A SATELLITE MONITORING AND CONTROL SYSTEM ON AN OPEN PLATFORM

Bharat Patel
Loral Data Systems
Sarasota, Florida

Dave LaVallee
Loral AeroSys
Seabrook, Maryland

ABSTRACT

Schedule and budget constraints have created a need to produce satellite control centers quickly and efficiently. This need drives the development of a reconfigureable satellite control center. This paper describes such a system. The system is built on an open platform. It can be produced quickly and adapted to a specific satellite or satellites easily. It performs the core functions necessary to monitor and control satellites, and is continually adding new functionality. The system's architecture and openness allow upgrading for new requirements or improvements in technology.

BACKGROUND

As governments and private companies re-think their satellite programs, it is evident that shorter development times and lower-cost solutions are necessary to make wise use of tight budgets. Custom solutions, built from scratch, are no longer an option. Although the ground segment of a satellite system represents a smaller portion of the budget than the flight segment, it is an area that is subject to substantial cost savings in both development and operations.

The ground segment incorporates three high-level functions. Telemetry to access the health and safety of the spacecraft and payload is down-linked. Commands are up-linked to control the spacecraft and payload, and to maximize its usefulness. Finally, payload data, such as science data for a NASA mission or communications data for a telephone company, is received and processed. These functions are performed in real-time, 24 hours per day, seven days per week. Figure 1 shows the physical layout of the ground segment and associated data paths for the three functions.

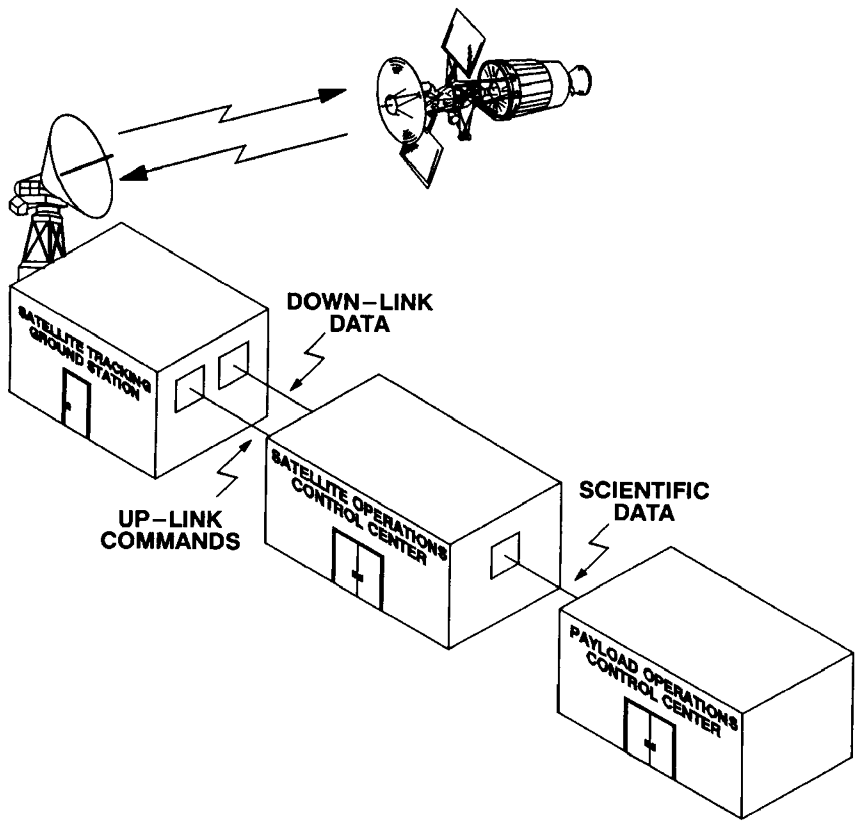

Figure 1. Physical Arrangement of System

An intimate relationship exists between the health monitoring function ("telemetry" as we know it) and the command function. This requires a unified system to perform both these tasks in the Satellite Operations Control Center. This system must also be able to separate and distribute payload data on the appropriate satellite programs, and to perform a monitoring function on that data. This ideal system must be flexible enough to be of a standard design, yet versatile enough to be uniquely configured for any given satellite. It must interface to each of a fleet of satellites, such that major users do not need to procure separate systems for their different craft. It must be made of state-of-the-art hardware and run by state-of-the-art software, yet be upgradable in the years of its life to take advantage of significant improvements in technology.

Such a system has been developed jointly by two divisions of Loral, the Data Systems Division with their telemetry experience and products, and the AeroSys Division with their satellite monitoring and command experience and software. That system is described here.

SYSTEM OVERVIEW

Functional assignments on a typical satellite program are shown in Figure 2. Since the Satellite Tracking Ground Station is often at a remote location with relation to its user

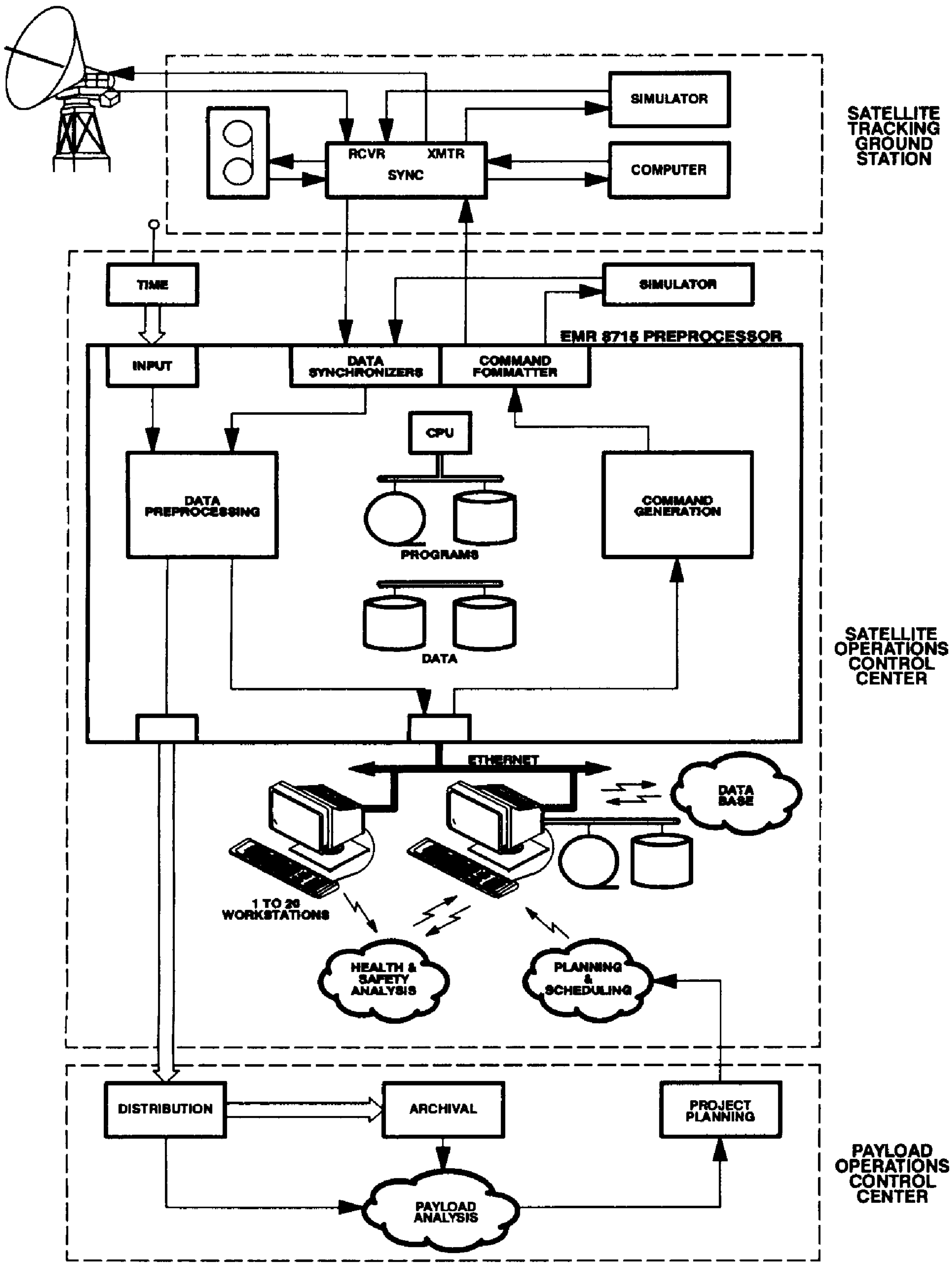

Figure 2. Functional Assignments

community, that Station' s functions are somewhat limited. The antenna is pointed to the satellite and tracks it automatically, bringing down-link data to a radio receiver and signal conditioners/bit synchronizers for reconstruction and output on the network. The information is archived locally also, and possibly the Station includes some display and command generation capability as a backup to the other sites. Commands from the network are formatted and transmitted on the up-link channel frequency.

Most data handling and command generation are centralized at the Operations Control Center, which is described as the focal point of this paper. The center is built around an existing telemetry system design and uses a powerful hardware preprocessor for most merging, processing, archival, and distribution. Each of several workstations has independent processing capability, while one has local ground station control and another has satellite command generation and initiation capability.

The system distributes payload data to a Payload Operations Control Center at the same or another location for detailed analysis.

DESIGN CONCEPTS

The design engineers for the system took advantage of recent developments in telemetry- related products as they planned this center. This started with the open systems concept, where both hardware and software elements are industry-standard and lend themselves to easy upgrades in the expected lifetime of the center.

Hardware openness starts with modular construction of the preprocessor, and interconnection of module setup and diagnostic ports on a VME bus. It includes archival of programs and data on a SCSI or SCSI-2 bus. It also involves distribution of data and commands at high rates on Ethernet, expandable to a fiber FDDI network.

Software openness involves Unix and/or POSIX as the operating system at the preprocessor and at every operator workstation. It extends to the use of industry-standard TCP/IP protocol and UDP for data networking, OSF/Motifstyle-guide-compliant interfaces for operator input, and ANSI SQL-based databases. In addition, all displays are in X-windows for versatility and ease of arrangement and interpretation.

A valuable fallout from this hardware and software openness is the ability to use workstations of several types from several manufacturers, all interfaced in hardware and software so that the seams are invisible.

Another design concept is the use of a distributed system, using a client-server paradigm. Software functionality is distributed among any number of workstations necessary to perform efficient operations. The system can be configured to execute functions on any processor with the necessary resources for that function. Redundancy can be configured automatically. The client-server architecture defines clean interfaces among subsystems. Modifications or upgrades to one subsystem do not affect other subsystems as long as the interface remains constant. This reduces the effects of changes and enhancements.

The core system configuration for the system is the Loral O/S90 Data and Command System. The interface for down-link data and up-link commands is through Loral's EMR 8715 Preprocessor. The operator interfaces are via computer-powered color graphic workstations; the center may have one to twenty from Sun, Digital, IBM, HP, or Silicon Graphics as desired. All interfaces between the preprocessor and workstations are via industry-standard Ethernet, so the system is easily expandable.

In each system, a satellite simulator is installed to receive commands and generate data for system checkout. The complexity of the simulator is subject to definition as system requirements are derived with the user.

The up-link/down-link interfaces between the Center and the satellite communications network reside in the hardware preprocessor. Since this is a hardware device, it is uniquely matched to the satellite or satellites by selection of modules. For example, modules are available for phase-shift-keyed data and/or commands, NASCOM data and/or commands, CCSDS packet data and/or commands, and continuous pulse-code-modulated (PCM) data and/or commands. These can even be mixed in a system to accommodate different types of satellites simultaneously. Data flow is shown in Figure 3, to illustrate the up-link and down-link relationships.

DOWN-LINK PERFORMANCE

As data comes in from the satellite, Loral's hardware preprocessor examines every measurement for limit conditions and archives them to disk. Each individual operator selects the telemetry points that they are interested in monitoring. These are normally predefined displays, but individual telemetry parameters may also be requested. In this way, the system captures, limit checks, converts to engineering units, and archives all data, but only those telemetry points actually being monitored are processed through the network.

All this up-front analysis takes place in one or more distributed-processor modules, built of high-speed, high-accuracy, high-resolution chips and run by efficient

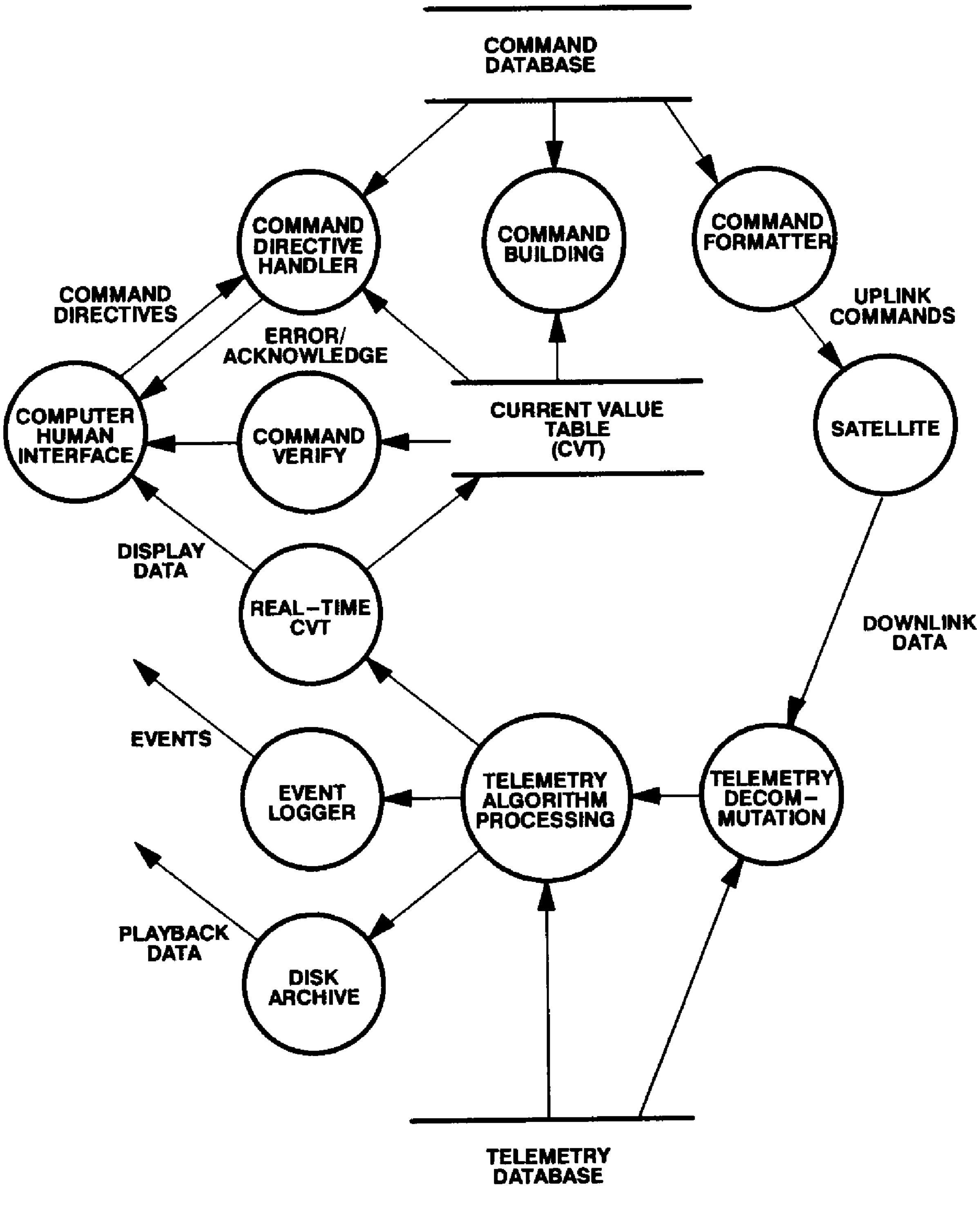

Figure 3. Data and Command Flow

algorithms which are defined by microcoded sequences. Loral or a user can modify the operation if desired to accommodate new processing requirements.

The same processor module converts measurements from binary numbers generated at the satellite to meaningful engineering unit designations such as degrees and meters. It examines each measurement to detect critical conditions, and generates an "event" message with the data to notify the operator of the alert or alarm state. Each event can be broadcast to all workstations, or transmitted to specific operators.

After each point is examined, the data goes to a preassigned location in a current-value table (CVT) in the preprocessor. This CVT then contains, at any given moment, the most recent values of all measurement points on board the satellite. Several times each second, the CVT is selectively broadcast over the network to each workstation which has requested data. Since only the required subset of the entire CVT is transmitted, Ethernet bandwidth is conserved. Each workstation operates independently, so that users can process satellite data differently and display different measurements in different ways simultaneously. Due to the open software standards, users can move from station to station with no difficulty.

A typical real-time display of data has one to four windows. Each can be dedicated to a specific device on the satellite, and displays the condition of that function graphically and also as engineering units for fast and accurate analysis. Alarm and event conditions are prioritized and shown by special colors, and an audible alarm can be generated if the operator wishes. One display is shown in Figure 4.

During the operations phase of a satellite, it frequently becomes necessary to depict the orientation of the satellite relative to external objects (e.g., Sun, Earth, Stars.). A software package called the "Spacecraft Attitude Compass" (SAC) supports determining sun lighting conditions on the satellite's solar panels, determining bright objects (i.e. the Sun) near a sensor's field of view, or determining the visibility of tracking sites.

This software presents spacecraft attitude data graphically. It computes and displays satellite orientation with respect to a wide range of external objects. The speed and accuracy of the SAC display reduces the risk and time it takes for satellite operators to isolate and diagnose anomalous satellite conditions.

The SAC display, Figure 5 consists of four concurrent displays:

- 3 - D dynamic representation of satellite and its sensor' s fields of view.

- Views of external objects (Earth, Stars, Sun ...) as would be seen from a selected sensor or given direction from the satellite.

- Satellite position shown over the Earth map that includes tracking station coverage zones, satellite day/night zones, and the South Atlantic Anomaly.

- Textual summary

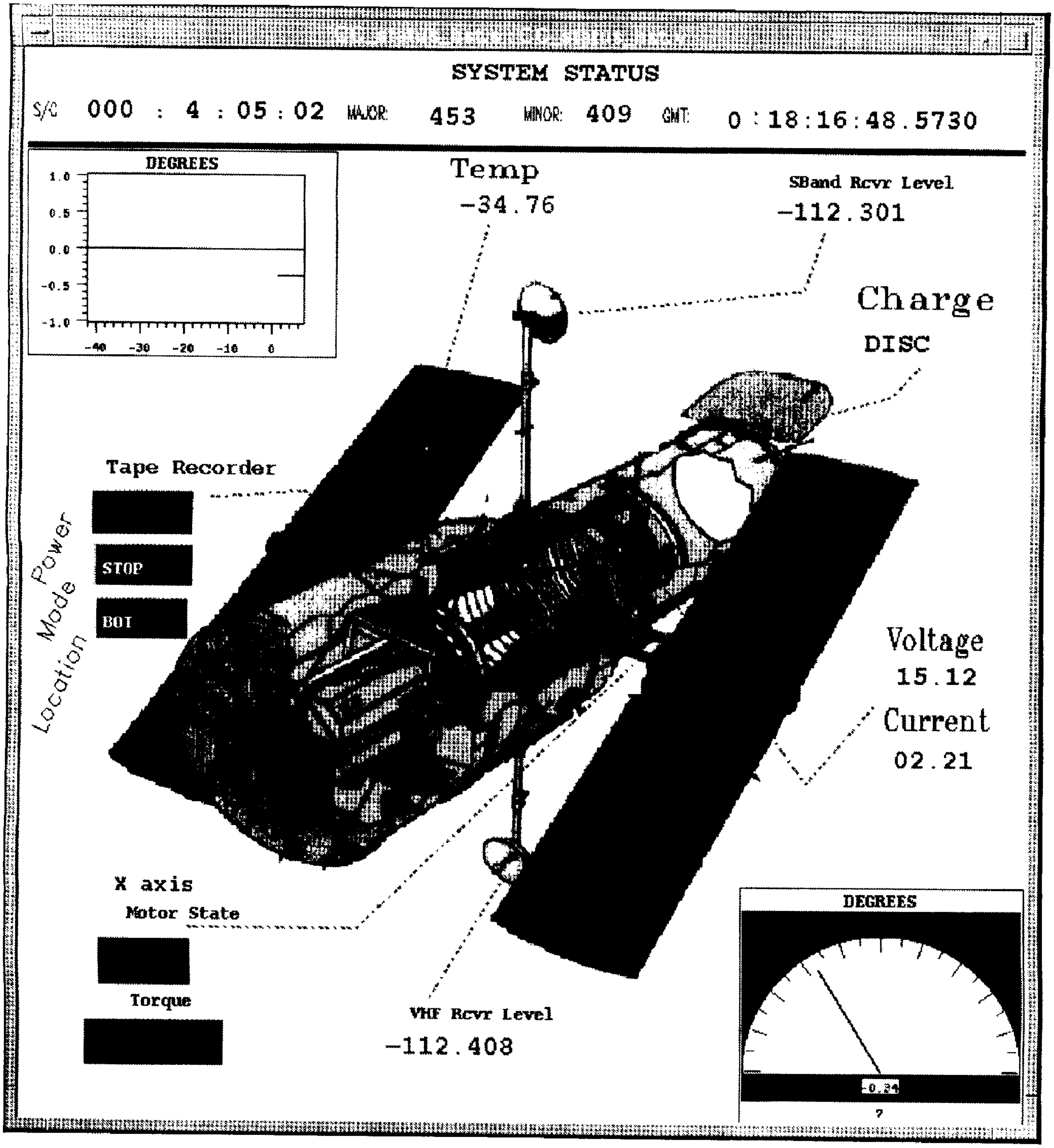

Figure 4. Real-Time Display

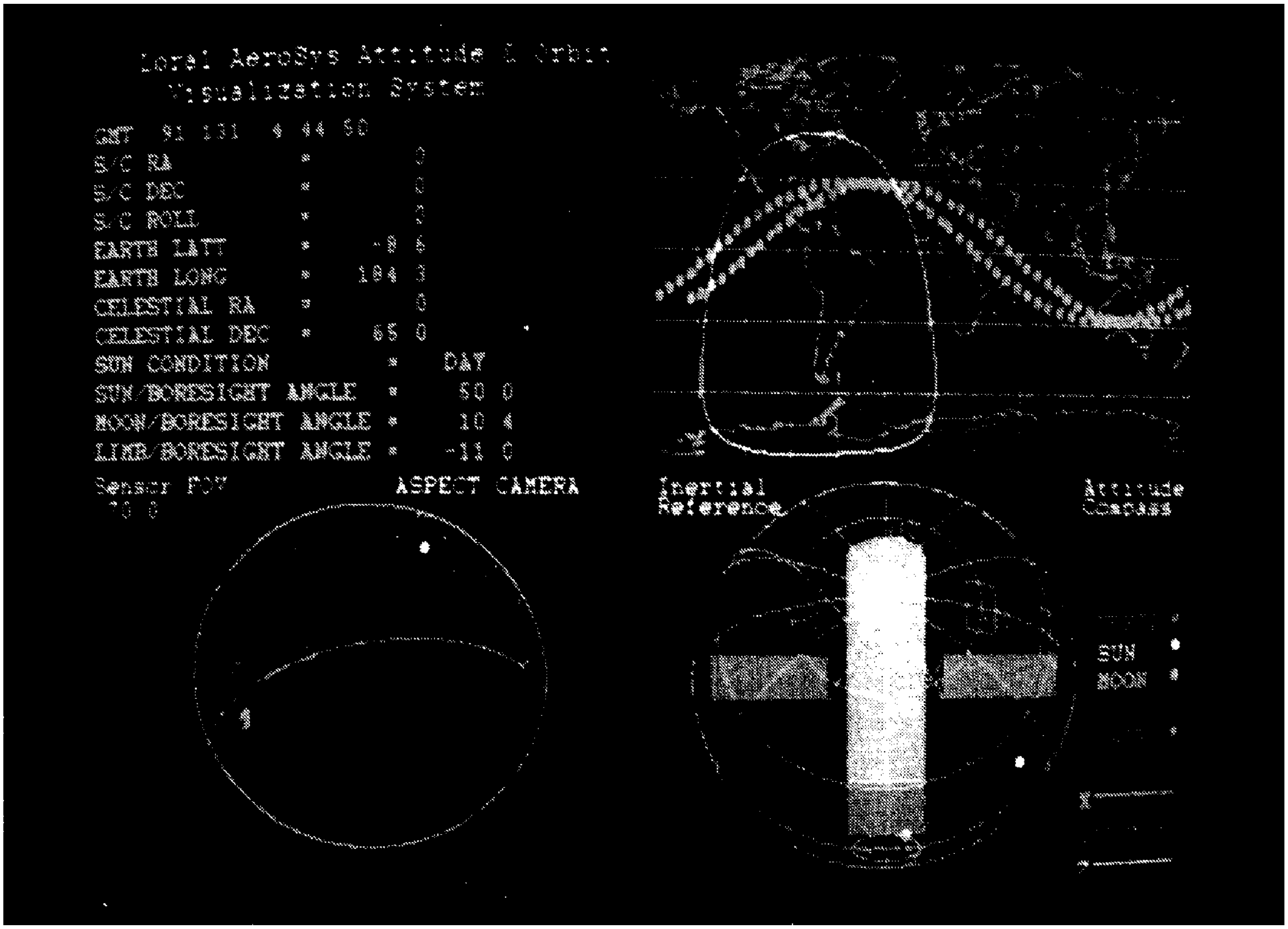

Figure 5. Satellite Attitude Compass

Simultaneously with real-time data display, the preprocessor routes down-link data to a hard disk for storage. The user at a workstation can recall data from an archival disk for display with or instead of real-time information, and/or can build a time-history sequence for trend analysis at a workstation, and/or can copy a hard disk periodically onto a removable cartridge tape for long-term archival. All up-link and down-link operation is implemented by a powerful SQL data base in the main workstation, the"system server". This resource contains all details necessary to set up the hardware in the down-link data path uniquely to handle each measurement which comes from the satellite. Satellite sensors are defined by alphanumeric names, such that any operator at any workstation can call up any measurement by that name for local display. The data base contains units of measurement (as degrees or meters), arguments for conversion of the measurement into these engineering units, and any other information which should be accessible to the system software for processing each satellite measurement.

UP-LINK PERFORMANCE

For up-link commands, the operator has a very detailed software resource. This spans the area from background scheduling through on-line commanding, and allows several priority levels if they are appropriate. The actual command initiation may be mouse-driven, or may be entered at the keyboard through use of Satellite Test and Operations Language (STOL). Further, the operator can pre-predefine command access via soft function keys, so that specific commands or command sequences can be initiated this way. The necessary precautions are taken on critical commands to insure that they are checked before transmission. A workstation position is predefined at log in to be the control point. This is the only location from which commands can be issued, so that configuration control of the satellite and ground system can be maintained. Commands can be issued in real-time and from a preplanned command schedule. In addition, all system responses to operator inputs such as command acceptance or rejection, counter verification, prerequisite transmission, and load verification are logged and displayed simultaneously in the "event" window. All commands are logged, such that the system maintains a complete history of all commanding activity.

Developing optimal or near-optimal schedules in a modern spacecraft control system is a difficult task. The problem is further complicated by the need to modify existing schedules in near-real time because of unexpected occurrences. A solution to this problem is found in the software called the "Request-Oriented Scheduling Engine" (ROSE), a resource-allocation scheduler package that supports each phase of the scheduling process. ROSE can schedule generic requests, ground system and spacecraft resources, and instrument-specific view periods as shown in Figure 6. ROSE provides batch scheduling, with various resource request selection and placement heuristics. It also supports interactive scheduling by providing the operator with information on resource conflicts, resource usage, and graphical displays. With the aid of efficient search techniques, ROSE supports intelligent incremental and reactive scheduling of user requests.

The command function includes both command validation and command verification. Command validation (before the command is actually sent to the spacecraft) is accomplished through syntax checks, data base lookups, prerequisite state checking, and transmitter probe verification. Command verification, on the other hand, confirms that commands are transmitted to the spacecraft and executed as expected. There are four common verification techniques: "command counter comparison," "end-action verification," "telemetry and command echo comparisons;" and "ground verification;" each is further defined here:

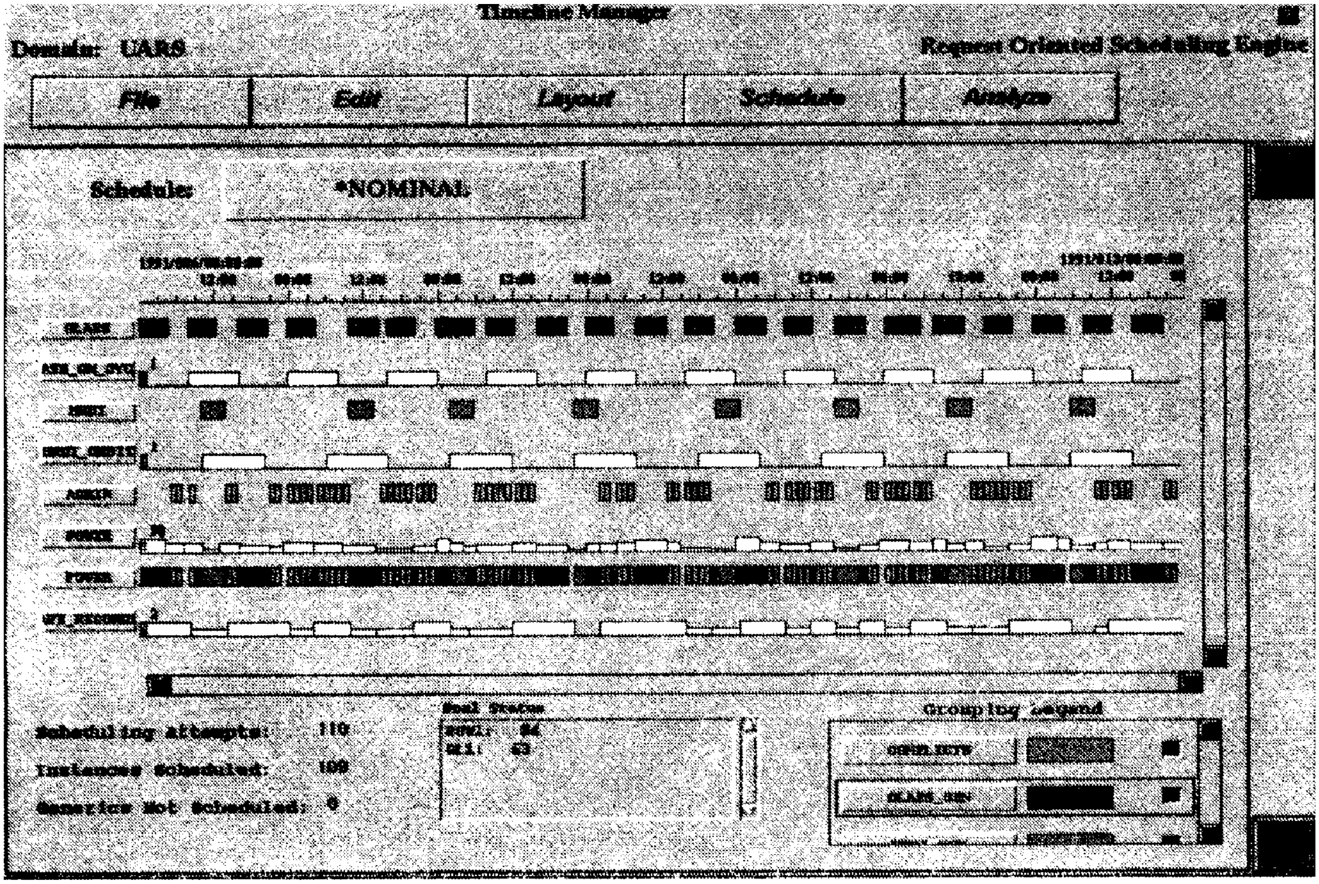

Figure 6. Request-Oriented Scheduling Engine

- Command counter comparisons require ground and spacecraft counters reflecting the number of commands transmitted and received. Transmitting a command updates the ground counter. When the spacecraft receives the command, the on board counter is updated. After waiting for the maximum propagation delay, the counters are automatically compared.

- End-action verification monitors the telemetry stream, looking for the appropriate change of state in the satellite based on the specific command.

- Command echoes can be generated by the spacecraft or ground station to show command receipt.

- Ground verification is a two-step process. A command is transmitted to the spacecraft and loaded on-board but not executed until the load verification is confirmed on the ground and an "execute" command is received by the spacecraft.

Information such as spacecraft software, ephemeris, and stored commands is loaded and verified in a similar manner, with the entire load being sent to the satellite and then confirmed with a memory dump from the satellite. Stored commands for low-earth orbiting satellites are often executed while the spacecraft is out of contact with the ground station. This requires that end-action verification occurs once contact is reestablished, and the predicted state of the spacecraft is compared with the actual downlink.

In a similar manner to the down-link architecture, the up-link data base contains all appropriate details relating to composition and validation of up-link commands. This software is integrated into the main system to generate menus, respond to operator inputs, and provide information to the operating software as it is needed.

SUMMARY

The system described here integrates the command and monitoring functions of a satellite ground station into a generic design which can be configured uniquely for each specific type satellite in use. It is built with industry-standard hardware and software modules, and can be upgraded during its lifetime to take advantage of advances in the state of the art.