the example system is to show the current state of the real-time side of the system using a block diagram format and to allow the user to easily modify the operation of the real-time system through the manipulation of this diagram. The top level VTS screen contains the data flow diagram shown in figure 5. This diagram is a graphical representation of the flow of data between various processes on the real-time system. This diagram consists of a group of icons that are interconnected to show the desired routing of data. The top level VTS screen also contains the main menu, which contains all of the commands necessary to configure and run the system.

As shown in figure 5, the PCM bit synchronizer in the example system is configured via the BIT SYNC icon and the PCM decommutator (frame synchronizer) is configured via the DECOM icon. All PCM stream definition and processing is defined and configured using the telemetry processing (TM PROC) icon. Double clicking the mouse on any of these icons will bring up the appropriate setup menu(s) for the function selected. When selected, the PCM stream definition setup menu provides an easy-to-use interface for defining major/minor frames, raw PCM measurands, derived parameters, parameter processing.

CONCLUSION

As the design and deployment time lines for military and non-military satellite programs continues to shrink, the need for open architecture, lower cost telemetry and control systems will continue to grow. The development and procurement of proprietary telemetry equipment and components that are available from only one vendor can no longer be justified in today's market place.

The trend for the 21st century and beyond is for less expensive, simpler satellites that can be mass produced and quickly deployed into low earth orbits. This capability will require low cost, flexible, easily reconfigurable telemetry & commanding equipment that can be used and reused to accommodate the fast pace and rapidly changing satellite development environment.

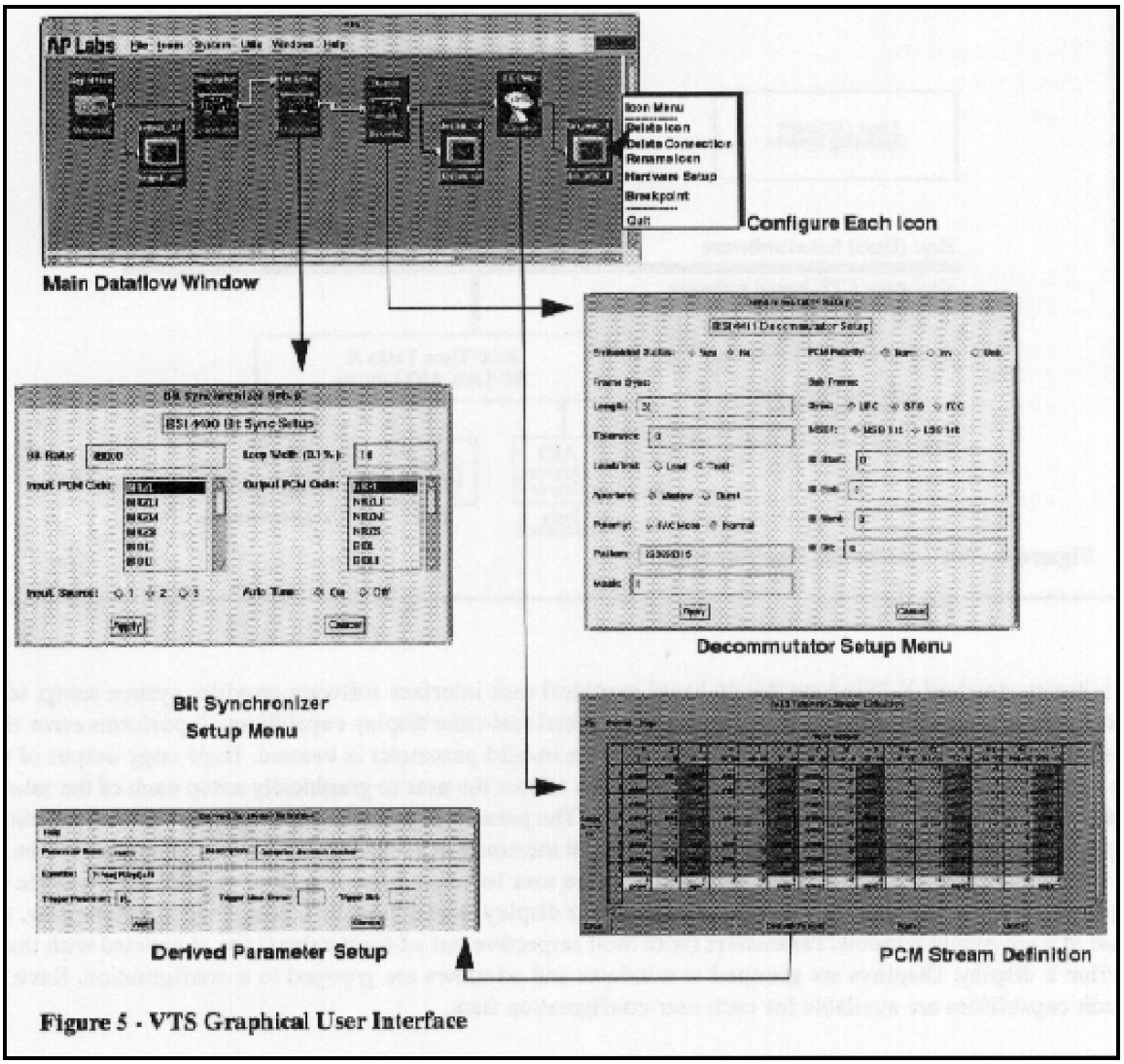

Figure 5 - VTS Graphical User Interface

ACKNOWLEDGEMENTS

The author wishes to thank Mark D. McMillen, David O. Gregory and John D. Finley of AP Labs and John Reeser of Berg Systems International for their assistance in the preparation of this paper. A special thanks also to Ms. Gina De Meo for her help.

REFERENCES

1. Advanced Processing Laboratories, VMEstation Telemetry System User's Guide, San Diego, California, October 1993.

2. Advanced Processing Laboratories, VMEstation Telemetry System CCSDS Command Icon User 's Guide, San Diego, California, April 1994.

3. Advanced Processing Laboratories, VMEstation Telemetry System CCSDS Telemetry Icon User's Guide, San Diego, California, April 1994.

4. McMillen, Mark D., Open Architecture Telemetry Processing Systems, Proceedings, International Telemetry Conference, Las Vegas, NV, October 1993.