Earth Observing System Data and Information System (EOSDIS) Test System (ETS) Detailed Design Specification Volume 2 Multimode Portable Simulator (MPS)

March 1996

Prepared Under Contract NAS5-31500
Task Assignment 15 507
By Computer Sciences Corporation
515-DDS-002

Prepared by:

J. McGarry Date
Task Leader, CSC

Concurred by:

W. Fuller Date
ETS Project Manager, GSFC, Code 515.1

Reviewed by:

M. Heffernan Date
Manager, Simulation Systems Department, CSC

Concurred by:

D. Lakins Date
ESDIS ETS Manager, GSFC, Code 505

Test Systems Branch
Goddard Space Flight Center
Greenbelt, Maryland

Preface

This document is volume 2 of a 5 volume set specifying the detailed design specifications for the Earth Observing System Data and Information System (EOSDIS) Test System (ETS). This volume describes the detailed design for the Multimode Portable Simulator (MPS).

Abstract

The Earth Observing System Data and Information System (EOSDIS) Test System (ETS) provides test capabilities for the ground system supporting Earth-observing spacecraft that conform to the Consultative Committee for Space Data Systems (CCSDS) Recommendations. ETS is designed to simulate and test EOSDIS ground systems and element interfaces. The simulation and testing capabilities cover only level-zero processing, including data delivery to the Distributed Active Archive Centers (DAACs).

This document provides the detailed design specifications for the ETS Multimode Portable Simulator Release 1.0. It includes an overview of each subsystem and a high-level description of each software task or module. Traceability matrices contained in the appendices provide cross references between MPS detailed requirements and software elements, and MPS detailed requirements and planned builds.

The Multimode Portable Simulator (MPS) is a low-fidelity EOS AM-1 spacecraft simulator. It will generate non-science telemetry formats and will receive and verify spacecraft commands. The MPS will operate in three modes: serial bitstream interface with EDOS, Nascom block interface via the Earth Orbiting System Data and Information (EOSDIS) Backbone network (EBnet to EDOS) and a networked interface with EOC (simulating EDOS formats).

Keywords: EOSDIS, ETS, MPS, simulation, design

Contents

Preface

Abstract

Section 1. Introduction

List of Figures

Section 1. Introduction

This document is Volume 2 of a five-volume set specifying the detailed design specifications for the Earth Observing System Data and Information System (EOSDIS) Test System (ETS) This volume describes the detailed design for Release 1 of the Multimode Portable Simulator (MPS). A guide to the other ETS design documents is provided in Volume 1.

1.1 Purpose and Scope

The Earth Observing System Data and Information System (EOSDIS) Test System's (ETS) Multimode Portable Simulator (MPS) software system is being developed by Computer Sciences Corporation (CSC) for the National Aeronautics and Space Administration (NASA) Goddard Space Flight Center (GSFC). The MPS system will provide low-fidelity spacecraft simulation to support EDOS and EOC testing, and ground system software checkout. The MPS is a test tool that will support the verification of EDOS processing of low-rate EOS AM-1 forward and return link data.

This detailed design specification (DDS) document establishes a formal understanding of the MPS detailed software and hardware design. This document defines the formats and contents of data flowing between the EDOS and MPS functions and the EOC and MPS functions. The system user's guide will further document the capabilities of the MPS.

The designs detailed in this document were developed from preliminary designs contained in the handout at the ETS Design Review presentation held November 30, 1995 (Reference 1) and detailed designs contained in the X-Ray Timing Explorer (XTE) Test and Training Simulator (XTTS) Detailed Design Specification (Reference 2).

1.2 Applicable Documents

The following documents were used to develop the MPS detailed design:

1. Presentation handout at ETS Design Review held November 30, 1995
2. NASA, 510-4DDS/0193 (CSC/SD-93/6090), *X-Ray Timing Explorer (XTE) Mission Operations Center (MOC) Detailed Design Specification*, March 1994

The following documents provide background information relevant to the MPS:

3. Computer Sciences Corporation (CSC), (CSC/515-SRS-002*), Earth Observing System (EOS) Data and Information System (EOSDIS) Test System (ETS) Requirements Specification, Volume 2 Multimode Portable Simulator (MPS)*, October 1995
4. National Aeronautics and Space Administration (NASA), Goddard Space Flight Center (GSFC), 515-4FRD/0294 (CSC/TR-94/6084), *Earth Observing System (EOS) Data and Information System (EOSDIS) Test System (ETS) Functional and Performance Requirements*, October 1995

5. --, 515-4SDD/0195 (CSC/SD-95/6025), *Earth Observing System Data and Information System (EOSDIS) Test System (ETS) System Design Specification*, May 1995

6. --, 510-4SRD/0193 (CSC/TR-93/6028), *X-Ray Timing Explorer (XTE) System Requirement Specification*, October 1993

7. --, *Interface Control Document (ICD) Data Format Control Book for EOS-AM Spacecraft (ICD-106)*, April 1994

8. --, *EOS AM-1 Detailed Mission Requirements*, Interim Draft Release, January 1996

9. --, DSTL-92-007, *Human-Computer Interface Guidelines*, August,1992

10. TRW, 510-ICD-EDOS/EGS, *Interface Control Document Between the Earth Observing System (EOS) Data and Operations System (EDOS) and the EOS Ground System (EGS)*, January 1996

11. Open Software Foundation (OSF), *OSF/Motif Style Guide*, Revision 1.2, 1993

12. Consultative Committee for Space Data Systems (CCSDS), CCSDS 301.0-B-2, *Recommendation for Space Data System Standards. Time Code*, Blue Book, Issue-2, April 1990

13. Hughes Information Technology Corporation, 209-CD-004-002, *Data Format Control Document for the EOS AM-1 Project Data Base,* October 1995

14. CSC, *Earth Observing System Data and Information System (EOSDIS) Test System (ETS) Integration Test Plan/Procedures*, Review, February 1996

15. NASA, 515-2STP/0195 (CSC/TR-95/6009), *Earth Observing System Data and Information System (EOSDIS) Test System (ETS) System Test Plan/Procedures*, Review, February 1996

16. ATSC, *Earth Observing System Data and Information System (EOSDIS) Test System (ETS) Acceptance Test Plan,* Draft, February 1996

1.3 Document Organization

This document, Volume 2 of the ETS detailed design specifications, contains four sections and three appendices. Section 1 provides an introduction to this document and lists the applicable documents.

Section 2 provides a design overview and introduces the software and hardware architecture. It identifies the software configuration items that comprise the MPS and whose detailed design is presented in this document. This section also addresses the development environment and development approach.

Section 3 presents the detailed design for the MPS subsystems. It contains the design for the user interface configuration item (CI), spacecraft simulation CI, MPS Utilities CI, and the MPS Operations Management Data (OMD) Simulator (OMDSIM) CI.

Section 4 identifies the status of design issues presented at the critical design walk-through conducted on 2/2/96.

Appendix A contains a traceability matrix of the level four requirements to the design elements contained in this volume and a mapping to the MPS build that will contain the required capability.

Appendix B describes the data files created and utilized by the MPS.

Appendix C contains a list of abbreviations and acronyms.

Section 2. Design Overview

2.1 Requirements Overview

Volume 2 of the ETS Requirements Specification (Reference 3) details both the explicit and derived software requirements for MPS Release 1. The explicit requirements were extracted from existing documentation, such as the ETS Functional and Performance Requirements (F&PR) Document (Reference 4) and the X-Ray Timing Explorer (XTE) Test and Training Simulator (XTTS) System Requirements Specification (Reference 6). The derived requirements were identified during software requirements analysis and are based on the baseline portable spacecraft simulator (PSS) reference architecture, implicit interface requirements, and traditional simulator functionality built into Code 515 developed PSSs.

2.2 Methodology

MPS software development will follow the CSC SEAS System Development Methodology (SSDM) as adapted by CSC Code 515 personnel for Simulations Operations Center (SOC) spacecraft simulator development.

2.2.1 Design

The MPS design is based on that of the XTE Test and Training Simulator (XTTS), as described in the XTE MOC Detailed Design Specification (Reference 2). The data driven, independent tasking design of the XTTS was used as a baseline and then expanded upon. The design of added functionality followed the structured design approach exhibited within the XTTS.

2.2.2 Coding

Each unit will have a prologue giving the name, purpose, arguments, return values, assumptions and constraints, abnormal termination conditions, and development history of each unit. The source code, written in ANSI standard C Programming Language, will be laid out according to standard SOC conventions with regard to layout and indentation. Program Design Language (PDL) and ample comments will be required. Each unit will be subject to peer review to ensure that the code in the unit performs all necessary functions, is free of obvious errors, and follows all applicable standards.

The user interface widgets (windows and window contents) will be generated using the Integrated Computer Solutions, Inc. (ICS) BuilderXcessory graphical user interface (GUI) builder tool. BuilderXcessory generates user interface language (UIL) which can be read and implemented by C language functions; or it can be translated into C language source code. The callback functions for the user interface will also be written in the ANSI standard C Programming Language.

2.2.3 Testing

The MPS will be unit, integration, and system-level tested to assure that the MPS satisfies the detailed requirements found in Volume 2 of the ETS Requirements Specification (Reference 3). These test activities will be completed prior to delivery of the system to NASA for acceptance testing.

Unit testing will be performed by the MPS developers on all new units and on existing units which require more than 25% modification (See Section 2.8 Development Approach).

Integration testing will be performed by an independent CSC integration test team. Plans for performing integration testing of MPS Release 1, Builds 1 and 2, can be found in the ETS Integration Test Plan (Reference 14).

System testing will be performed by an independent CSC system test team. Plans for performing system testing of MPS Release 1, Builds 1 and 2, can be found in the ETS System Test Plan/Procedures (Reference 15).

Acceptance testing will be performed by an independent AlliedSignal Technical Services Corporation (ATSC) acceptance test team. Plans for performing acceptance testing of MPS Release 1 can be found in the ETS Acceptance Test Plan (Reference 16).

2.3 Design Approach

The design of the MPS displays and processes were primarily based on reusing the XTE MOC Detailed Design Specification (Reference 2). The organization and presentation of information within the MPS displays were initially based on the user interface of the XTE Test and Training Simulator (XTTS). The MPS tasks which accept user options and perform the spacecraft simulation were also initially based on the corresponding subsystems within the XTTS.

The GUI and simulation software were then modified to support the MPS level 4 software requirements and conform to Motif Style guidelines.

2.4 Structure Chart Definitions

Structure charts (diagrams illustrating the control hierarchy of the modules or units composing a software item) depict units, data and control couples, and control flow. Figure 2-1 shows the symbols and conventions used in the structure charts. The charts were drawn using VISIO, version 2.0, by Shapeware Corporation.

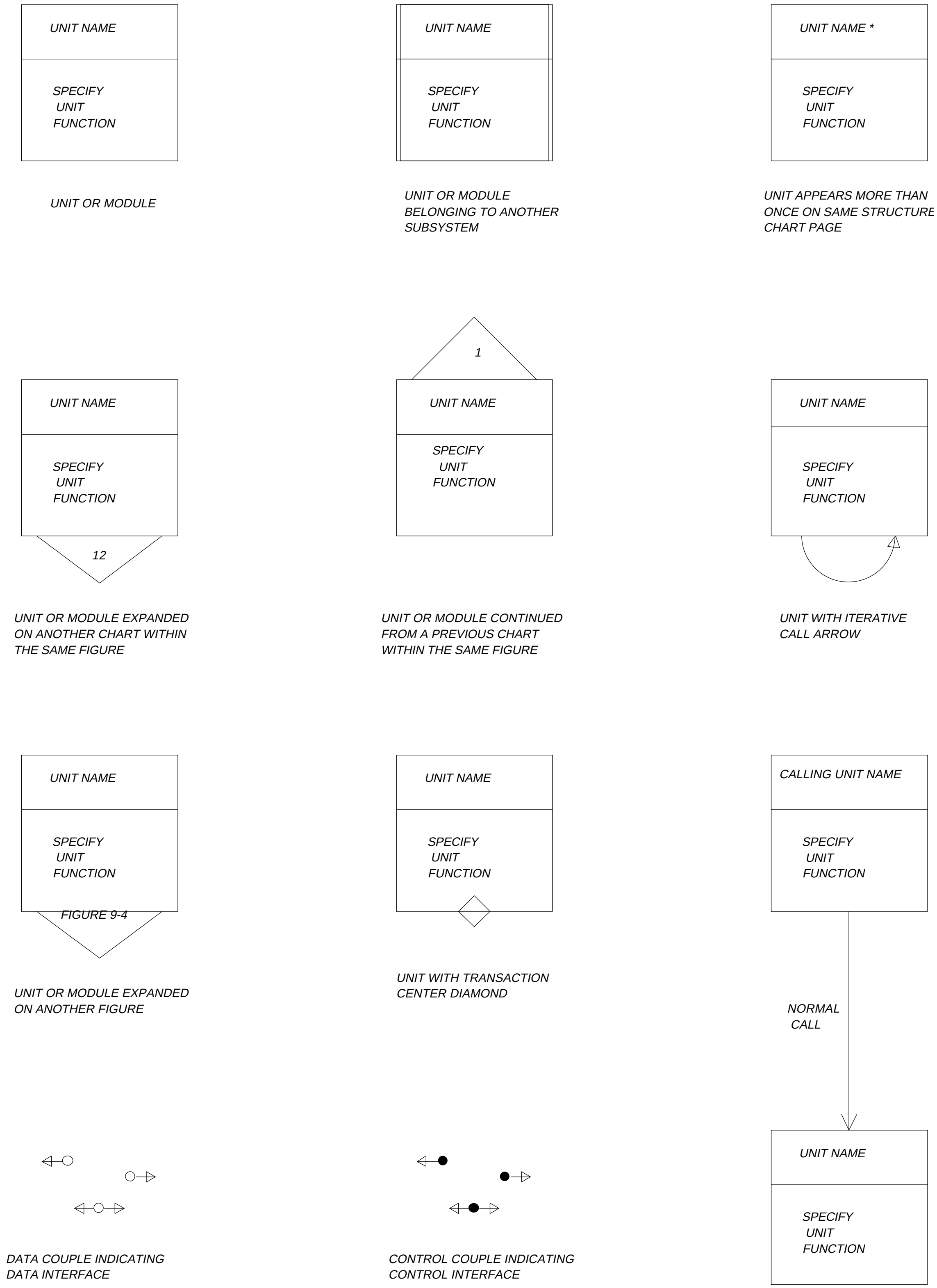

Figure 2-1. Structure Chart Conventions

Each of the boxes on a structure chart is a module, a task, or a unit. Each module or unit has a name, given at the top of the box, and a description, given in the lower part of the box. Double bars on each side of a unit indicate that the unit is either a library unit or belongs to a different task from the one being described on the chart.

Tasks are represented by boxes containing the name of the executive unit of that task. The executive unit always has a name of the form xx_main, where "xx" is the two-character abbreviation of the task name.

Connections between units are shown as arrows originating at calling units and terminating at called units.

A normal C function call is represented by a solid arrow connection. The initiation or termination of a task is represented by a dotted or dashed arrow connection.

A triangle at the bottom of a box shows that the unit is broken down further on another chart. A number in the triangle indicates the page number within the same figure where that routine is expanded. A figure number in the triangle indicates the figure in which further expansion of that unit is illustrated. As a general rule, the breakdown of units in subsequent charts follows a sequential pattern.

Parameters can be passed into and out of the routines as arguments in a function call. The passing of data parameters is indicated by a small, labeled data-transfer arrow with a circle on the end. An open circle indicates a data couple parameter, and a solid circle indicates a control couple.

A curved arrow enveloping the lines going into a group of routines indicates that this group of routines is called iteratively whenever the calling unit executes. A hollow diamond indicates a transaction center, which shows that only one of the routines is used when the calling unit executes. The diamond and the arrow shown together indicates that, for each iteration of the group, only one of the routines of the group is called.

An asterisk after the name at the top of a box indicates that the unit is repeated and appears more than once on the same chart page. This may be used to indicate recursion.

2.5 Hardware Architecture

The MPS consists of a portable front end component and a simulator platform. The MPS front end component will consist of a notebook personal computer (PC) networked with an external component contained in the simulator platform chassis. The front end is composed of: a Pentium-based notebook PC running at 75 MHz with 16 MB of memory, 520 MB removable hard drive, 640x840 active matrix color display, serial and parallel ports, mouse pointing device, and an Ethernet (thin net) card. The external component contained in the simulator platform chassis consists of a Motorola RISC single board VME-based computer (an MVME 187-003B card hosting a MC68040 microprocessor running at 25 MHz with 16 MB memory), TVME712-E transition module/Ethernet transceiver, 1 GB hard drive, and streamer tape drive.

The simulator platform will consist of a collection of 6U cards which occupy single slots within a VME chassis. The simulator platform is composed of a Motorola, MVME 167-34B card (hosting a MC68040 microprocessor and an Ethernet controller), four AVTEC, High-Speed Input/Output (HSIO) cards, one PC Synchronizable Time Generator card, a 64MB DRAM, a 1 GB Hard Disk Drive, a 3.5" 1.44MB Floppy Disk Drive, a 8mm Cartridge Tape, and, a bubble jet printer.

2.6 Software Architecture

MPS is divided into four software configuration items (SWCIs), which include the MPS User Interface SWCI, MPS Utilities SWCI, Spacecraft Simulation SWCI, and Operations Management Data Simulator (OMDSIM) SWCI. The Spacecraft Simulation SWCI is further divided into five subsystems, which include the Initialization Subsystem, Config(uration) Support Subsystem, Telecommand Subsystem, Telemetry Subsystem, and Modeling Subsystem. The five subsystems of the Spacecraft Simulation SWCI run on the VME main processor, and the remaining three SWCIs run on the VME-based computer which is part of the MPS front end. Figure 2-2 depicts the software architecture as mapped to the hardware. Figure 2-3 is a hierarchical breakdown of the MPS into SWCIs and subsystems. Figure 2-4 is a hierarchical breakdown of the real-time system (running on the VME main processor) down to the task level.

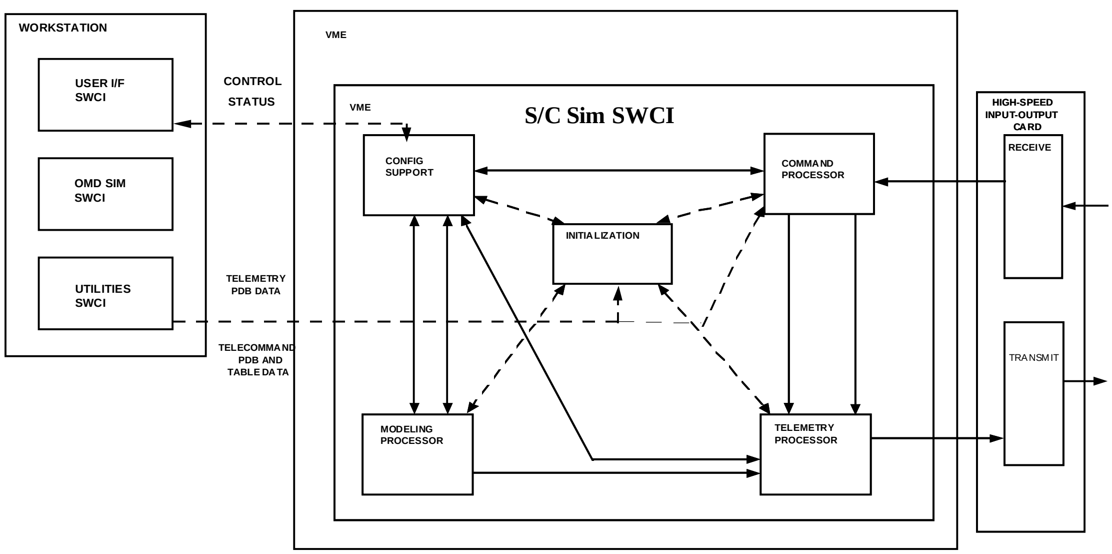

Figure 2-2. Software Architecture

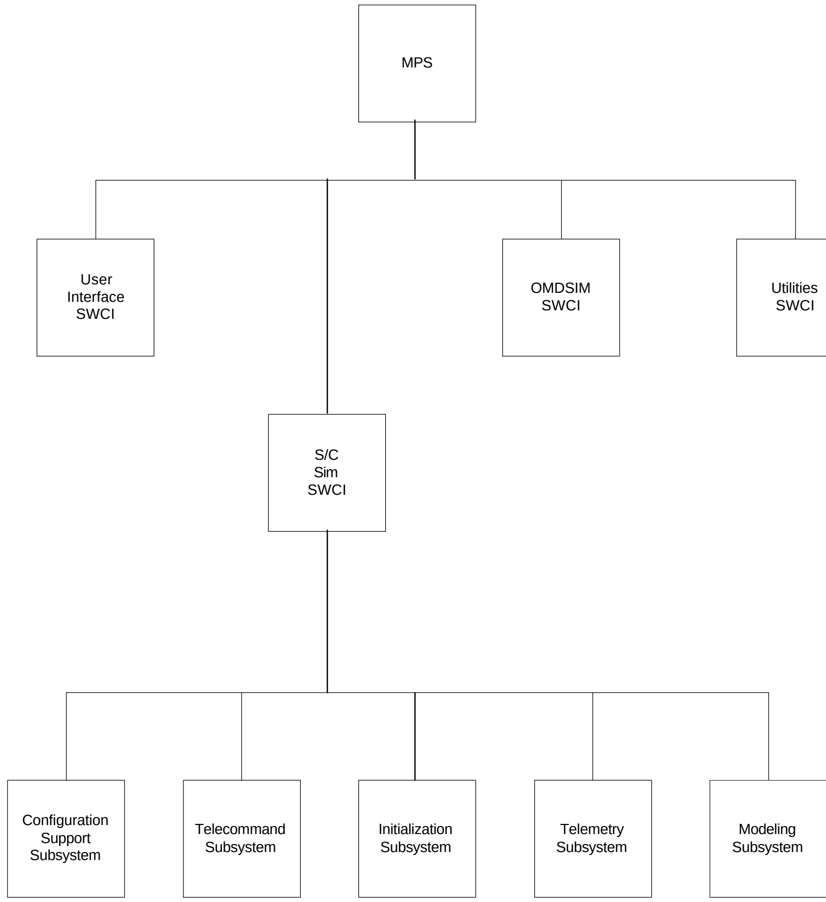

Figure 2-3. SWCI/Subsystem Breakdown

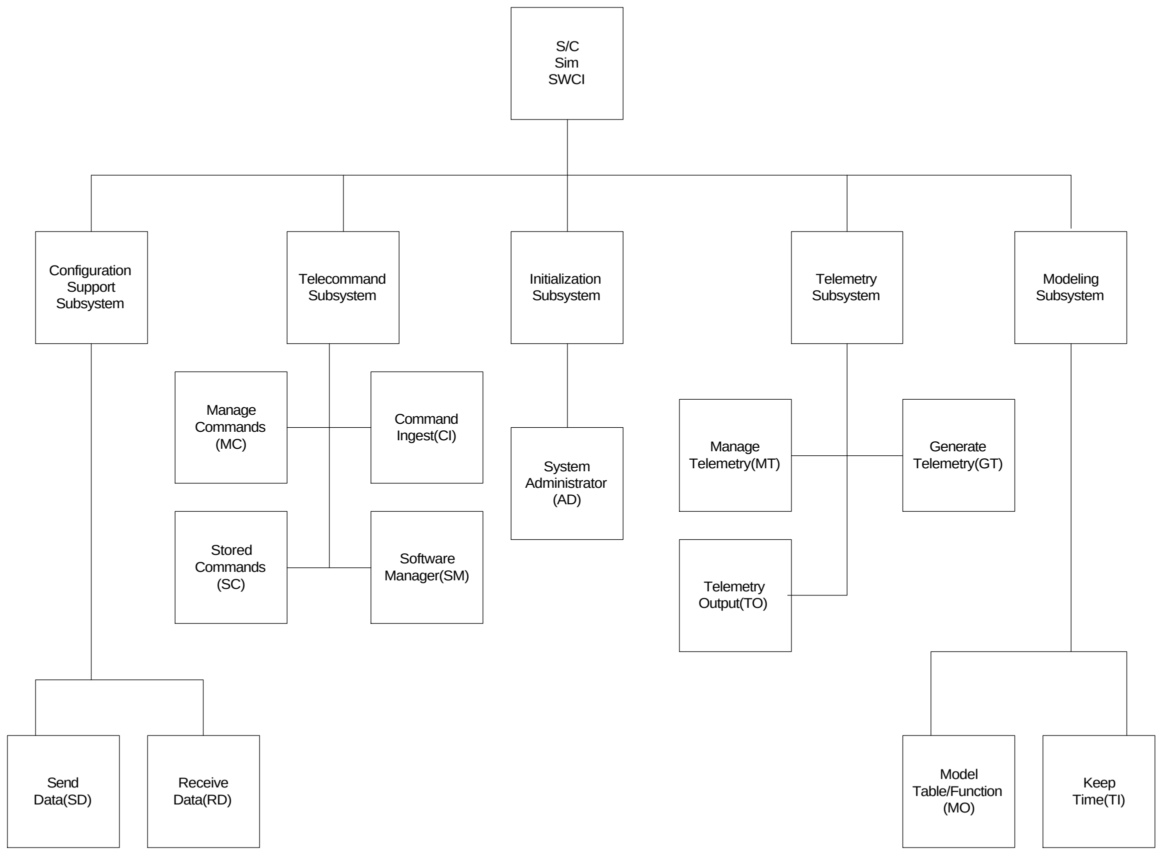

Figure 2-4. Real-Time System Task Breakdown

2.6.1 MPS User Interface SWCI

The MPS User Interface SWCI provides the operator with the ability to configure for telemetry generation and transmission and command reception and processing. The User Interface SWCI formats and forwards operator directives to the Configuration Support Subsystem and receives and displays data sent from the Configuration Support Subsystem.

2.6.2 MPS Utilities SWCI

The MPS Utilities SWCI converts the ASCII-formatted project database (PDB) files into a binary format. The binary PDB files are stored on the VME and serve as input to the Initialization Subsystem.

2.6.3 OMDSIM SWCI

Volume 1 of the ETS DDS provides a detailed description of the OMDSIM SWCI which is being developed for the ETS Low Rate System (LRS). The LRS OMDSIM will be rehosted on the VME-based computer which is part of the MPS front end. A software interface from the MPS to the OMDSIM will provide for an automated start/stop of Customer Operations Data Accounting (CODA) simulation messages coordinated with the start and stop of real-time telemetry transmission.

2.6.4 Spacecraft Simulation SWCI

2.6.4.1 Initialization Subsystem

The Initialization Subsystem brings up the VME software and sees that each of the tasks in the various subsystems completes its initialization successfully. Each task does its own initialization as well, but this is covered under the specific task. The Initialization Subsystem also transfers the binary PDB files to VME memory and informs the Telemetry and Telecommand Subsystems of their location.

2.6.4.2 Configuration Support Subsystem

The Configuration Support Subsystem receives directives from the User Interface SWCI, forwards configuration directives to the corresponding subsystem (Modeling, Telemetry, or Telecommand) and routes requested data packets back to the User Interface SWCI.

2.6.4.3 Telecommand Subsystem

The Telecommand Subsystem (also referred to as the Command Subsystem) receives, validates, and parses the incoming command stream and distributes valid commands to the appropriate task for execution. It also processes memory loads and dumps and uses the information contained in the memory loads to emulate the execution of stored commands.

2.6.4.4 Telemetry Subsystem

The Telemetry Subsystem reads in the telemetry information derived from the PDB and formats and sends real-time and playback telemetry accordingly.

2.6.4.5 Modeling Subsystem

The Modeling Subsystem provides for time driven scenario files and algorithm driven modeling of parameters.

2.7 Development Environment

The MPS development environment encompasses two independent environments. The MPS GUI development and the Spacecraft Simulation development environments are described below.

The MPS User Interface and Utilities SWCI will be developed using the BuilderXcessory development tool. User interface objects will be generated by BuilderXcessory in User Interface Language (UIL) with callbacks used to access application code based on user initiated events. Development personnel will use X-terminals to access an HP Apollo Series 700 Model 755 workstation running the HP-UX operating system. The UIL objects will be compiled, linked and downloaded to the MVME 187 card which will host Unix System V operating system and an ethernet interface. Testing of the User Interface's functionality will be performed through the verification of the MPS Spacecraft Simulation SWCI response to directives and requests for data.

MVME 187 Communication software will also be developed using the HP. These tasks will be written in the C programming language and integrated within the MPS User Interface and Utilities SWCI.

The MPS Spacecraft Simulation SWCI will be developed using an existing operational spacecraft simulator as a building block. The XTTS software components will be loaded onto a PC based network. Software modifications, cross-compilations, and linking will be performed from any networked PC. The resulting executables will then be installed on the MVME 167 card which hosts the PDOS operating system and an ethernet connection.

MVME 167 Communication software will also be developed using the PCs. These tasks will be written in the C programming language and integrated within the MPS Spacecraft Simulation SWCI.

2.8 Development Approach

The MPS will be developed over two builds. The first build will support the AM-1 spacecraft command and telemetry simulation. The second build will support an EDOS simulation using the same command and telemetry data. The MPS software development effort supporting each build will be partitioned into three separate efforts. The development of the MVME based configuration items, the HP based configuration items, and the communication subsystem will proceed in parallel.

The development effort of the Spacecraft Simulation SWCI will actually be a software upgrade to the baseline (XTTS) CCSDS spacecraft simulator's software. As enhancements to the baseline tasks are completed, the updated task will be integrated within the baseline and tested using a PC-based user interface which drives the baseline system. Unit testing will only be performed on units which require more than a twenty-five percent modification. In this manner, the development team will always have access to a functional CCSDS simulator.

The communication tasks will be developed for both the MVME 187/Unix card and the MVME 167/PDOS card. There will be two TCP/IP socket connections between the cards. One socket will connect the MVME 187 based client to the MVME 167-based server and the other will connect the MVME 167 based client to the MVME 187-based server. This system will be tested using a GUI and spacecraft simulation prototype which will verify communication between the elements.

Both the MPS Spacecraft Simulation and User Interface components will then be updated with appropriate hooks to the communication package and tested. Once verified, the user interface will transition from a PC to that of the MVME 187/Unix based GUI.

Section 3. Subsystem Descriptions

3.1 Overview

The MPS subsystem description reviews the design of the four software configuration items (SWCIs) that comprise the MPS. The design of the MPS is extensively based on the reuse of software components from the SOC baseline libraries. The design of the MPS User Interface, Spacecraft Simulation, and Utilities SWCIs were all based on the XTE Test and Training Simulator. The OMDSIM will be reused from the ETS Low Rate System's OMD Simulator. A description of each of the MPS SWCIs is based on reuse from the corresponding areas.

3.2 User Interface SWCI Design

The MPS User Interface is created in the main MPS program that executes on the MVME 187/Unix single board computer. It communicates with the MPS VME subsystems via sockets sending operator input to the VME-based configuration support subsystem which then distributes the information to the telemetry or telecommand subsystem. The configuration support subsystem returns monitoring information to the user interface which displays the requested information on the correct monitor display.

3.2.1 Conventions

Conventions used in this document for the MPS User Interface window features are detailed in Section 1.5, volume 1 of the ETS detailed design specifications.

3.2.2 Display

3.2.2.1 Main Window

The main window of the MPS, shown in Figure 3-1, appears when the MPS is invoked. It contains the primary controls of the MPS system. The work area (top panel) of the window cannot be resized. The window's message area can be resized to a minimum of three lines or a maximum that fills the screen.

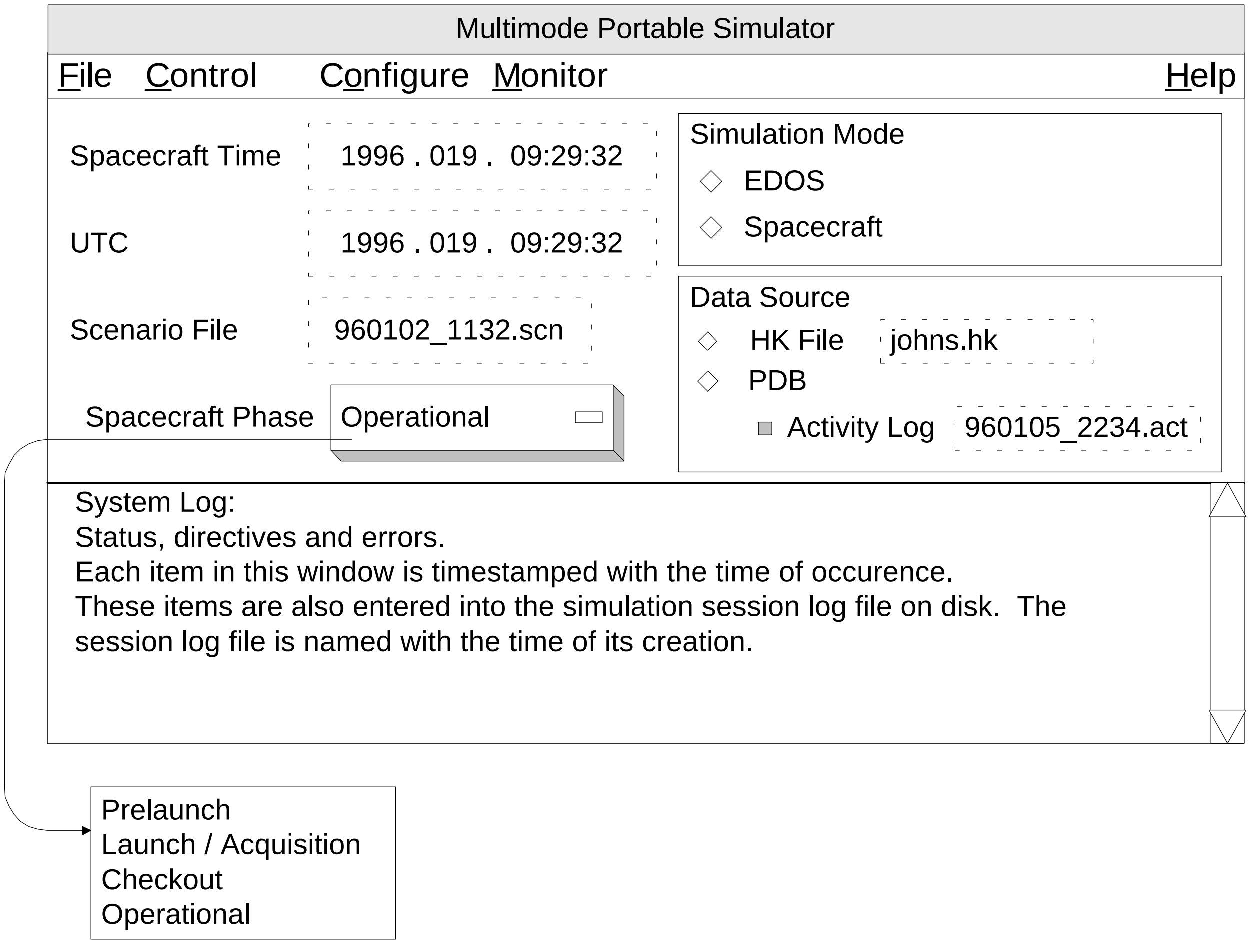

Figure 3-1. MPS Main Window

Spacecraft time and UTC are defaulted to the current system time until they are set by the operator. To override these times, click on *Set Time*... under Control, as shown in Figure 3-2.

If a scenario is being executed, the filename is displayed; otherwise, the scenario field is blank and the Scenario File label is dimmed.

The MPS Simulation Mode is set by selecting one of the options below "Simulation Mode". In the EDOS simulation mode, the MPS communicates with the EOC in EDOS formats using network protocols. In the spacecraft simulation mode, the MPS interfaces with EDOS as if through the Space Network (SN) or contingency Ground Network (GN). The selected simulation mode determines many of the options available on other windows and default settings; options not available for a particular mode are desensitized. For playback data transmission in the EDOS simulation mode, the data is Rate Buffered by default unless the operator unsets one or both channels in the submenu *Create RB files* > under Control.

The source of telemetry data is set by selecting one of the options below "Data Source". Selecting **HK File** causes a file selection box to appear. Selecting an HK file from the box and clicking on **OK** in the file selection box causes the filename to be displayed in the field to the right of "HK File". The HK file field is dimmed when the data source is PDB. When **HK File** is selected, the MPS will use the identified file as a data source and create up to two streams of telemetry based on file contents. When **PDB** is selected, the MPS will operate in a dynamic mode, creating packets based on AM-1 database information. The **PDB** option allows for parameters to be set using activity log information. Selecting **PDB** sensitizes the **Activity Log** button. Selecting **Activity Log** causes a file selection box to appear. Selecting an activity log file and clicking on **OK** in the file selection box causes the filename to be displayed in the Activity Log file field. This field is dimmed and the **Activity Log** button is desensitized when the data source is HK file. The data source cannot be changed while the telemetry is active.

Selecting an option from the Spacecraft Phase menu determines telemetry combinations as identified in the AM-1 detailed mission requirements (Reference 8).

Below the work area is a scrolled text area. This is where the system log is displayed. The system log includes process status, command directives, and errors. Every entry in the log is timestamped with the time of occurrence. These items are also entered into the system log file on disk. The file is named with the date and time of its creation.

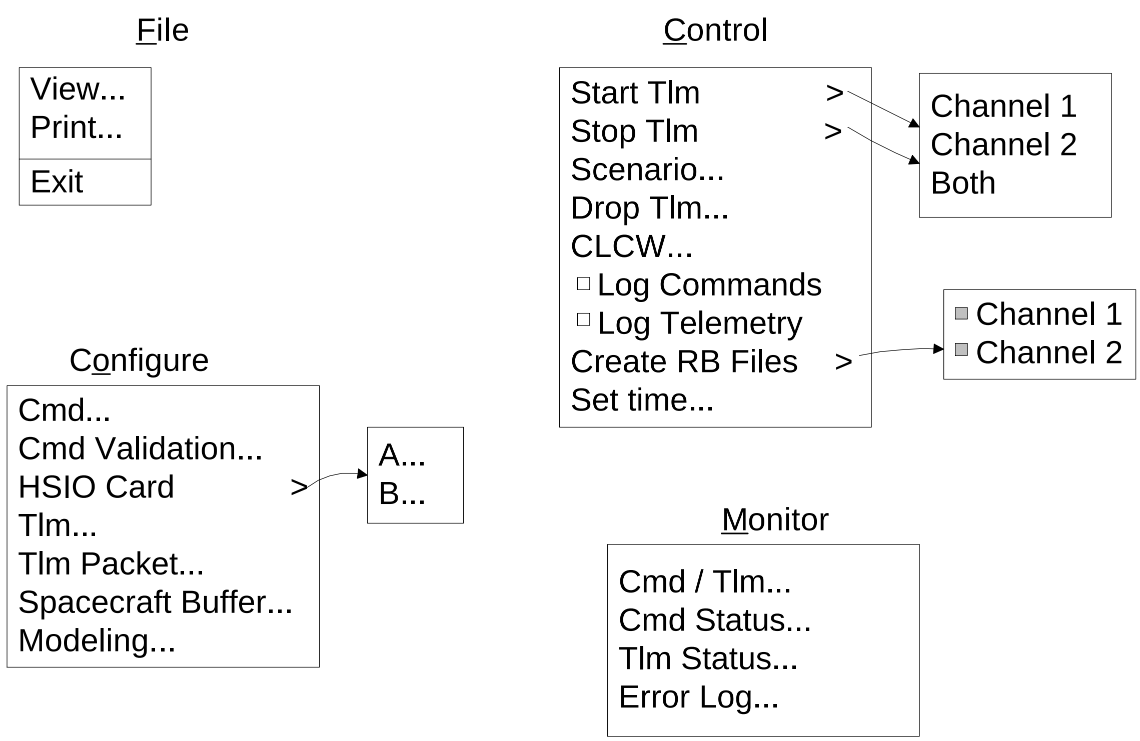

Figure 3-2. Main Window Menu Bar

3.2.2.1.1 File

The File menu provides access to disk files that are associated with the MPS.

View or edit a file by selecting the *View...* option.

Print a file by selecting *Print...*.

3.2.2.1.2 Control

The Control menu provides control of MPS functions.

Telemetry transmission is started by selecting *Channel 1*, *Channel 2*, or *Both* in the submenu of *Start Tlm >*. If telemetry is not active, the *Start Tlm >* is sensitized and *Stop Tlm >* option is desensitized. When telemetry is started in the EDOS simulation mode, the MPS sends a message to the OMD Simulator to start CODA transmission. If the OMD Simulator is not active, a statement to that effect is added to the system log at the bottom of the main window.

Telemetry is stopped by selecting *Channel 1*, *Channel 2*, or *Both* in the submenu of *Stop Tlm >*. If telemetry is active for the respective channel, the *Stop Tlm >* option is sensitized and the *Start Tlm >* option is desensitized. When telemetry is stopped in the EDOS simulation mode, the MPS sends a message to the OMD Simulator to stop CODA transmission. If the OMD Simulator is not active, a statement to that effect is added to the system log at the bottom of the main window.

Selecting *Scenario...* brings up the Scenario dialog box. The box enables the operator to select, start, and stop a scenario.

The *Log Commands* option is a toggle. Setting it brings up a prompt dialog box for the operator to specify a filename. The default filename is timestamp.cmd; timestamp indicates the date and time the file was created. After the filename and directory are established, the file is opened and the MPS begins logging received spacecraft commands. Unsetting the *Log Commands* option causes the MPS to stop logging commands and to close the command log file.

The *Log Telemetry* option is a toggle. Setting it brings up a prompt dialog box for the operator to specify a filename. The default filename is timestamp.tlm; timestamp indicates the date and time the file was created. After the filename and directory are established, the file is opened and the MPS begins logging telemetry to it. Unsetting the *Log Telemetry* option causes the MPS to stop logging telemetry and closes the telemetry log file.

Create RB Files menu options are toggles. Rate buffered files will be generated as a result of EDU transmission when these options are set, which is the default.

The *Set Time...* option causes the Set Time dialog box to appear. This dialog box allows the operator to change the UTC and spacecraft times.

3.2.2.1.3 Configure

Selecting any option under Configure causes the corresponding configuration dialog box to appear. The dialog boxes enable the operator to configure the system for receiving commands, transmitting telemetry, or setting modeling parameters**.**

HSIO > is sensitized (available) only when the MPS is operating in spacecraft simulation mode. The High Speed Input Output (HSIO) card needs to be configured to transmit telemetry and to receive commands in the MPS spacecraft simulation mode.

3.2.2.1.4 Monitor

Selecting any option under Monitor causes the corresponding monitor dialog box to appear.

3.2.2.2 File Menu Options

The following diagrams represent the dialog boxes that appear in response to the various options in the File menu. A generic file selection box, shown in Figure 3-3, is used to select files for the File operations.

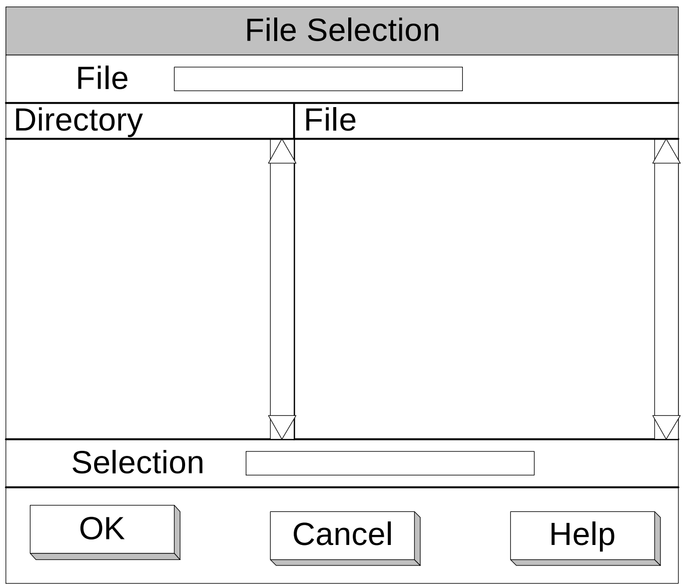

Figure 3-3. Example of File Selection Box

The File filter, e.g. "x*.scn", appears in the field to the right of the File label. The directory being searched is indicated in the Directory section. Selecting a directory in this section changes the directory of the search. Directory access will be limited to prevent unauthorized access. All files matching the file filter in the specified directory are listed in the File section. Selecting a file in this section puts the filename in the Selection field below. Clicking on **OK** causes the selected file to be used for whatever purposes the parent dialog requires. Clicking on **Cancel** closes the File Selection Box without any file being selected.

View...

The generic file selection box shown in Figure 3-3 is used for selecting the file for viewing or editing. Selecting a file and clicking on **OK** in the file selection box brings up a terminal emulator window which automatically executes the preferred editor on the selected file. The preferred editor is indicated via the UNIX environment variable EDITOR. If EDITOR is not specified, vi is used. If the file is not write protected, it can be modified. The window is closed automatically when the operator exits the editor.

Print...

The generic file selection box shown in Figure 3-3 used for selecting the files for printing. Selecting one or more files and clicking on **OK** in the file selection executes a UNIX system command to print the selected file to the default printer. The files are printed without any further operator intervention.

Exit

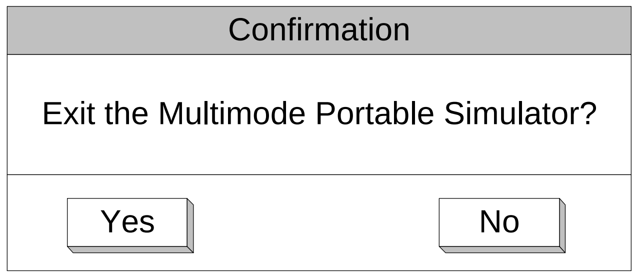

Figure 3-4.   Exit Confirmation Dialog Box

Clicking on **Yes** on the dialog box shown in Figure 3-4 continues shutting down the MPS system. Clicking on **No** returns control to the MPS’s main Window. If telemetry transmission or command reception is in progress, a question box with the message “Telemetry transmission is active. Continue shutting down MPS?”; or “Commands are being received. Continue shutting down MPS?”. Clicking on **Yes** continues shutting down the MPS. Clicking on **No** returns control to the MPS’s main window.

3.2.2.3 Control Menu Options

The following diagrams represent the dialog boxes that appear in response to the various options in the Control menu shown in Figure 3-5.

Scenario...

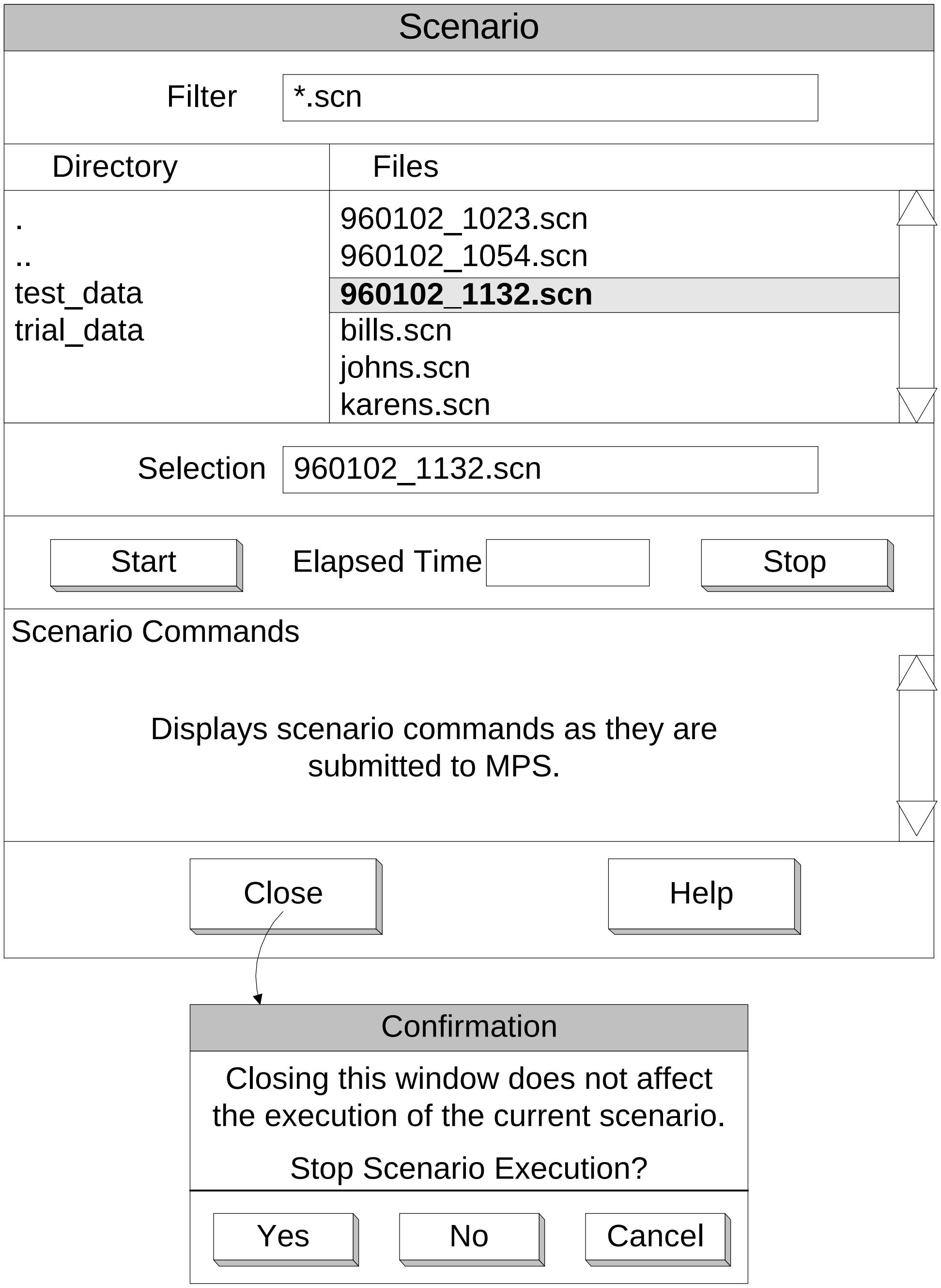

Figure 3-5. Scenario Dialog Box

The Scenario dialog box shown in Figure 3-5 allows the user to initiate and monitor the progress of a time ordered sequence of preset simulator commands. These simulator commands are read from a scenario file. The scenario file is edited via a text editor such as vi and contains telemetry configuration directives and the relative times they are to be executed.

The scenario file is selected in the normal way for file selection boxes. Selecting a file and clicking on the File Selection Box's **OK** button enables the **Start** button.

The directives in the selected file are executed by clicking on **Start**. Execution is stopped by clicking on **Stop** or when the scripted scenario steps are exhausted. The time elapsed since the scenario started is displayed immediately after execution is started. Each item in the file is displayed as it is submitted to the MPS in the scrolled panel at the bottom.

To avoid executing more than one scenario at a time, the **Start** button is made unavailable (dimmed) when a scenario is started. The **Stop** button and the elapsed time display are dimmed until a scenario is started

Click on **Close** to close the window. **Close** does not affect scenario execution; therefore, if a scenario is being executed, a confirmation dialog box is displayed asking if the operator wants to stop execution of the scenario. Clicking on the confirmation's **Yes** button stops the scenario execution and closes both the confirmation dialog box and the Scenario dialog box. Clicking on the confirmation's **No** button closes both dialog boxes; the MPS continues executing the scenario. If a scenario is being executed when the window is closed, the Scenario dialog box must be displayed again to stop scenario execution if it is not allowed to complete in its own time. Clicking on the **Cancel** button returns control to the Scenario dialog box without closing it.

Drop Tlm...

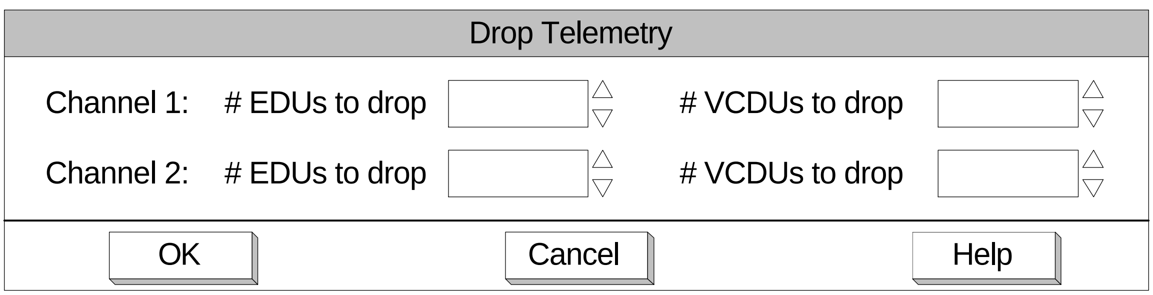

Figure 3-6. Drop Telemetry Dialog Box

The Drop Telemetry dialog box shown in Figure 3-6 allows the user to suppress the transmission of telemetry on one or both channels. The "# EDUs to drop" label and field is sensitized only when the MPS is operating in the EDOS simulation mode. The "# VCDUs to drop" label and field is sensitized only when the MPS is operating in the spacecraft simulation mode.

The number of EDUs or VCDUs to drop for both channels can be edited directly or by using the spinners to increase or decrease the numbers by a given increment. The increment depends on the range of drops possible.

Clicking on **OK** configures the specified drop numbers and closes the dialog box. Clicking on **Cancel** closes the dialog box; no changes are saved.

CLCW...

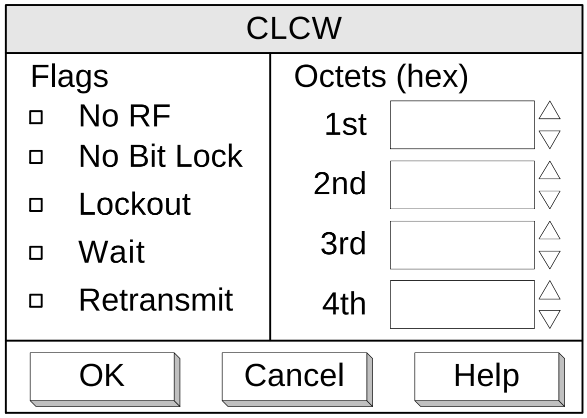

Figure 3-7. CLCW Configuration Dialog Box

The CLCW Configuration dialog box shown in Figure 3-7 allows the operator to override the current value of the CLCW by toggling bit values (set equals 1) or by entering a hexadecimal value for one or more of the four CLCW octets. The correspondence between the toggled bit variables and CLCW field names is found in section 5-5 of the EDOS EGS Interface Control Document (Reference 10).

Clicking on **OK** configures the CLCW flags and octets as indicated. Clicking on **Cancel** cancels configuration modifications.

Set Time...

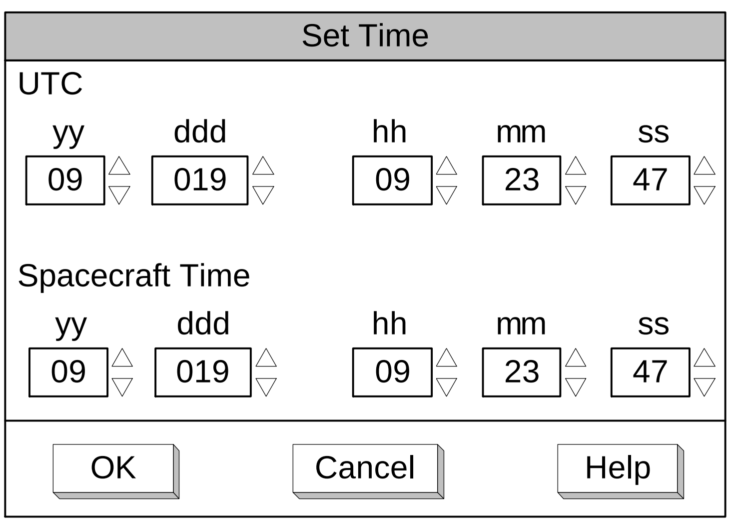

Figure 3-8. Set Time Dialog Box

The Set Time dialog box shown in Figure 3-8 allows the user to set both spacecraft time and UTC.

UTC is used to timestamp blocked transmission of CADUs. UTC can be read from the timecard or from the Set Times dialog box. Spacecraft time is used to timetag telemetry packets. The UTC and spacecraft time are initialized by default to the system time. Use the spinners (up and down arrows) to change a field or click on a field itself to edit it manually.

Clicking on **OK** sets the time. Clicking on **Cancel** closes the dialog box without changing either time.

3.2.2.4 Configure

The following diagrams represent the dialog boxes that appear in response to the various options in the Configure menu.

Command...

There are two different dialog boxes for configuring command, one for each simulation mode. The corresponding Command Configuration dialog box appears for the simulation mode that MPS is operating in.

The Command Configuration dialog box for the spacecraft simulation mode, Figure 3-9, provides the settings for the selected HSIO receiving the commands and the command header values to be verified, if command header verification is turned on.

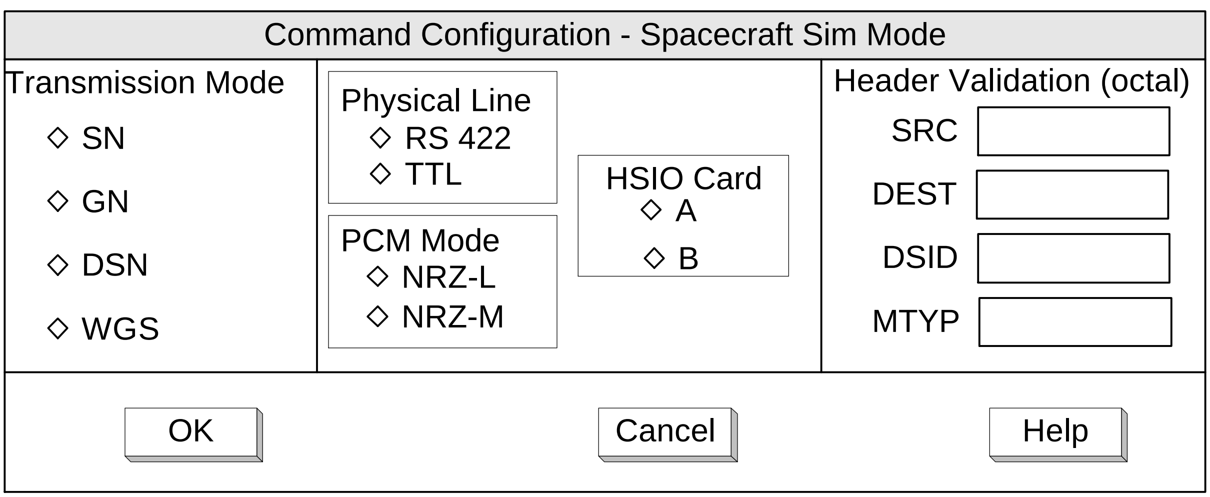

Figure 3-9. Command Configuration - Spacecraft Sim Mode Dialog Box

Both Space Network (SN), or TDRSS, and Ground Network (GN) formats are simulated in the spacecraft simulation mode. In the SN transmission mode, which is the nominal communications support for EOS AM-1, data are received and transmitted as a serial bitstream of clock and data. In the GN transmission mode, which includes contingency support provided by GN, Deep Space Network (DSN) and Wallops Ground System (WGS) ground stations, command and telemetry data are encapsulated in Nascom 4800-bit blocks. EBnet provides the blocking and deblocking functions for EDOS.

The Command Configuration Spacecraft Sim Mode dialog box shown in Figure 3-9 allows the operator to configure the selected HSIO for an identified communication mode. Selecting any of the GN block modes (i.e., GN, DSN, or WGS) will result in the sensitizing of the header validation panel. An octal value is required for header validation items which are edited directly. The header validation panel is dimmed when the transmission mode is SN because there are no headers associated with the serial data stream.

Setting the transmission mode in this dialog box sets it also for the Telemetry Configuration dialog box. Setting the transmission mode in the Telemetry Configuration dialog box sets it here as well.

Clicking on **OK** closes the dialog box and causes the configuration to be set immediately. Clicking on **Cancel** closes the dialog box; any changes to the command configuration are lost.

The Command Configuration dialog box for EDOS simulation mode (Figure 3-10) allows the user to enter expected values for fields in the EDOS ground message header attached to the EOC command data blocks, including source, destination, spacecraft ID, message type, and expected sequence number. The entered values are compared against the respective fields in incoming command data blocks. If the actual value does not match the expected value, an error message is written to the system log file, but the command is still processed.

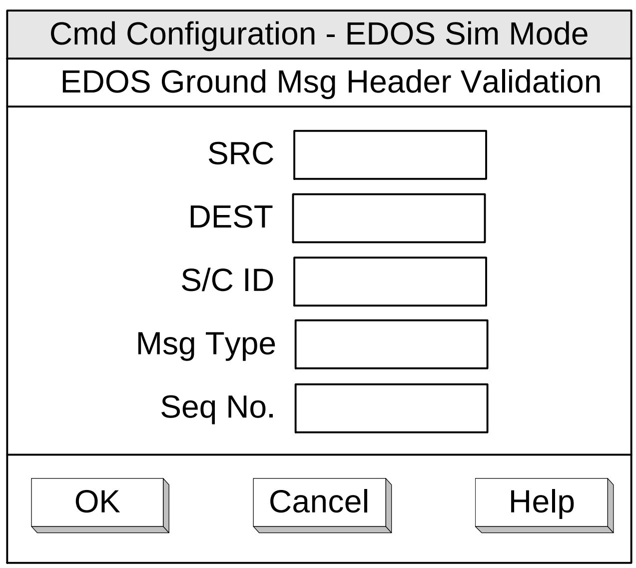

Figure 3-10.   Command Configuration - EDOS Sim Mode Dialog Box

Clicking on **OK** closes the dialog box and causes the configuration to be set immediately. Clicking on **Cancel** closes the dialog box; any changes to the command configuration are lost.

CMD Validation...

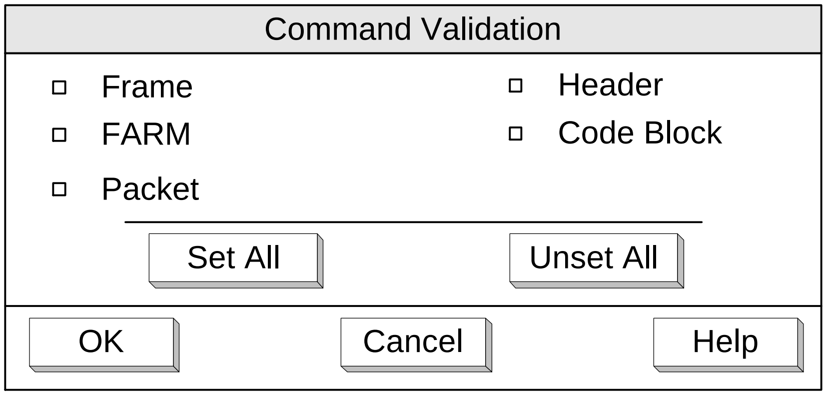

Figure 3-11. Command Validation Dialog Box

The Command Validation dialog box shown in Figure 3-11 allows the operator to turn on or off selected validation options. The Header and Code Block options are sensitized only when the Simulation mode is Spacecraft and the Transmission mode is one of the blocked modes (GN, DSN, or WGS).

Clicking on **Set All** sets all validation options relevant to the current simulation and transmission modes. Clicking on **Unset All** unsets all validation relevant to the current simulation and transmission modes.

Clicking on **OK** configures the command validation options. Clicking on **Cancel** cancels configuration changes.

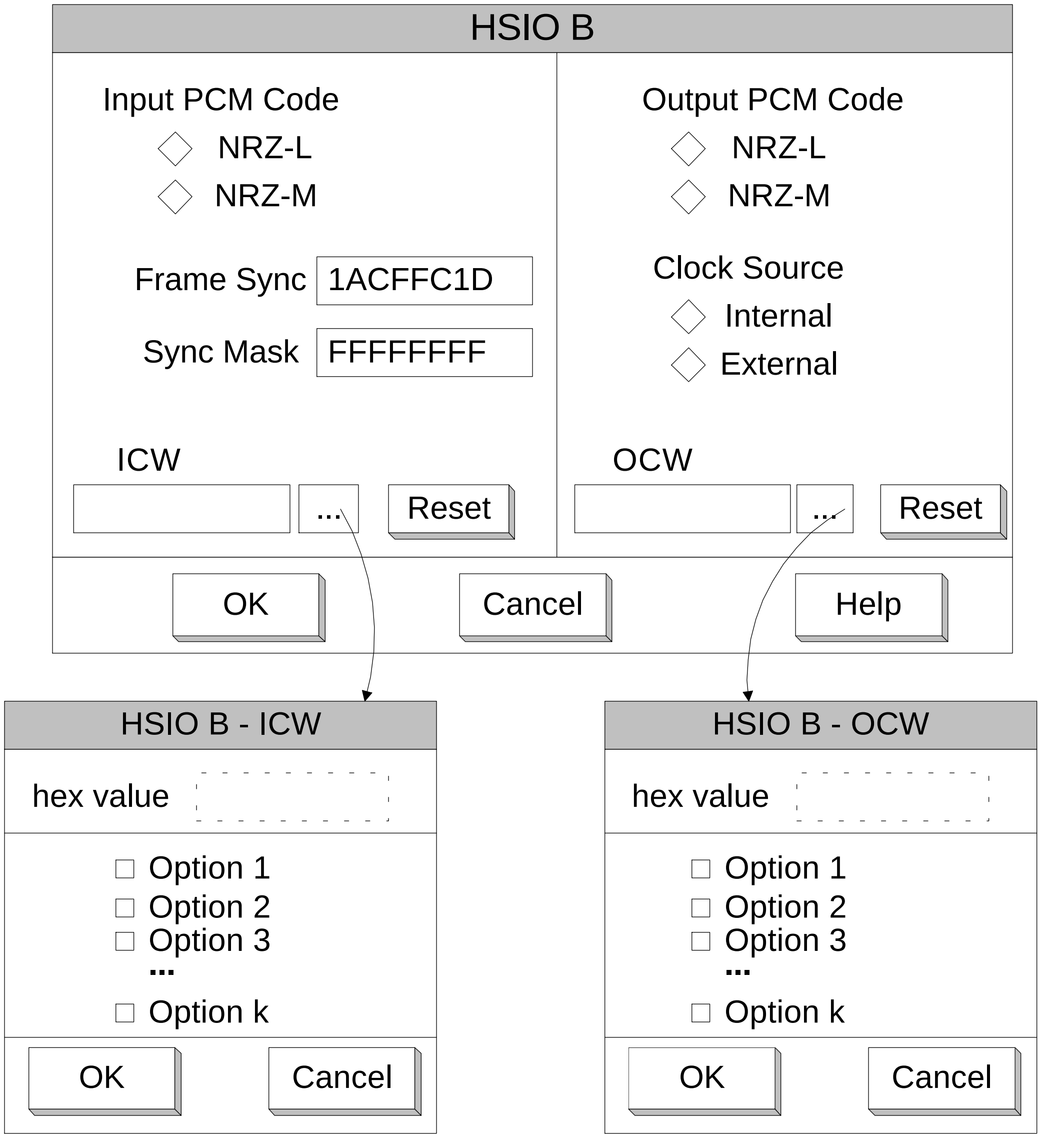

Figure 3-12. HSIO Configuration Dialog Box

The HSIO card being configured is identified in the title of the dialog box shown in Figure 3-12. The input and output PCM codes are set by selecting a code. Clock Source is also a radio box. The Frame Sync and Sync Mask are set by editing directly.

The input control word (ICW) and output control word (OCW) are set in the same way: Their hexadecimal values are editable directly but also can be changed by selecting individual options. Clicking on **"..."** brings up a secondary dialog box that gives a list of ICW/OCW options. The options are set according to the initial value or what the operator has already typed in the ICW or OCW fields directly. The options will be specified at a later date. The hexadecimal value resulting from the options selected is displayed in the Hex Value box. This allows the operator to become familiar with the hexadecimal values required for common option settings. Clicking on

OK in the secondary box sets the ICW or OCW. Clicking on **Cancel** in the secondary box closes the secondary box; the ICW or OCW remains as it was before clicking on **"..."**. The **Reset** button to the right of **"..."** resets the icw or ocw to the value it had when the dialog was initiated.

Clicking on **OK** in the HSIO Configuration dialog box shown in Figure 3-12 configures the HSIO card. Clicking on **Cancel** cancels the configuration changes.

TLM...

There are two dialog boxes for the Telemetry Configuration; one for spacecraft and one for EDOS simulation mode. The appropriate Telemetry Configuration dialog box appears depending on the simulation mode selected for the MPS.

The Telemetry Configuration dialog box for the spacecraft simulation mode, shown in Figure 3-13, allows the operator to identify the transmission mode, the transmission rate for each channel, and NASCOM block header values for GN transmission formats.

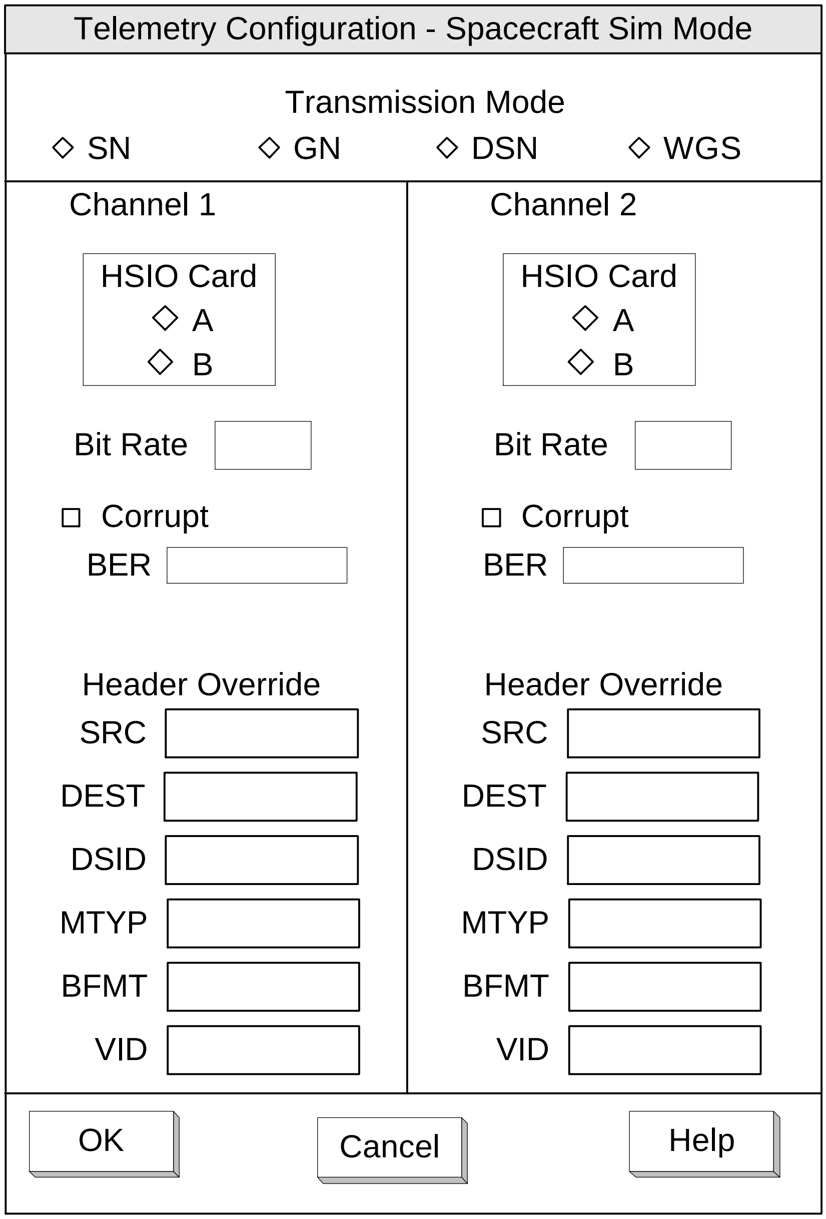

Figure 3-13. Telemetry Configuration - Spacecraft Sim Mode Dialog Box

For spacecraft simulation mode, selecting the SN transmission mode causes the "Header" entries to become desensitized because SN is a serial mode and the header entries are for blocked transmission mode only. The other transmission modes are blocked. When a blocked (i.e., GN, DSN, or WGS) transmission mode is selected, an octal override may be entered for each header field. In either SN or blocked simulation mode, selecting **Corrupt** sensitizes the Bit Error Rate (BER) field and causes a bit to be corrupted (e.g., inverted) at the user-specified BER (i.e., the BER specifies an interval of X bits such that every Xth bit is inverted for the duration of the transmission.). Change the BER by editing the field.

Setting the transmission mode in this dialog box also sets it for the Command Configuration dialog box. Setting the transmission mode in the Command Configuration dialog box sets it here as well.

Clicking on **OK** causes the telemetry transmission to be configured as specified in the dialog box. Clicking on **Cancel** closes the dialog box without affecting the telemetry configuration.

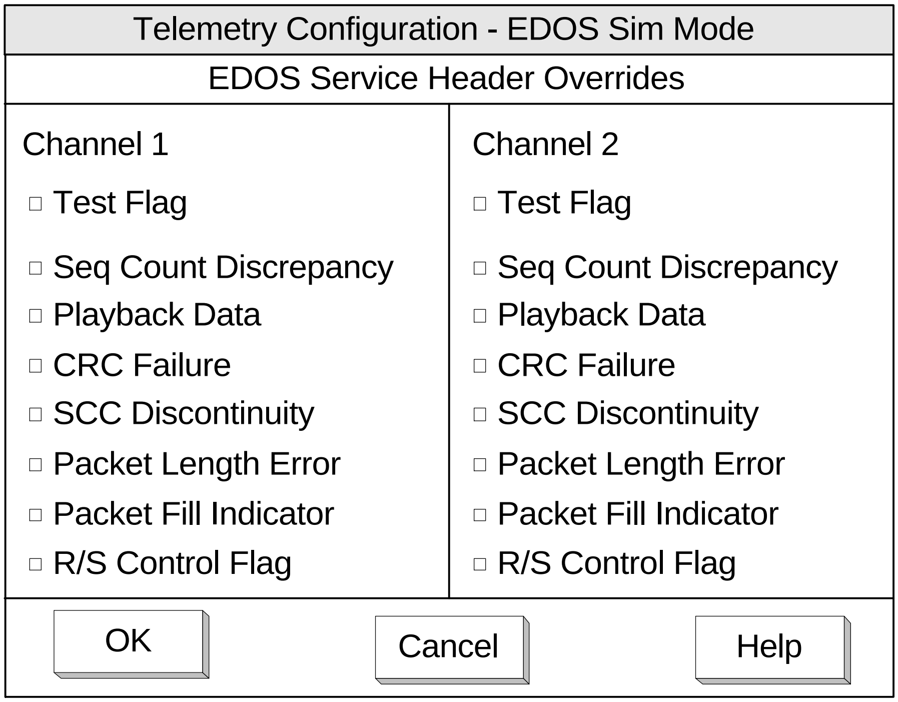

Figure 3-14.   Telemetry Configuration - EDOS Sim Mode Dialog Box

The Telemetry Configuration dialog box for EDOS Simulation mode, shown in Figure 3-14, allows the operator to set header bit fields within the EDOS Service Header for a selected EDU, including test/operations data flag, source sequence count (SSC) discrepancy, playback data flag, CRC failure indicator, packet length error, packet fill indicator, and Reed-Solomon (R/S) control flag.

Clicking on **OK** causes the telemetry transmission to be configured as specified in the dialog box. Clicking on **Cancel** closes the dialog box without affecting the telemetry configuration.

TLM Packet...

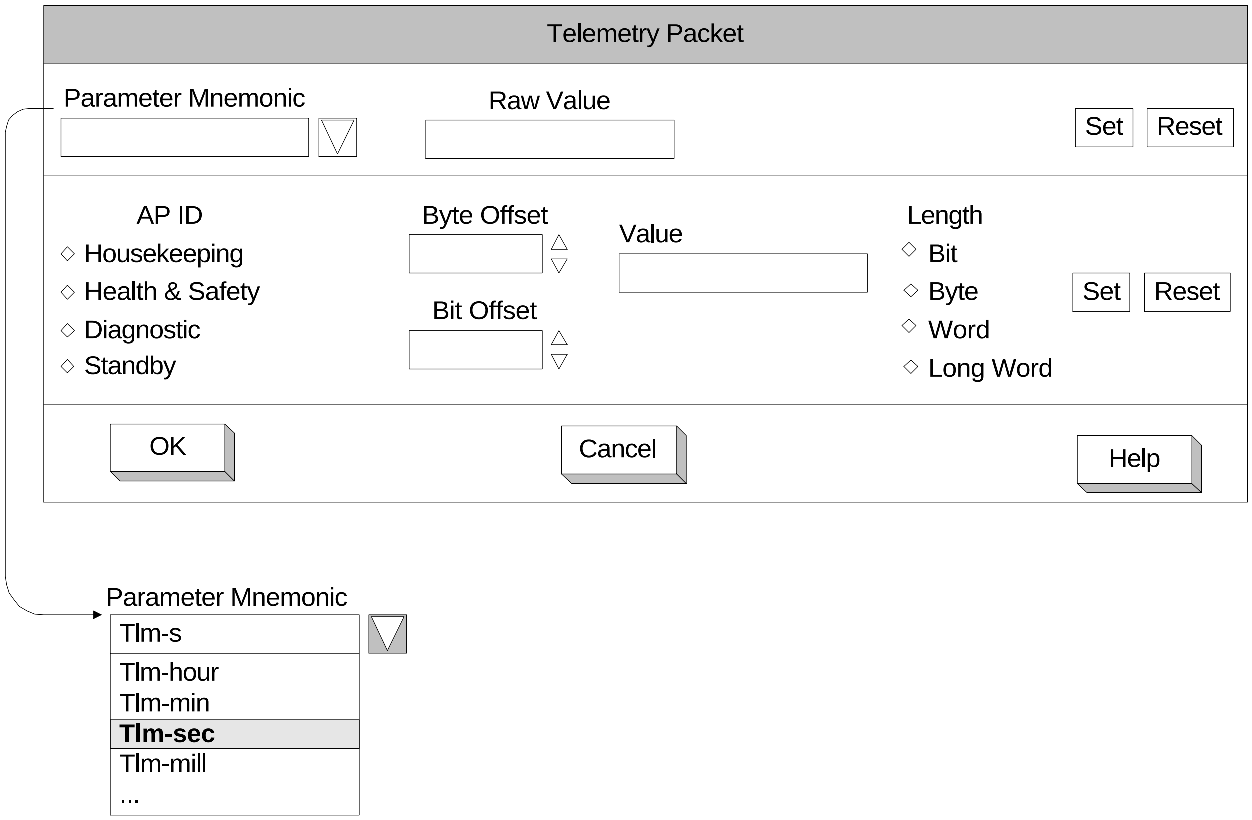

Figure 3-15. Telemetry Packet Dialog Box

If an initialization table is not provided using the modeling file option, then all telemetry discrete parameters are initialized to their off state and all analog parameters to their minimum values as given in the PDB. The Telemetry Packet dialog box shown in Figure 3-15 allows the operator to set the value of a selected parameter, or to enter a value into a selected location within an identified packet.

The Parameter Mnemonic field is editable. Clicking on the down arrow brings up a list of mnemonics which are selectable. The Parameter value is assumed to be raw; it can only be set by editing. Click on the **Set** button to set the selected parameter's value. Click on **Reset** to restore the parameter to its previously set value.

The telemetry value can also be set by specifying an APID and a byte and bit offset. Again, the value is assumed to be raw. The length of the value if other than the default must be specified. Clicking on the **Set** button sets the specified position or parameter's value. Clicking on **Reset** restores the position or parameter to its previously set value.

Any number of telemetry parameters and/or positions can be modified without closing the dialog box. Each parameter and position are set via the **Set** buttons. If the operator sets a parameter and a position that coincides with that parameter, the parameter value is taken and then modified by the position value.

Clicking on **OK** configures the telemetry. Clicking on **Cancel** cancels all changes made to all parameters and positions during the current dialog session.

Spacecraft Buffer...

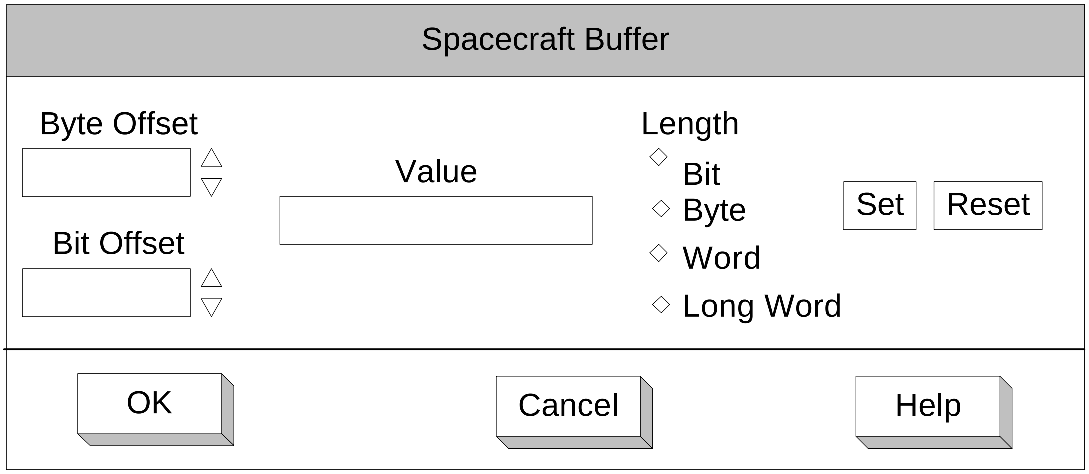

Figure 3-16. Spacecraft Buffer Dialog Box

The MPS Spacecraft Buffer is an area of simulated spacecraft memory, which is used for either table or memory loads. The Spacecraft Buffer dialog box shown in Figure 3-16 allows the operator to set a value within the table or memory load, depending on which is being accessed in a given simulation.

The Byte and Bit Offsets are editable directly or can be set by spinning the values up or down by a TBS increment. The Value is editable and is assumed to be raw. The length must be specified. Clicking on **Set** sets the value for the specified byte and bit offset. Clicking on **Reset** restores the value to its previously set value.

Any number of positions can be modified without closing the dialog box. Each position is set via the **Set** buttons. Clicking on **OK** configures the spacecraft buffer. Clicking on **Cancel** cancels all changes made to all positions during the current dialog session.

Modeling...

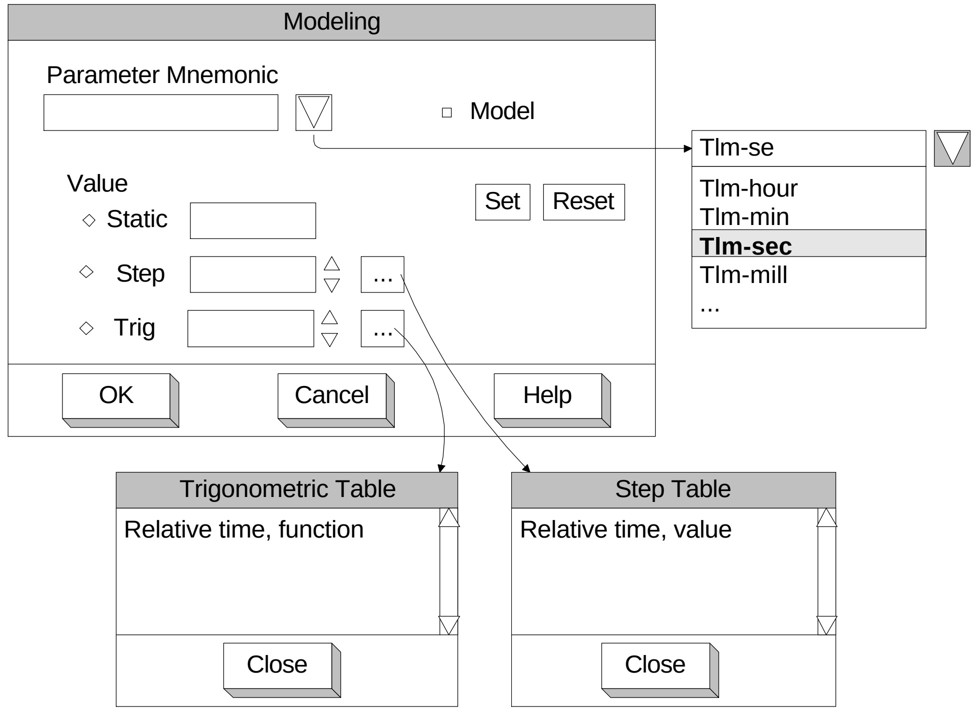

Figure 3-17. Modeling Dialog Box

The Modeling dialog box shown in Figure 3-17 allows the operator to start and stop selected modeling functions for an identified parameter. The modeling files which support this option will reside on the VME, and are provided by the flight operations team. The files are not modifiable, however, the association between the modeling algorithms and telemetry parameter can be changed.

The Parameter Mnemonic field is editable. Clicking on the down arrow brings up a list of mnemonics which are selectable.

Selecting **Model** turns the feature on or off. Selecting **Static**, **Step** or **Trig** causes the parameter value to be associated with the type and makes the field sensitized. The static value is assumed to be raw; it can only be set by editing and the parameter will hold that value until it is reset. The Step and Trig fields are editable. Clicking on the Step and Trig spinners causes their values to be increased or decreased by TBS increment. Clicking on **"..."** causes more information to be displayed about the functions specified. Clicking on **Close** causes the secondary dialog box to be closed, no other action is taken.

Clicking on **Set** sets the selected parameter. Clicking on **Reset** restores the parameter to its previously set value.

Any number of modeling parameters can be modified without closing the dialog box. Each parameter value is saved via the **Set** button. Clicking on **OK** configures the modeling and closes the dialog box. Clicking on **Cancel** cancels all changes made to all parameters during the current dialog session.

3.2.2.5 Monitor

The following diagrams represent the dialog boxes that appear in response to the various options in the Monitor menu.

CMD/TLM...

The operator can request current information about data reception and transmission through the Command/Telemetry Monitor dialog box shown in Figure 3-18.

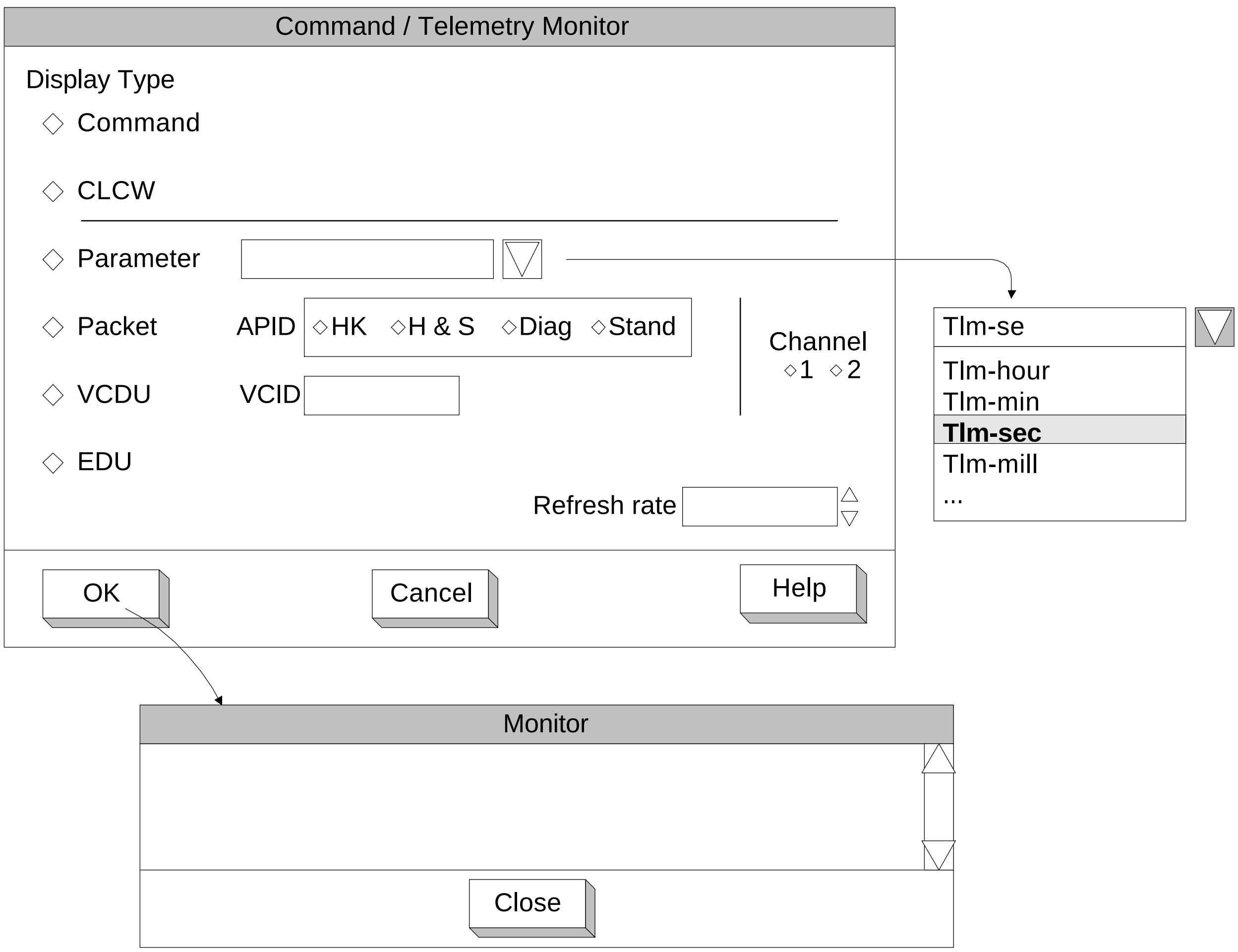

Figure 3-18. Command/Telemetry Monitor Dialog Box

The Command/Telemetry Monitor dialog box allows the operator to view, in hex, a display of the last received command, transmitted telemetry unit, or current CLCW. From this dialog box the operator can also monitor changes in the value of a selected telemetry parameter, or a given packet, VCDU, or EDU.

Selecting Parameter sensitizes the parameter mnemonic and display units. Selecting Packet sensitizes the APID options and channel. Selecting VCDU sensitizes the VCID field and channel options. Refresh Rate applies to all data units that can be monitored. A very large value can be entered to freeze the display.

Clicking on **OK** brings up a dialog box to monitor the selected command or telemetry unit. Clicking on **Cancel** cancels the command and telemetry monitor.

The monitor dialog box displays the type in its title. The units for the parameters and channel number are also given in the title. Clicking on **Close** in the monitor dialog box stops the monitor task and closes the dialog box.

CMD Status...

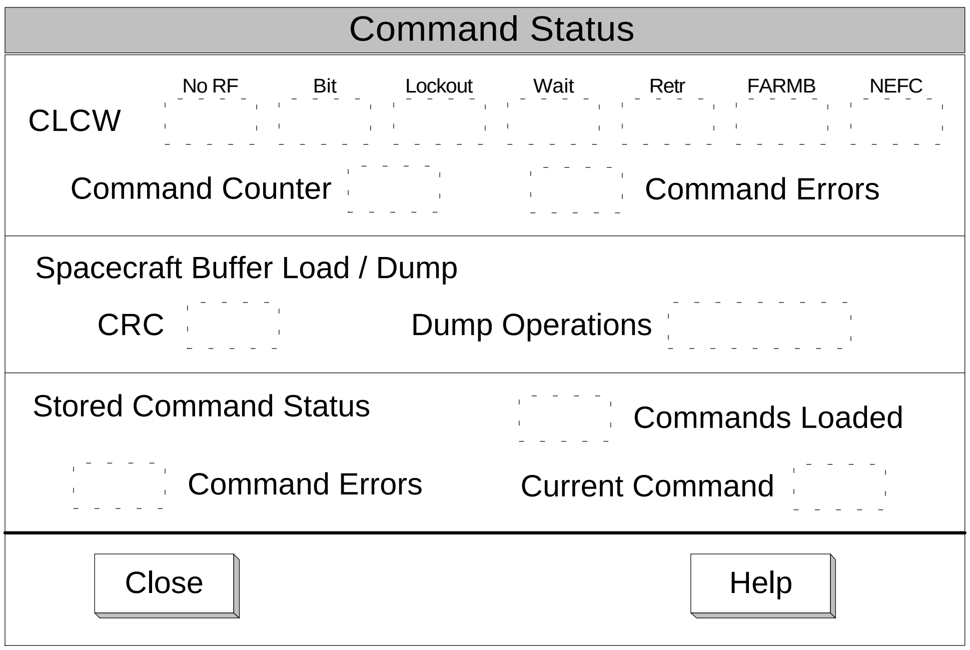

Figure 3-19. Command Status Monitor Dialog Box

The Command Status dialog box shown in Figure 3-19 allows the operator to monitor bit fields within the current CLCW, the number of commands received, and the current count of command errors. Table and Memory load and dump information is also provided.

The CRC status of a load can be Pass or Fail. Dump Operations can be Enabled, Disabled, In Progress, or Completed. The number of commands loaded is the maximum value for Current Command. The number of Command Errors given is the number of errors within the commands loaded.

Clicking on **Close** closes the dialog box.

Tlm Status...

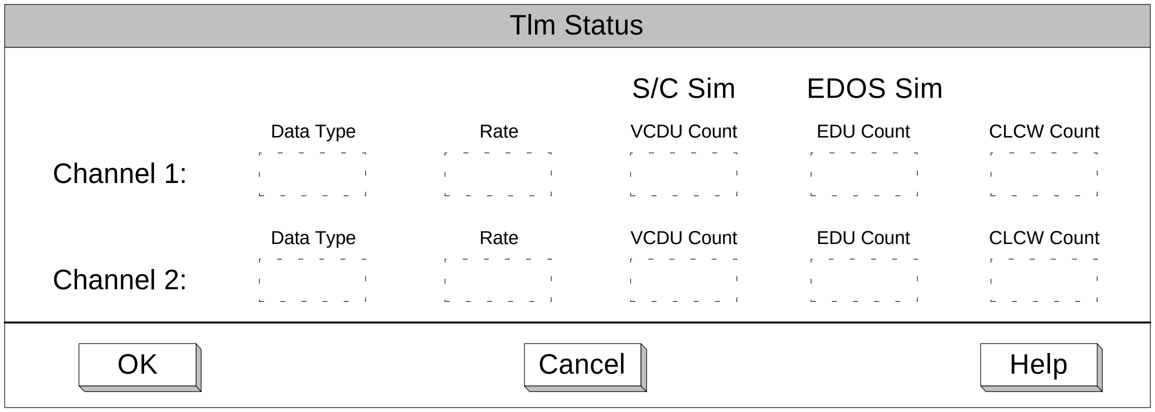

Figure 3-20. Telemetry Status Monitor Dialog Box

The Telemetry Status dialog box shown in Figure 3-20 allows the operator to monitor channel transmission statistics.

If no telemetry is being transmitted on a channel, that channel is dimmed and its values are blank. The VCDU Count is sensitized only when MPS is operating in Spacecraft simulation mode. The EDU Count and CLCW Count are sensitized only when MPS is operating in EDOS simulation mode.

Clicking on **Close** stops the monitor task and closes the dialog box.

Error Log...

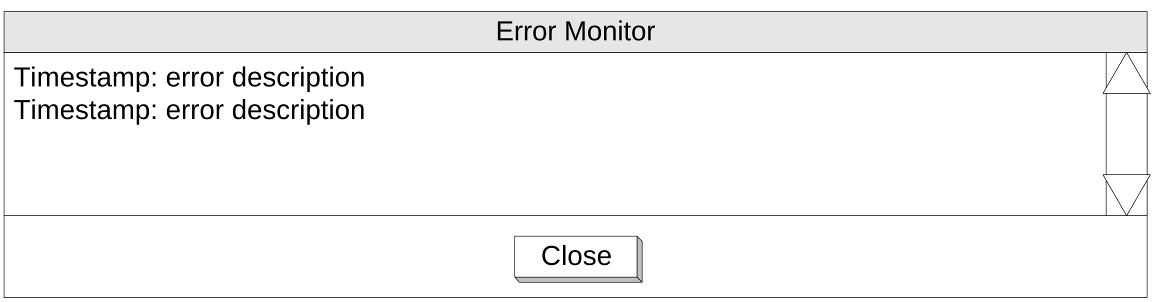

Figure 3-21. Error Monitor Dialog Box

The Error Monitor dialog box shown in Figure 3-21 allows the operator and developer to identify any real-time software anomalies.

Errors are displayed here as they appear in the System Log panel on the main window. Only command and telemetry errors appear in the Error Log.

Clicking on **Close** closes the dialog box.

3.3 Spacecraft Simulation SWCI Design

The real-time system on the VME runs under the PDOS operating system. PDOS is a multi-tasking, real-time operating system that facilitates the breakdown of a system into separate tasks that communicate with one another using services provided by the operating system. Typically, a task will wait on one or a small number of events. When one of these events occurs, the task will be awakened by the operating system. It will determine what sort of event has taken place and respond accordingly. An event may be a time-out, a message received, or a flag set or cleared.

The approach taken for this design was an object- or data-oriented one at the task level. Each task is considered to consist, not of the functions that it calls, but of the functions that control and manipulate the data belonging to the task. The functions and the data each may be divided into two categories, private and public. Private data is data that other tasks may access only on a very controlled basis. The task's private, or internal, functions manipulate this data when the necessary manipulations are not driven by another task. The task's public functions provide controlled access to the private data to other tasks. Many of these public functions correspond to commands or operator directives. This approach allows for more controlled access to data and should enhance reusability.

For each task, there is a brief description of that task's purpose, a list and description of the task's public functions and public data, and a structure diagram of the task's internal functions. Public functions will also appear in a task's structure diagram when the public function provides a significant service to that task's internal functionality. However, most of the public functions for a task are either operator directive functions or command functions and will therefore not appear in any structure diagram.

As described in section 2 (Figures 2-2 through 2-4), the MPS Spacecraft Simulation SWCI consists of the following five subsystems:

- Initialization
- Configuration Support
- Telecommand
- Telemetry
- Modeling

Each of the above subsystems are comprised of a group of related tasks. The following two character abbreviations are used for those tasks. The abbreviations are referenced in the structure charts and supporting design material for each of the five subsystems described in Section 3.3.

AD	System Administrator
CI	Command Ingest
GT	Generate Telemetry
MC	Manage Commands

MO	Modeling
MT	Manage Telemetry
RD	Receive Directives
SC	Stored Commands
SD	Send Data
SM	Software Manager
TI	Keep Time
TO	Telemetry Output

3.3.1 Initialization Subsystem

The initialization subsystem consists of a single task, the System Administrator (AD) which supports the MPS VME task initialization.

3.3.1.1 AD - System Administrator

The AD task is responsible for overseeing the initialization process, seeing that each task gets access to the public functions that it requires.

Public data:

> Global Function Table - A table of tables, where each unique table consists of functions made available by a corresponding VME task. This technique allows for better data management, via indirect access to global data, which eliminates the potential for data corruption by multiple simultaneous access to a unique element.

Public functions:

> ut_getgl: Accepts public functions from all tasks and inserts them into Global Function Table.

Internal functions:

> See Figure 3-22

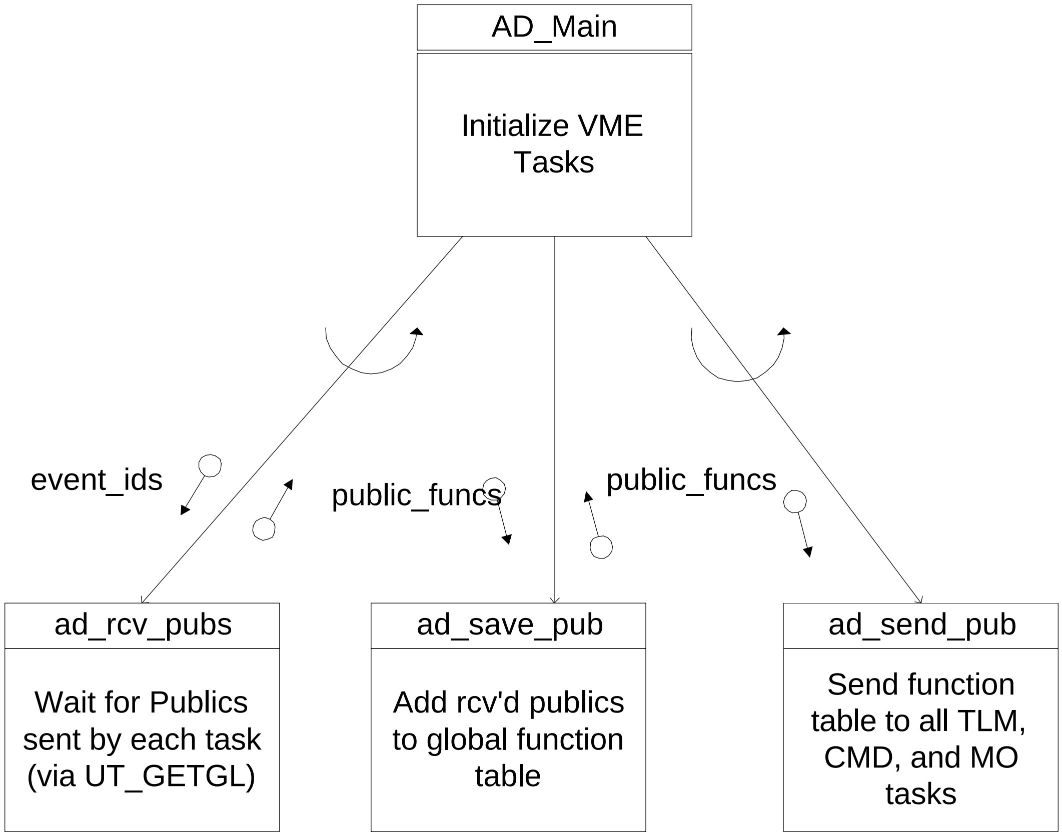

Figure 3-22. System Administrator Task

3.3.2 Configuration Support Subsystem

The configure operator support subsystem consists of two tasks:

RD	Receive Directives	Receive and process user messages
SD	Send Data	Send data to the user interface

These tasks are based on a UNIX/PDOS communication prototype which verified correct data transmission from both a GUI based dialog box to a VME server (RD) and the VME client (SD) to the GUI based data server. These tasks are new to MPS and are not based on reuse from TTS.

3.3.2.1 RD - Receive Directives

The RD task is responsible for interfacing with the front-end in order to receive messages. It uses a TCP/IP socket connection to wait for and read in a message from the user interface, then forwards the message to the appropriate task.

Public data:

None

Public functions:

None

Internal functions:

See Figure 3-23

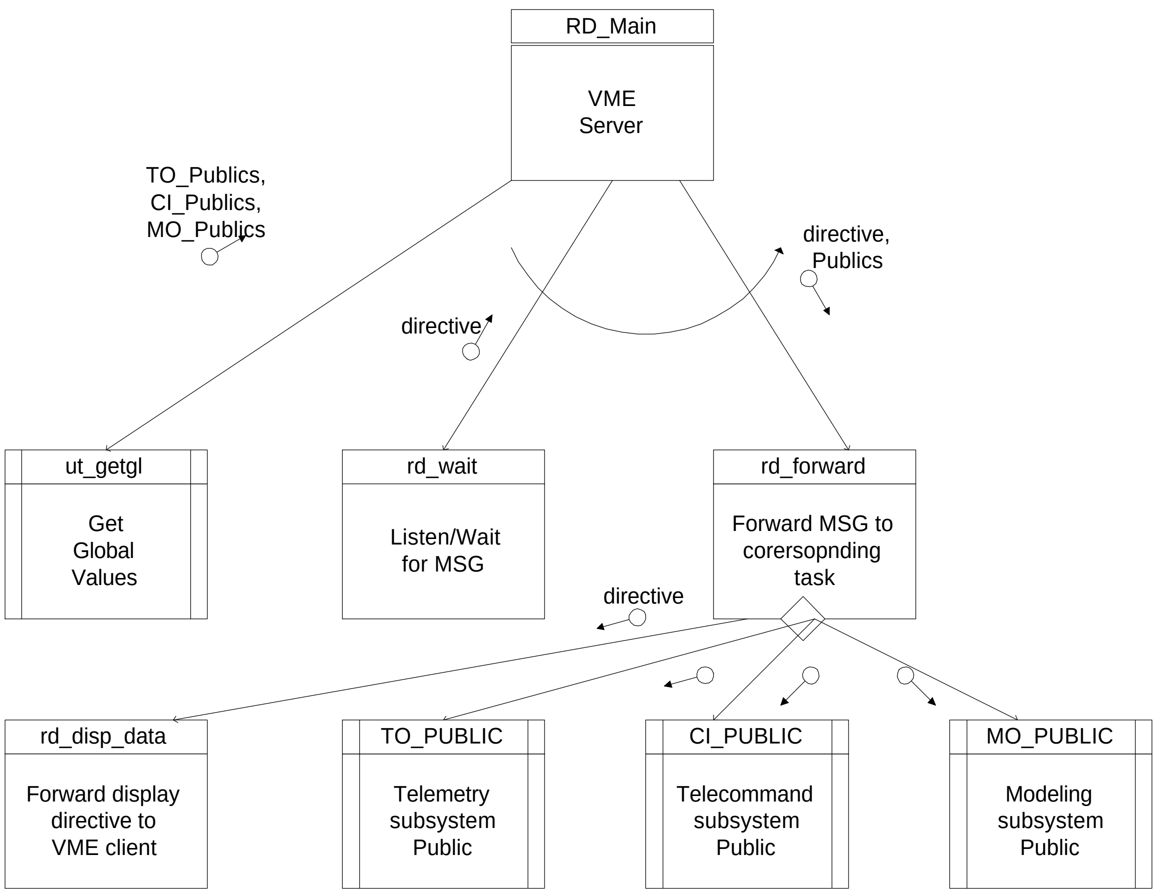

Figure 3-23. Input Messages Task

3.3.2.2 SD - Send Data

The SD task is responsible for gathering all dynamic status display information and sending it to the front-end for display to the operator. It wakes up periodically to transfer data to active displays from a global data area which is written to by the Command, Telemetry, and Modeling tasks.

Public data:

Socket id	Socket used by VME to transfer data to the GUI
Data Area	Partitioned global area which contains telemetry and command status updates, and updated command and telemetry display data. Active data (supporting open windows) will be packaged and transferred from the data area based on the user-specified refresh rate.

Public functions:

None

Internal functions:

See Figure 3-24

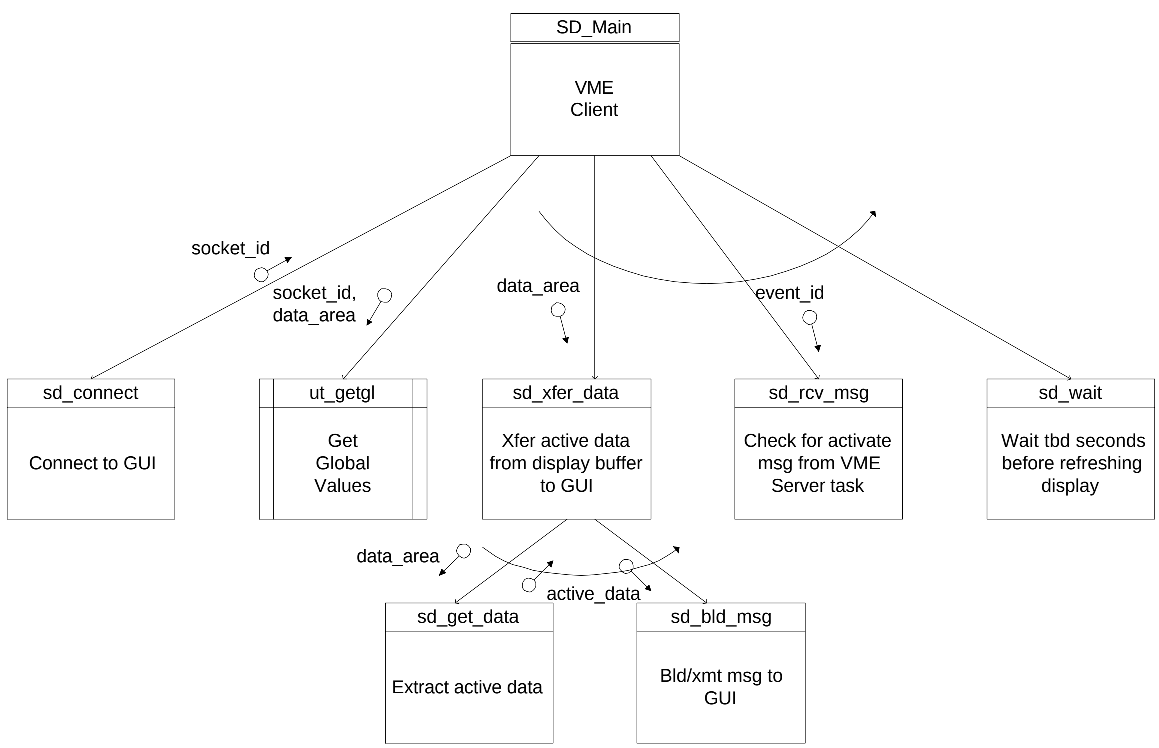

Figure 3-24. Display Status Task

3.3.3 Telecommand Subsystem

The telecommand subsystem handles all commands, whether received from an external source, generated by the operator, or built from stored command data loaded into the simulated memory of the AM-1 on-board computer, called the spacecraft controls computer (SCC). It consists of four tasks:

MC	Manage Commands	Read in PDB telecommand information.
CI	Command Ingest	Receive and validate commands.
SC	Stored Commands	Execute stored commands.
SM	Software Manager	Process loads and dumps.

3.3.3.1 MC - Manage Commands

The MC task is responsible for reading in the telecommand database information and providing access to this information to the other tasks. The telecommand database information is read in at system initialization. Once this is successfully completed, MC becomes an inactive task.

Public data:

Command Table	Hash table of all valid commands and supporting command information read from the AM-1 PDB.

Public functions:

mc_get_cmd_mnem	Given a command mnemonic, returns a pointer to the command record.
mc_get_cmd_bin	Given a command in binary, returns a pointer to the command record.
mc_report_cmd	Given a pointer to the command record, formats and sends to the operator a message indicating that the command has been executed.
mc_xqt_bin	Given a pointer to the command record, sets the end-item verifiers for the command.

Internal functions:

See Figure 3-25

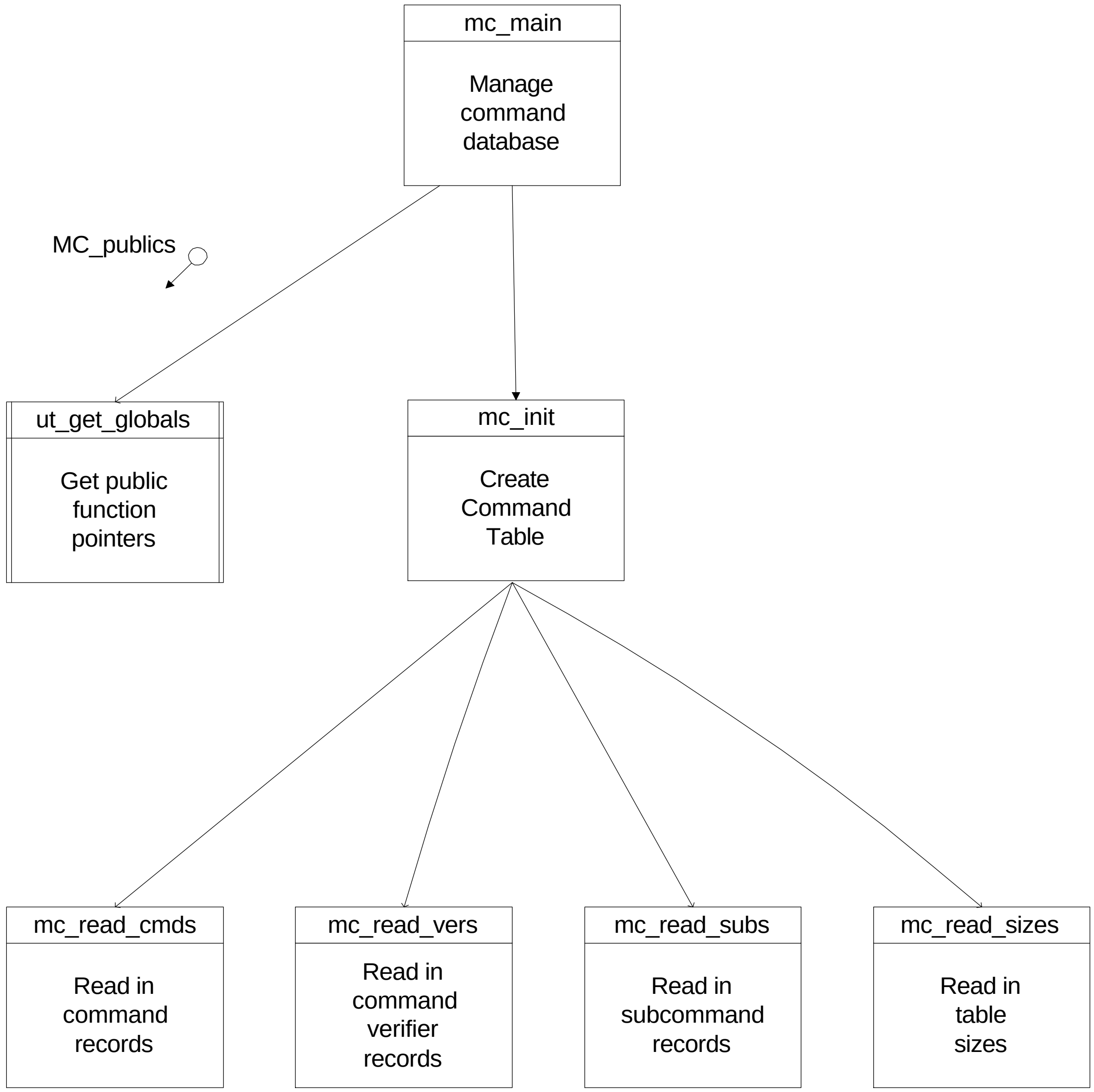

Figure 3-25. Manage Commands Task

3.3.3.2 CI - Command Ingest

The CI task is responsible for receiving commands from an external source, validating them against the PDB, and simulating execution by setting the corresponding end-item verifiers. If the MPS is running in spacecraft simulation mode, the CI will configure the selected HSIO card to receive commands. If running in EDOS simulation mode, the CI public function which supports mode changes will establish the socket connection with the EOC (acting as a server). Following that, the task will be awakened by an interrupt from the HSIO card or upon receipt of message, indicating that a command link transmission unit (CLTU) or UDP/IP command data block has been received. The received command is then validated and parsed using standard CCSDS protocols. The end item verifiers of the resulting commands are then set.

Public data:

CLCW	Command link control word for virtual channel 1.
Block Header Values	Expected values of Nascom and command block header fields.

Public Functions:

ci_clcw	Sets flags in the command link control word.
ci_cmdhdr	Sets values expected in telecommand stream Nascom block headers.
ci_udphdr	Sets values expected in received command block headers.
ci_cmdval	Enables or disables elements of the command validation process, as indicated.
ci_setmd	Sets command simulation mode
ci_logcmd	Enables or disables command logging

Internal functions:

See Figure 3-26

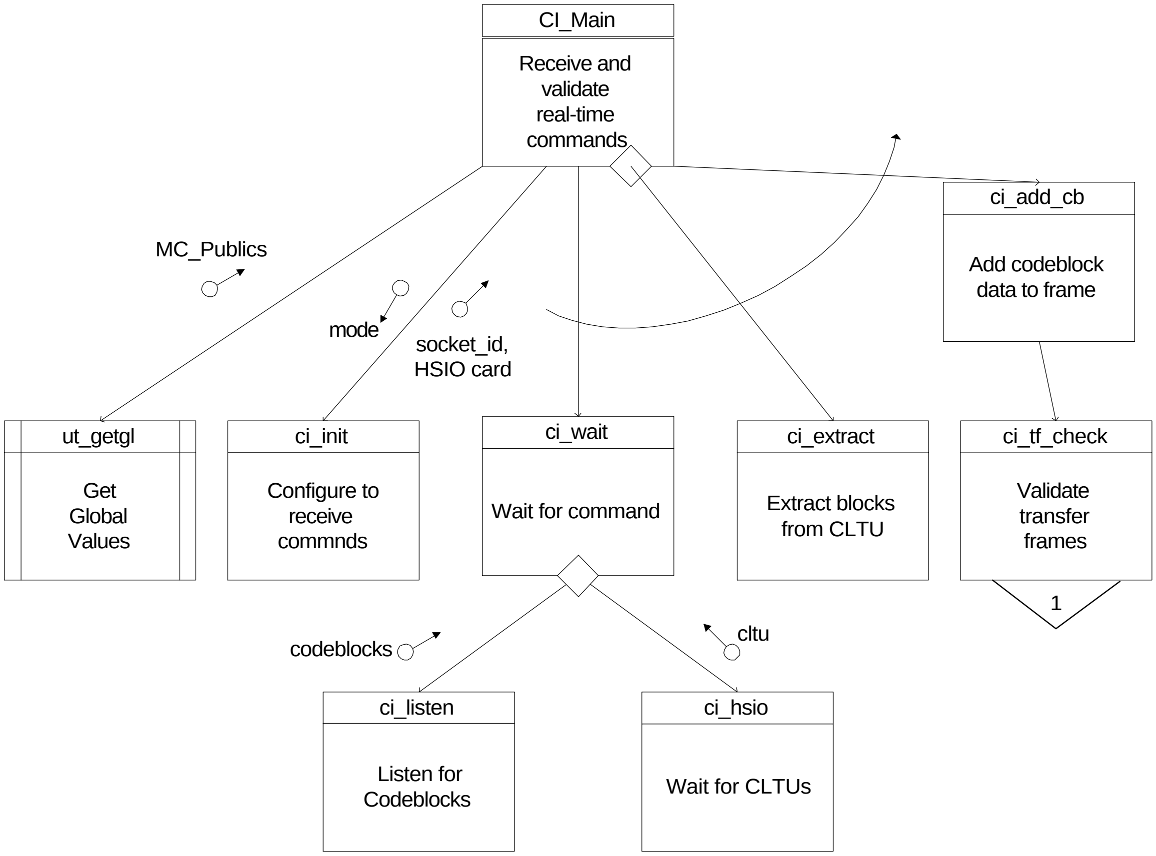

Figure 3-26. Command Ingest Task (1 of 2)

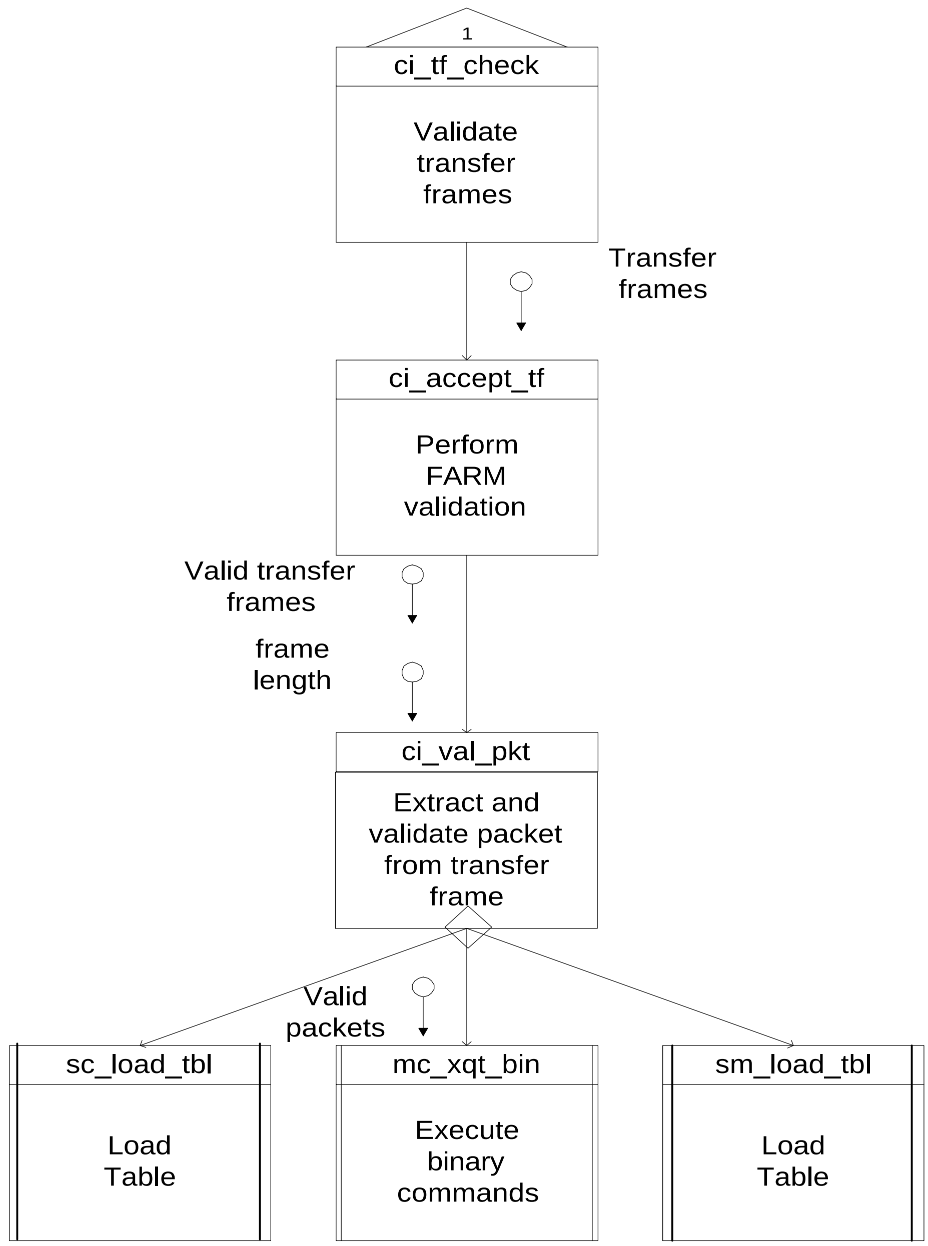

Figure 3-26. Command Ingest Task (2 of 2)

3.3.3.3 SC - Stored Commands

The stored commands task maintains both the absolute time commands (ATC) and relative time commands (RTC) stored command tables and passes stored commands to the MC task for execution when their time-tags indicate that they should be executed. The MPS will load a time sequence buffer with either ATCs or RTCs, verify the load, and sequentially process each command without regard to time.

Public data:

Time Sequence Buffer	Table of commands (both RTC and ATC format).

Public functions:

sc_load_tbl	Loads stored command table.

Internal functions:

See Figure 3-27

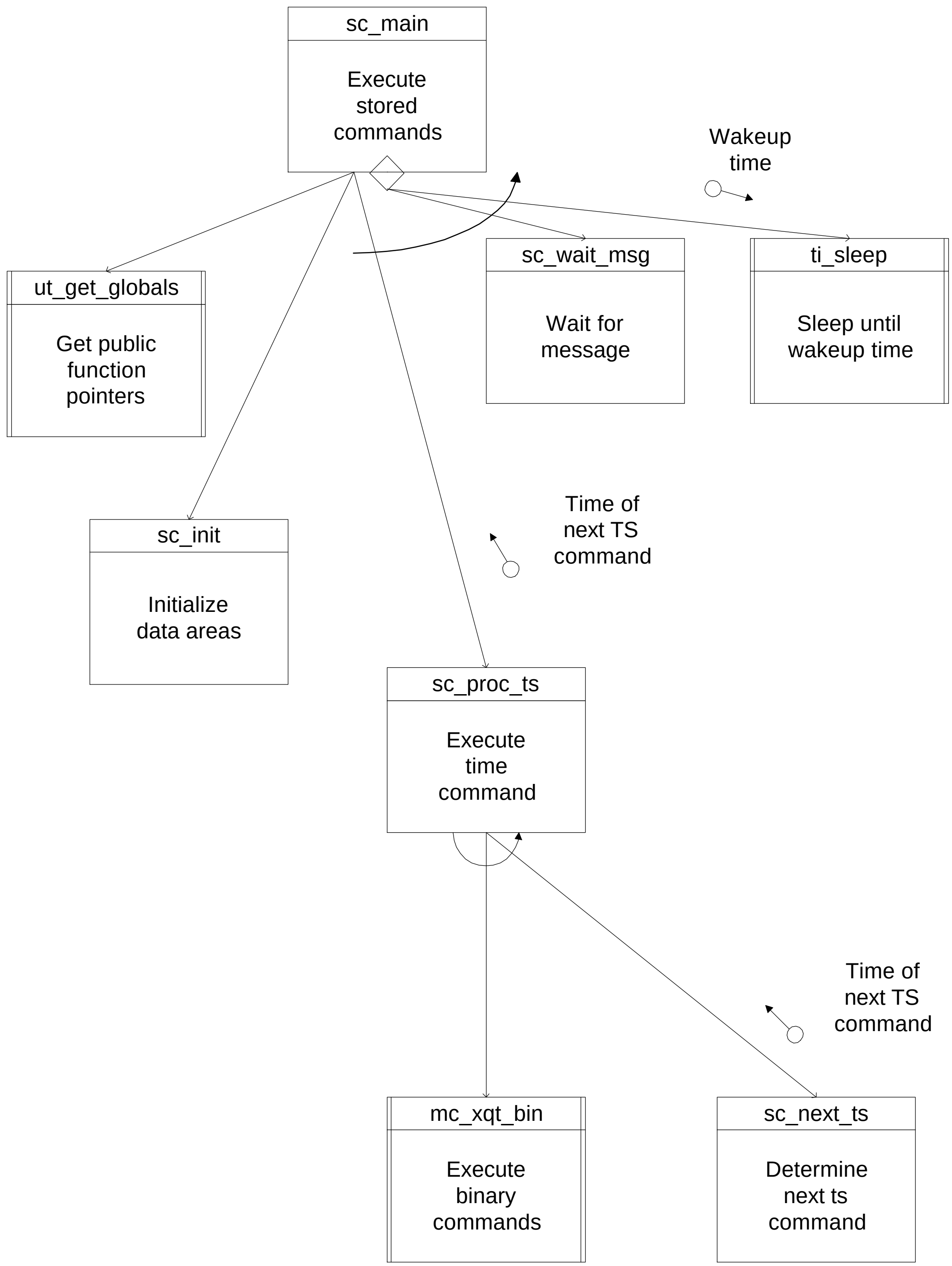

Figure 3-27. Stored Commands Task

3.3.3.4 SM - Software Manager

The MPS SM task is responsible for supporting SCC memory/table loads and dumps. Loads are supported via SM's public functions (called by the CI task to support a memory or table load).

The internal function of the SM task is to support the dump of the loaded data by communicating with the telemetry subsystem (TO task).

Public data:

Working Storage Buffer — Table which contains either a memory or table load.

Public functions:

sm_load_mem — Loads memory.

sm_load_tbl — Loads a table.

Internal functions:

See Figure 3-28

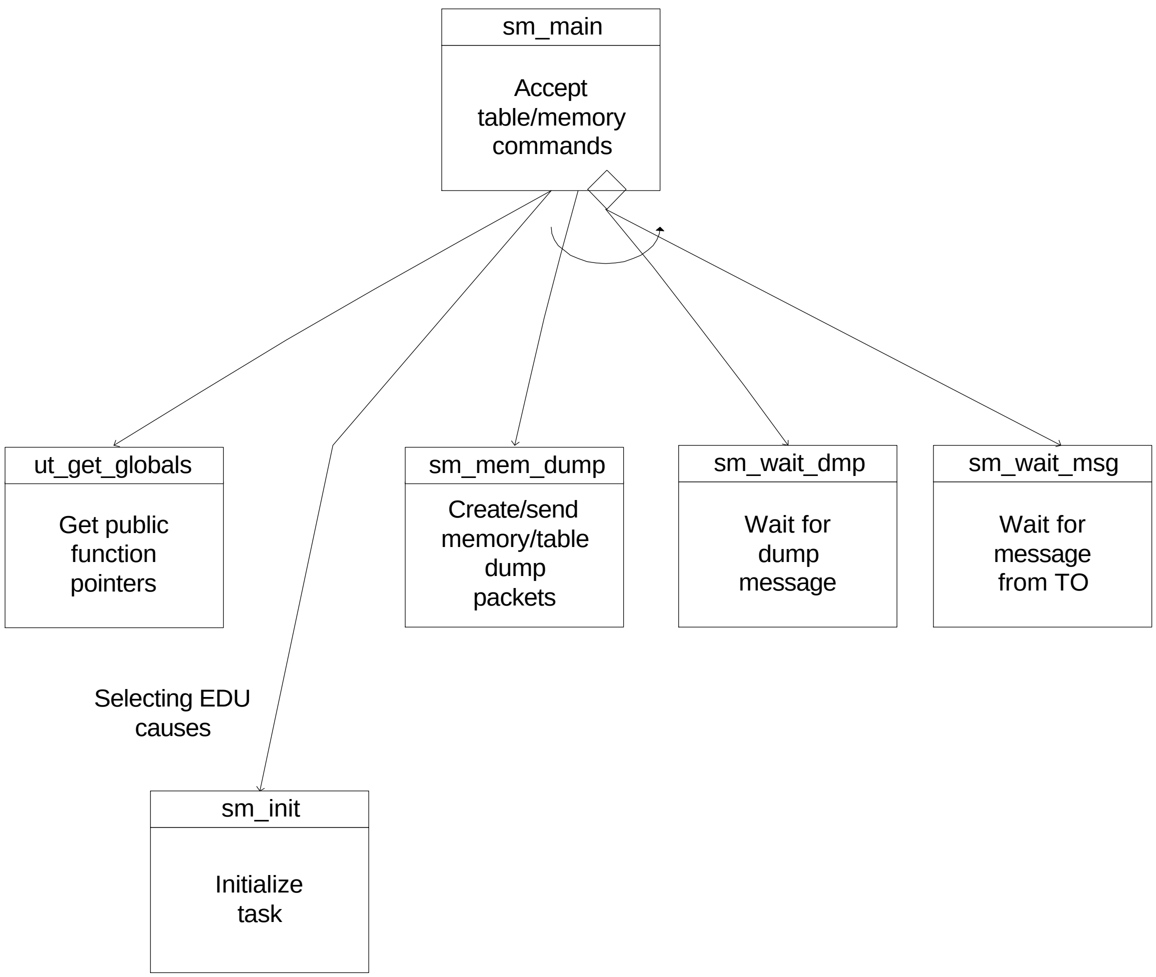

Figure 3-28. Software Manager Task

3.3.4 Telemetry subsystem

The telemetry subsystem consists of three tasks:

MT	Manage Telemetry	Read in PDB telemetry information
GT	Generate Telemetry	Generate telemetry packets
TO	Telemetry Output	Format real-time telemetry data for output

3.3.4.1 MT - Manage Telemetry

The MT task reads in PDB telemetry parameter data and makes it available to the rest of the system. The data is read in at initialization. After initialization, the task becomes inactive.

Public data:

Telemetry Packet Table	Table of AM-1 packet information extracted from PDB.
Telemetry Parm Table	Table of AM-1 telemetry information extracted from PDB.

Public functions:

mt_cvt_eu	Converts a raw telemetry value to engineering units.
mt_get_table	Returns a pointer to the telemetry parameter table.
mt_get_tlm	Gets a telemetry value given the mnemonic.
mt_set_tlm	Sets a telemetry value given the mnemonic.
mt_upd_tlm	Updates a telemetry value given the descriptor.
mt_get_pkt	Gets the contents of a given telemetry packet.

Internal functions:

See Figure 3-29

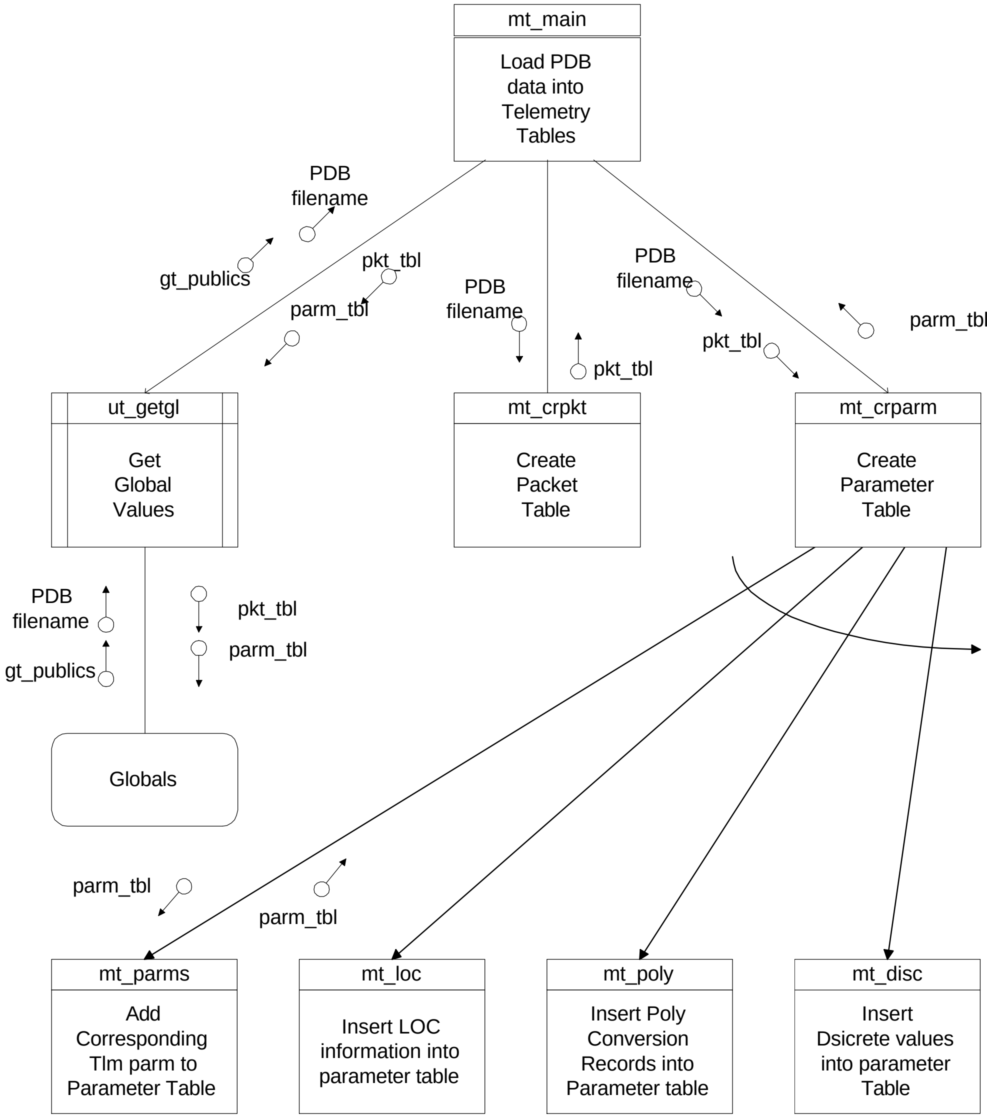

Figure 3-29. Manage Telemetry Task

3.3.4.2 GT - Generate Telemetry

The GT task formats all AM-1 packets based on information from the PDB. Once packet initialization is complete, updates to the packets will occur as a result of the modeling subsystem settings or end-item verifier processing. These updates will be supported by GT's public functions.

GT's internal function only supports the interleaving of activity log data and the extracted H/K data provided by the ETS HRS.

Public data:

AM-1 Health and Safety Packet

AM-1 H/K Packet

AM-1 1 Kbps Diagnostic Packet

AM-1 16 Kbps Diagnostic Packet

Public functions:

gt_get_apid	Returns a packet address given the packet APID.
gt_get_pkt	Returns a packet address given the packet mnemonic.
gt_set_mode	Sets the output mode to the given value.
gt_set_pkt	Sets a value in a packet given the packet mnemonic and a location within the packet.
gt_setparm	Sets a telemetry parameter value based on operator input.
gt_intvl	Sets sample rates for APIDs.
gt_update	Updates packet with passed value.

Internal functions:

See Figure 3-30

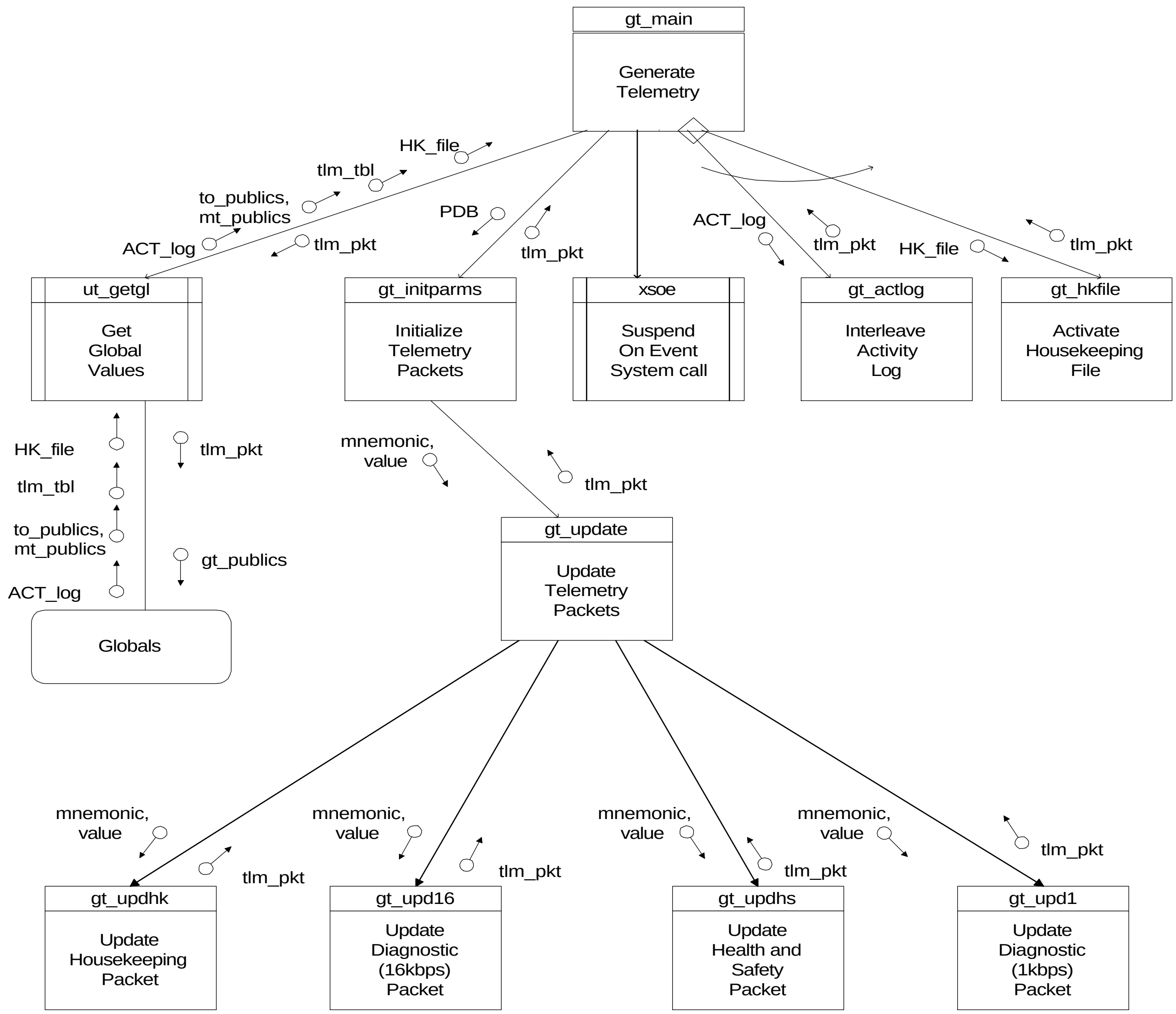

Figure 3-30. Generate Telemetry Task

3.3.4.3 TO - Telemetry Output

The TO task generates two real time streams of AM-1 telemetry using the packets provided from the GT task. Depending on the MPS simulation mode, the TO task creates either VCDUs or EDUs from the available packets, and transmits them to either the EDOS or EOC, respectively.

Public data:

Channel 1 VCDU

Channel 2 VCDU

Channel 1 EDU

Channel 2 EDU

CLCW

Public functions:

to_drop_vc	Drops VCDUs from the given virtual channel.
to_start	Starts telemetry.
to_stop	Stops telemetry.
to_tlmmode	Sets mode of telemetry output.
to_update_clcw	Updates current CLCW.
to_tlmcfg	Sets telemetry transmission configuration.
to_tlmhdr	Sets the telemetry header.
to_tlmerr	Insert bit errors in telemetry frame.
to_enablers	Turn on/off R/S encoding

Internal functions:

See Figure 3-31

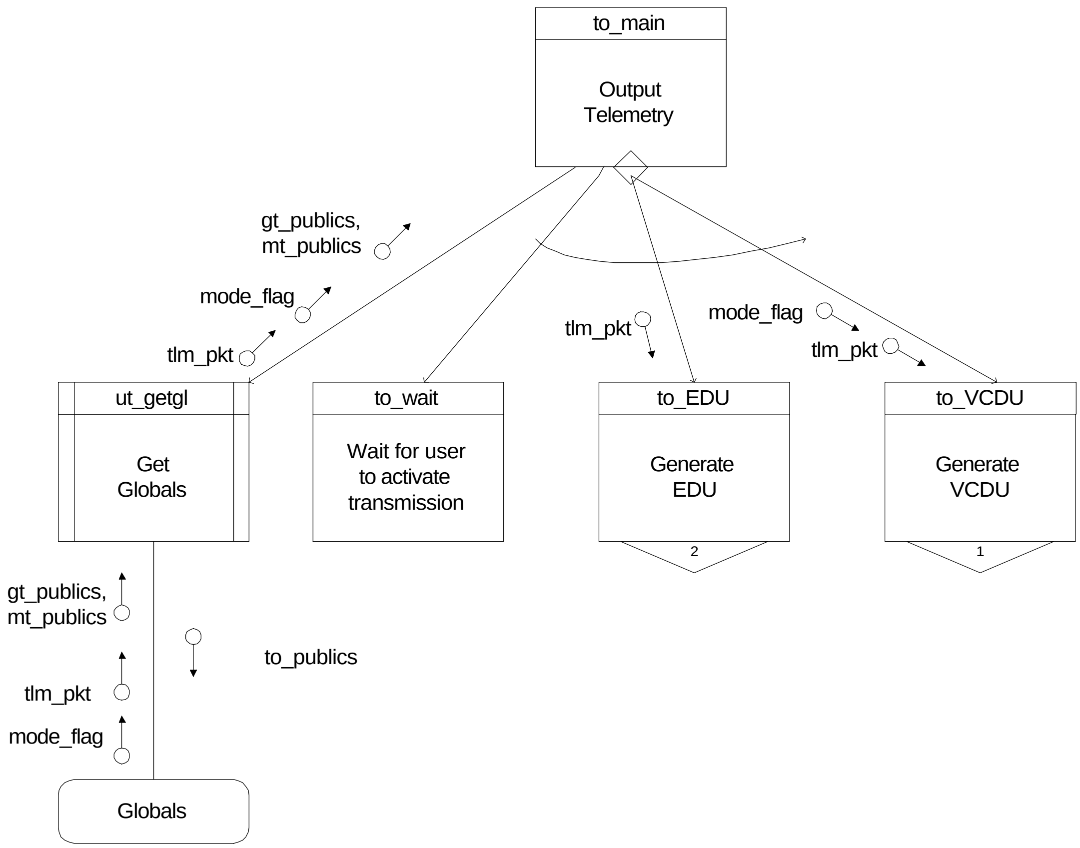

Figure 3-31. Telemetry Output Task (1 of 3)

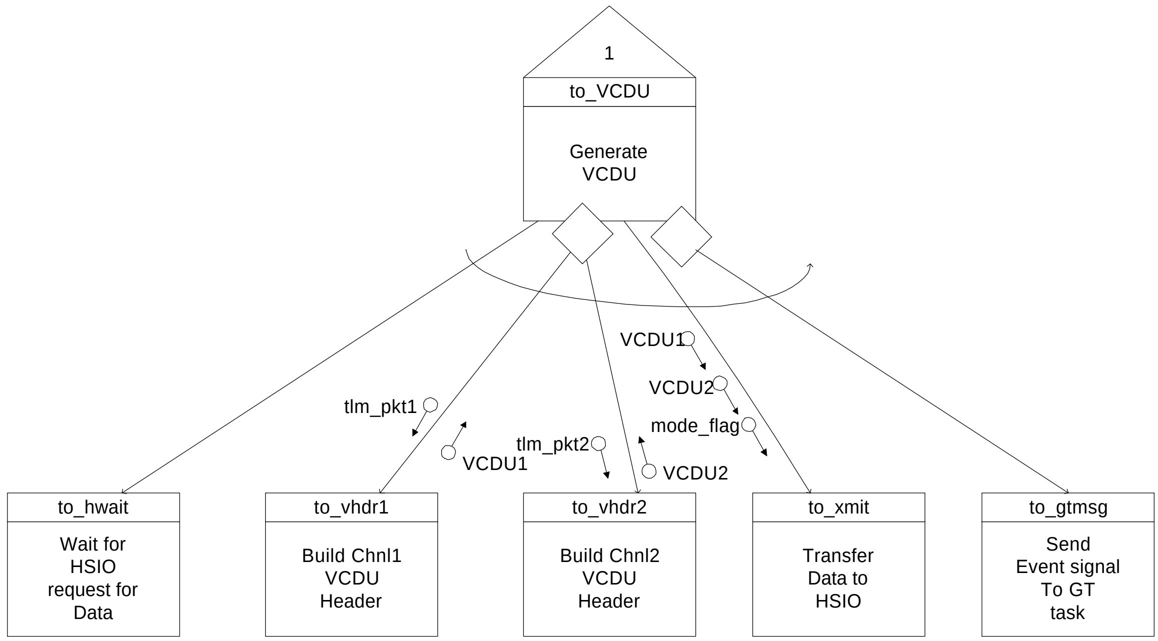

Figure 3-31. Telemetry Output Task (2 of 3)

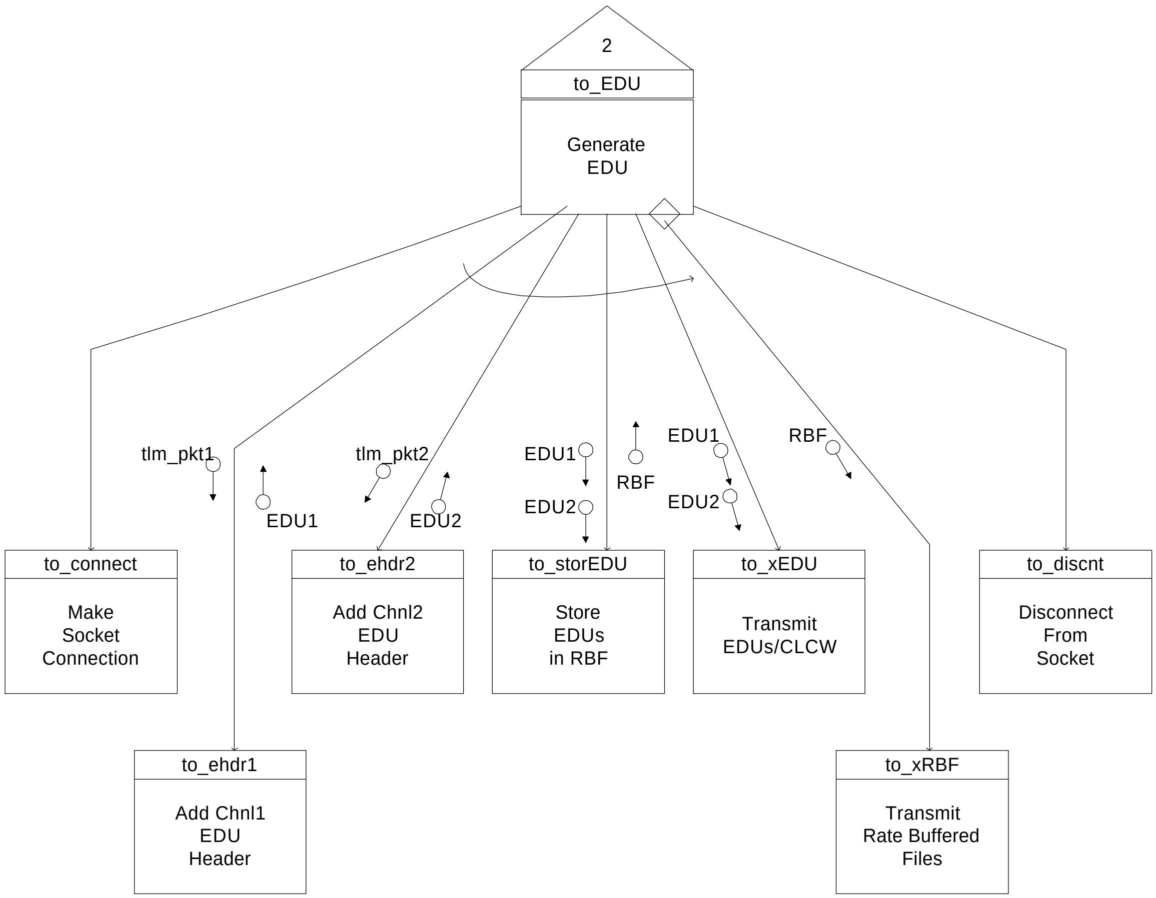

Figure 3-31. Telemetry Output Task (3 of 3)

3.3.5 Modeling Subsystem

The modeling subsystem consists of the following tasks:

TI	Keep Time	Maintain spacecraft time
MO	Modeling	Generate telemetry parameter values using tables or functions

3.3.5.1 TI - Keep time

The TI task is responsible for maintaining spacecraft time and providing timing and time-tag services to the other tasks. It waits on a predefined PDOS software interrupt that takes place at regular intervals, whereupon it updates the various time values according to the elapsed time.

Public data:

Spacecraft Time

GMT

Public functions:

ti_sleep	Puts the calling task to sleep until the spacecraft time indicated.
ti_set_gmt	Sets the GMT to the value given.
ti_set_sc	Sets the spacecraft time to the value given.
ti_delay	Puts the calling task to sleep for the given amount of time.
ti_get_gmt	Returns the current GMT value.
ti_get_sc	Returns the current spacecraft time value.

Internal functions:

See Figure 3-32

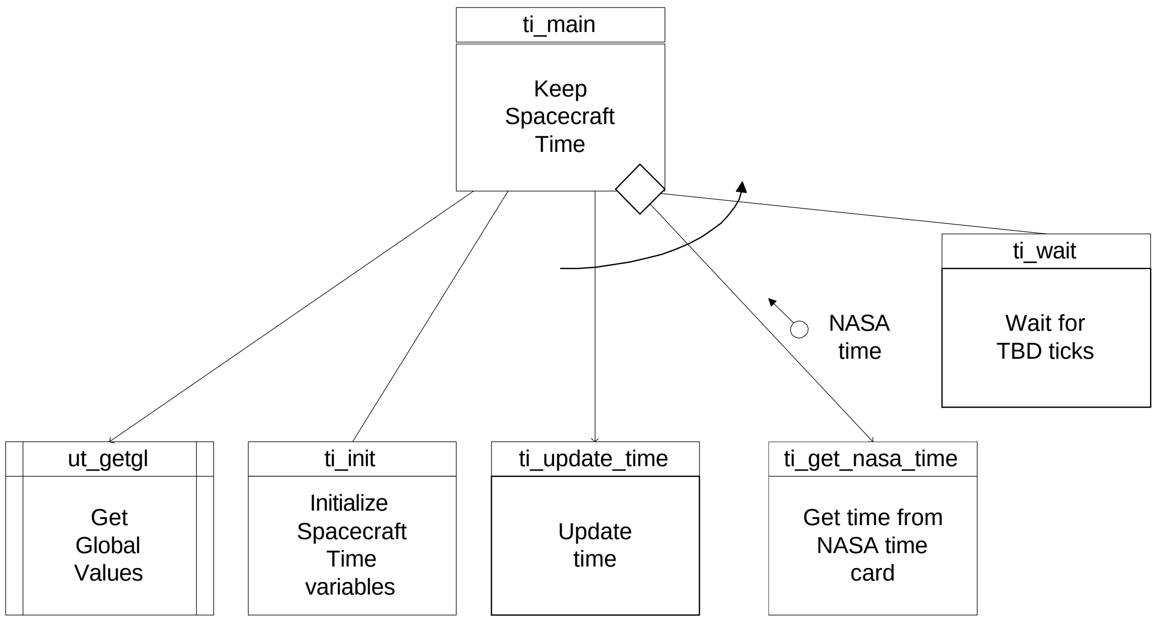

Figure 3-32. Keep Time Task

3.3.5.2 MO - Modeling

The MO task is responsible for setting telemetry parameters based on table and algorithm information supplied at initialization or by the operator.

Public data:

Algorithm Table	Table of telemetry mnemonics, each of which is associated with a trigonometric function.
Discrete Value Table	Table of telemetry mnemonics, each of which is associated with a table of lookup values.

Public functions:

mo_active	Turn modeling on or off for the selected parameter.

Internal functions:

See Figure 3-33

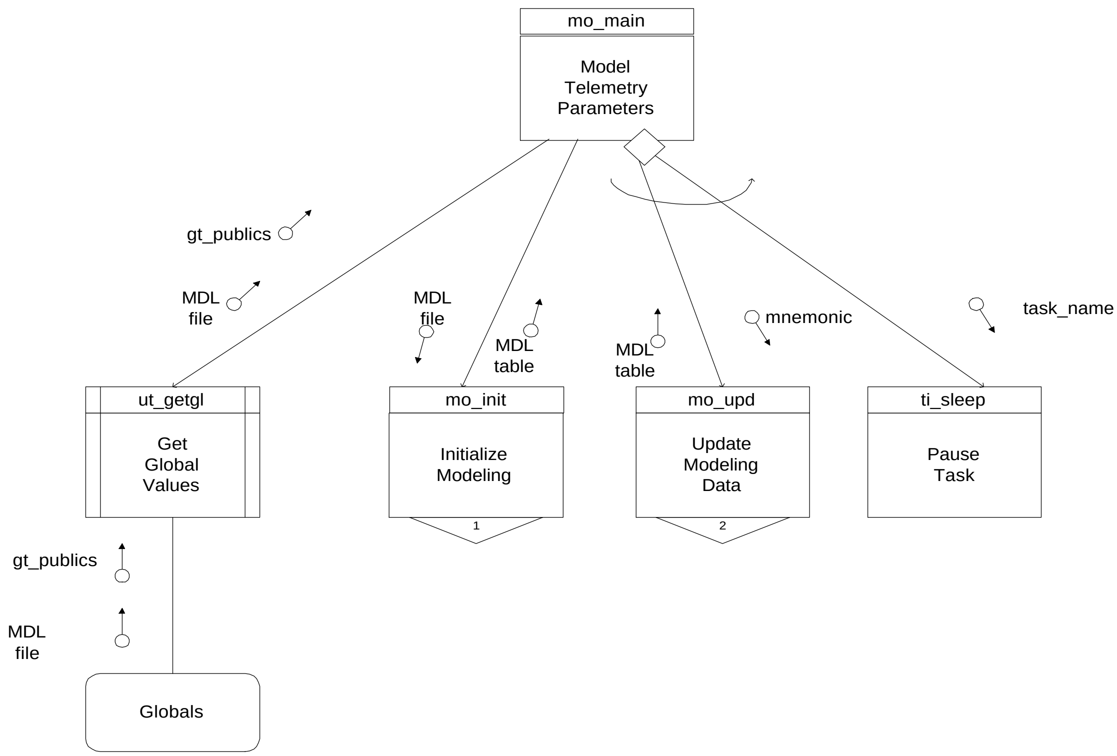

Figure 3-33. Model Task (1 of 3)

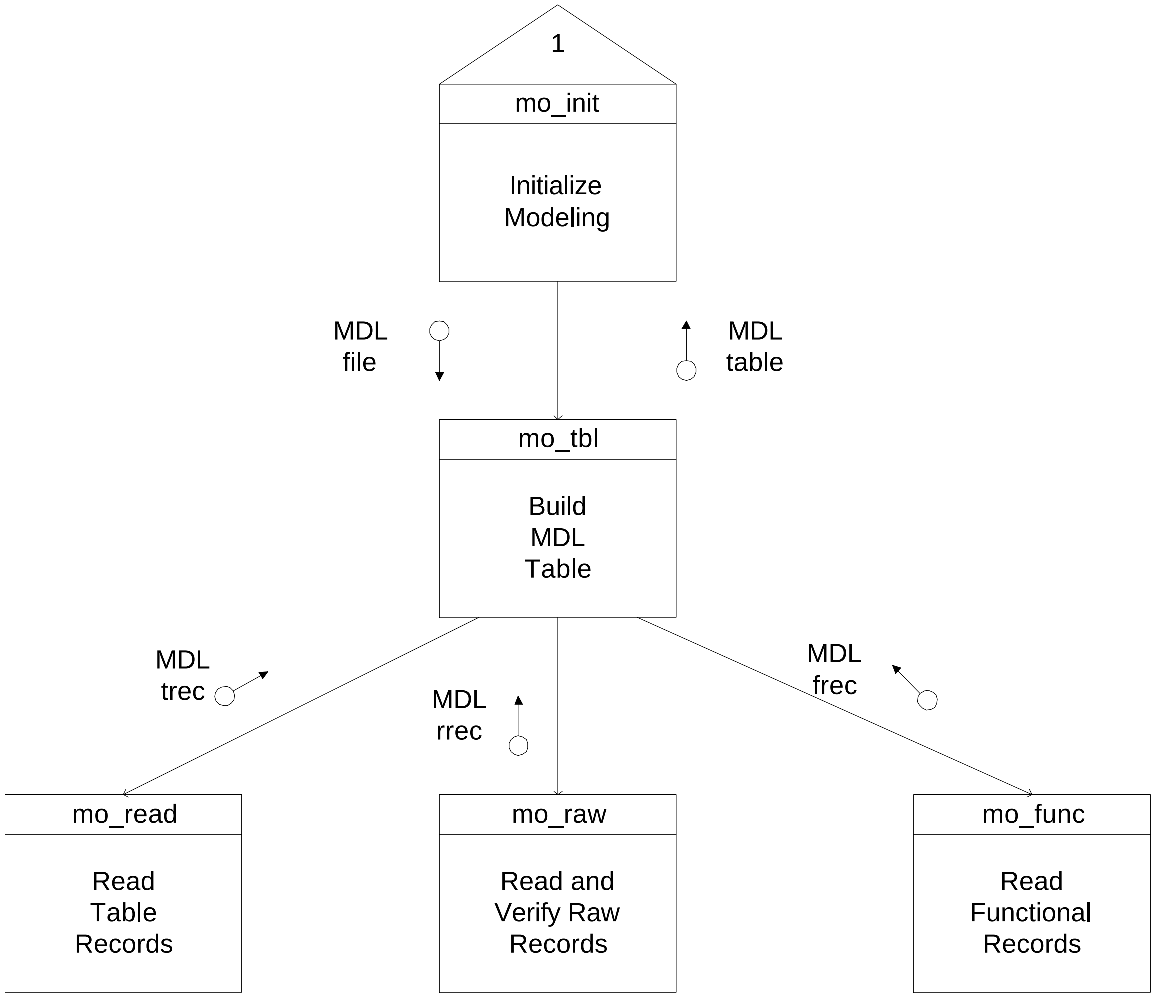

Figure 3-33. Model Task (2 of 3)

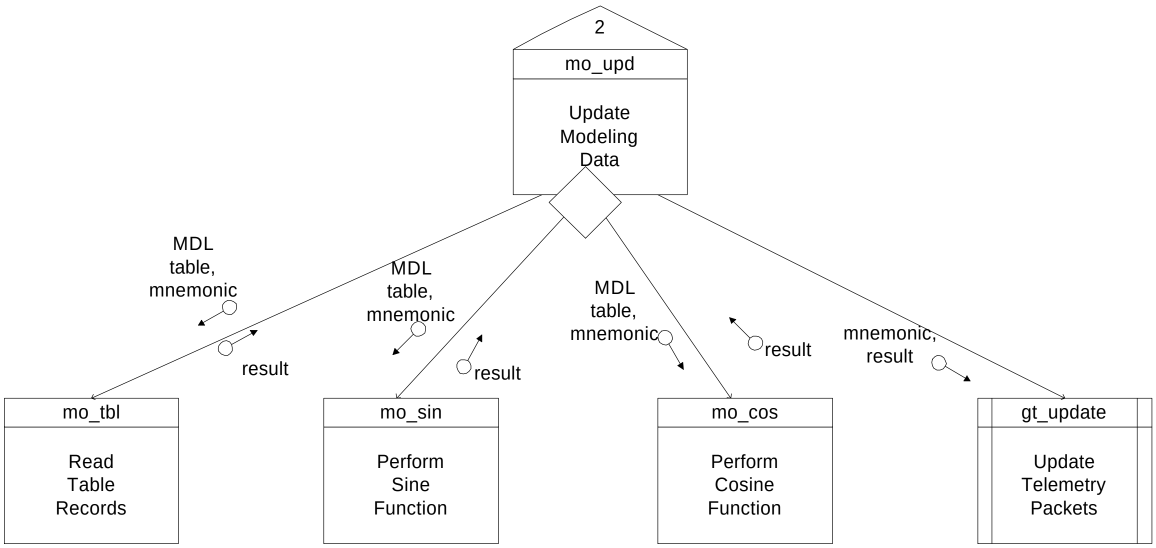

Figure 3-33. Model Task (3 of 3)

3.5 Utilities SWCI Design

The MPS Utilities SWCI consists of an offline process which converts the PDB files to a VME readable format. The capability to convert the files will be performed by running an independent task that resides on the MVME187. The Convert PDB Files process is an independent, offline function that creates internal MPS database files that are used to generate telemetry and verify commands during MPS operations. The Convert PDB files dialog box is reviewed below.

3.5.1 Convert PDB Files

The Convert PDB Files process is independent of other MPS SWCIs.

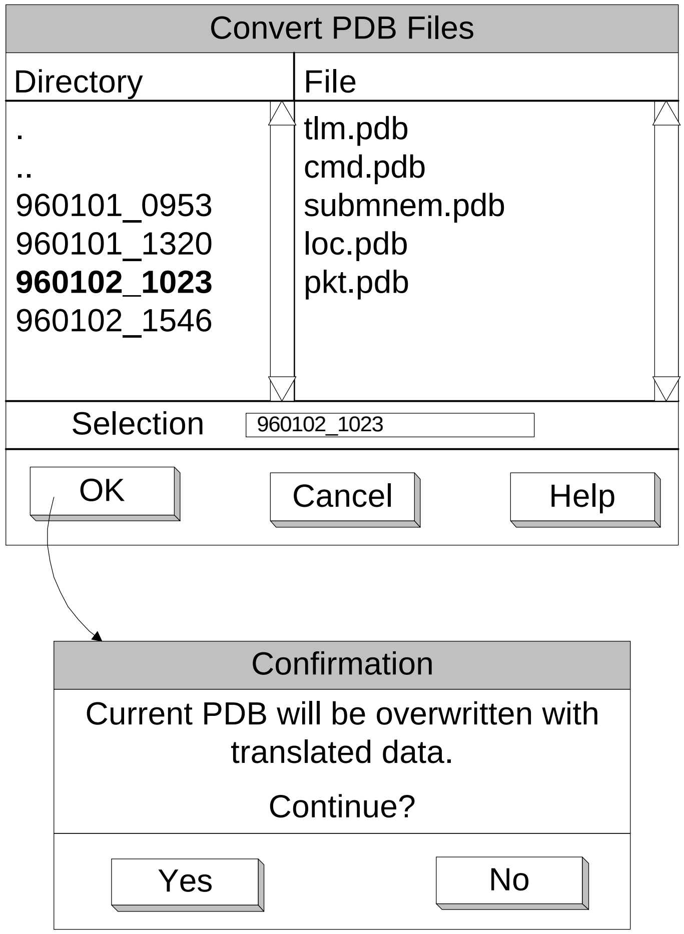

Figure 3-34. Convert PDB Main Window

This dialog box shown in Figure 3-34 differs from a normal file selection box in that the directory is the focus. Selecting a directory and selecting **OK** causes all PDB files in that directory to be converted. A confirmation dialog box reminds the operator that the files will be modified and asks to continue. Selecting **Yes** continues with the conversion process; it also closes the window and stops the PDB conversion process when the process is complete. Selecting **No** closes the window and stops the PDB conversion process.

Upon completion, the converted files are transferred from the MVME 187/Unix System V platform to the MVME 167 via FTP.

3.6 OMDSIM SWCI Design

The ETS/LRS OMDSIM will be reused in its entirety without modification. The OMDSIM's CODA contents will not be dependent on any Spacecraft Sim SWCI telemetry. The only coordination requested, between the MPS Spacecraft Sim SWCI and the MPS OMDSIM, is that CODA transmission be started and stopped with EDU transmission. This will be provided by a start/stop message from the MPS Telemetry Output task. CODAs will then be generated by the MVME 187, once every 5 seconds.

Section 4. Design Issues Resolution

As a result of a meeting on 2/22/96 with EOC developers and FOT personnel, design change recommendations contained in Section 4 of the 1/29/96 review copy of the ETS DDS Volume 2 will be accepted with minor revisions. The following reflects the wording changes made to Appendix A of this document and Appendix A of the ETS Integration Test Procedure/Plans document in light of this meeting.

The following are the modified Modeling requirements:

V2-31110-15: MPS shall provide modeling files which define static, table, or algorithm files based on telemetry mnemonic.

V2-32120-15: MPS shall set initial telemetry parameter values as identified in a previously defined telemetry initialization table.

V2-33200-02: MPS shall provide the operator with the capability to access model tables, giving values for each AM-1 telemetry parameter at specific points in time.

V2-33200-03: MPS shall provide the operator with the capability to access model functions and coefficients.

The following requirement has been removed due to lack of detail:

V2-32310-12: MPS shall be capable of transmitting command echo blocks (TBD).

The following Solid State Recorder (SSR) requirement has been removed due to lack of need:

V2-32120-16: MPS shall execute telemetry directives that control the MPS solid state recorders.

As per the discussion on 2/22/96, there will be no SSR emulation, i.e., no SSR data manipulation will be supported by the ETS MPS software. An operational procedure (combination of modeling and scenario files) will generate real-time statistics. In addition, telemetry directives which provide dataset management, i.e., close, playback, retransmit, release, etc., will not be supported for either H/K buffer dumps or Science Recorder management.

The following SSR requirement has been expanded to specify exactly what will be included in the playback stream. This requirement applies to the MPS running in the spacecraft simulation mode.

V2-32440-12: MPS shall simulate playback of recorded telemetry data by generating and transmitting a static H/K packet in response to a Dump H/K buffer command.

Based on the change noted above, the following spacecraft simulation mode requirement has been removed:

V2-32440-11 MPS shall provide for the storage of housekeeping telemetry to be used as playback data.

The capability to store generated EDUs of housekeeping data and to transmit the files via KFTP are included in the MPS EDOS simulation mode as requirements V2-32430-04 and V2-32430-08, respectively.

The following "data conversion" requirements have been removed due to interface changes:

V2-31130-02: MPS shall accept directives that control the editing, reporting, and conversion of an ASCII formatted Model Parameters file to a binary format of the Model file.

V2-31130-03: MPS shall accept directives that control the editing, reporting, and conversion of resident telemetry files into CCSDS packet formats.

V2-33200-01: MPS shall convert ASCII-formatted modeling files into a binary format which can be processed by Simulate S/C C&DH and EDOS Processing SWCI.

The following requirements have been removed due to lack of need:

V2-31110-09 MPS shall build and forward simulation configuration commands that set packet intervals (sample rate) for all real-time APIDs generated by the AM-1 spacecraft.

V2-32120-06 MPS shall execute telemetry directives that set sample rates for real-time APIDs generated by the AM-1 spacecraft.

Appendix A. MPS Requirements to Design Traceability

This matrix provides a "forward" reference from the MPS level 4 detailed requirements to design element names as identified in this detailed design document. The matrix is in level 4 requirement number order in the following format:

- **Level 4 Req Spec**—MPS level 4 requirement specification number contained in the ETS Requirements Specification Volume 2 document, dated 10/31/95 (Reference 3).

  NOTE: A requirement tagged as [MOD] represents a requirement which has been modified from the level 4 requirement contained in the requirements specification document. A requirement tagged as [DEL] represents a requirement which has been deleted from the requirements specification document. Until the requirements specification document (Reference 3) is updated, the level 4 requirements given in the following table supersede those given in the requirements specification document.

- **Requirement description—**The textual representation of this requirement as written in the ETS Requirements Specification Volume 2 document. (See note above concerning requirements tagged as [MOD] or [DEL].)

- **Design Element —** Unit name identified within the structure charts or public function lists described for each MPS task. All units which are subordinate to identified unit will assume the parent's requirement mapping.

- **Build —** The MPS build which satisfies this requirement.

NOTE: Each volume in this five-volume set contains its own level 4 traceability matrix which provides a design cross reference for the design elements contained in that specific volume.

Level 4 Req Spec	Requirement Description	Design Element	Build
V2-31100-01	MPS shall acknowledge operator request within 2 seconds of its entry.	Mps_User_IF.c	1
V2-31100-02	MPS shall start execution of operator request within 5 seconds of its entry.	Mps_User_IF.c	1
V2-31100-03	MPS shall accept and validate all operator directives.	Mps_User_IF.c	2
V2-31100-04	MPS shall be capable of logging all operator directives.	Mps_User_IF.c	2

Level 4 Req Spec	Requirement Description	Design Element	Build
V2-31110-01	MPS shall build and forward simulation configuration commands that define the simulation mode as either spacecraft simulation mode or EDOS simulation mode.	Mps_User_IF.c	2
V2-31110-02	MPS shall build and forward simulation configuration commands that set spacecraft time.	Mps_User_IF.c	1
V2-31110-03	MPS shall build and forward simulation configuration commands that configure command processing for either serial or blocked mode when running as a spacecraft simulator.	Mps_User_IF.c	1
V2-31110-04	MPS shall build and forward simulation configuration commands that enable or disable any element of command validation.	Mps_User_IF.c	2
V2-31110-05	MPS shall build and forward simulation configuration commands that set expected values in the command block header when validation is turned on and system is configured for blocked mode processing.	Mps_User_IF.c	1
V2-31110-06	MPS shall build and forward simulation configuration commands that set the CLCW.	Mps_User_IF.c	2
V2-31110-07	MPS shall build and forward simulation configuration commands that configure telemetry processing for either serial or blocked mode when running as a spacecraft simulator.	Mps_User_IF.c	1
V2-31110-08	MPS shall build and forward simulation configuration commands that set values in the telemetry block header when system is configured for blocked mode processing.	Mps_User_IF.c	1
V2-31110-09 [DEL]	MPS shall build and forward simulation configuration commands that set packet intervals (sample rate) for all real-time APIDs generated by AM-1 spacecraft.	n/a	n/a
V2-31110-10	MPS shall build and forward simulation configuration commands that set values of telemetry parameters (based on mnemonic).	Mps_User_IF.c	2
V2-31110-11	MPS shall build and forward simulation configuration commands to start and stop telemetry transmission for two physical channels. The selection of rate and the option to start and stop transmission shall be independent for each channel.	Mps_User_IF.c	1

Level 4 Req Spec	Requirement Description	Design Element	Build
V2-31110-12	MPS shall accept directives that result in erroneous telemetry header values.	Mps_User_IF.c	2
V2-31110-13	MPS shall accept directives that result in erroneous R/S encoding values.	Mps_User_IF.c	2
V2-31110-14	MPS shall build and forward simulation configuration commands to turn on and off selected event modeling.	Mps_User_IF.c	2
V2-31110-15 [MOD]	MPS shall provide modeling files which define static, table, or algorithm files, based on telemetry mnemonic.	Mps_User_IF.c	2
V2-31110-16	MPS shall build and forward simulation configuration commands to change between static, table, or algorithm models.	Mps_User_IF.c	2
V2-31110-17	MPS shall provide the capability to create and store simulation timelines.	Mps_User_IF.c	2
V2-31110-18	MPS shall be capable of executing simulation timelines.	Mps_User_IF.c	2
V2-31110-19	MPS shall build and forward simulation configuration commands to start and stop logging of commands and telemetry	Mps_User_IF.c	2
V2-31120-01	MPS shall build and forward display requests for all Spacecraft and EDOS telemetry configuration windows which accept user input.	Mps_User_IF.c	2
V2-31120-02	MPS shall build and forward display requests for all Spacecraft and EDOS command configuration windows which accept user input.	Mps_User_IF.c	2
V2-31120-03	MPS shall build and forward display requests for all Spacecraft and EDOS modeling windows which accept user input.	Mps_User_IF.c	2
V2-31120-04	MPS shall build and forward display requests for all Spacecraft and EDOS telemetry status displays.	Mps_User_IF.c	2
V2-31120-05	MPS shall build and forward display requests for all Spacecraft and EDOS command status displays.	Mps_User_IF.c	2
V2-31130-01	MPS shall accept directives that control the editing, reporting, and conversion of ASCII formatted PDB to binary format of the PDB.	Pdb_main.c	2
V2-31130-02 [DEL]	MPS shall accept directives that control the editing, reporting, and conversion of an ASCII formatted Model Parameters file to a binary format of the Model file.	n/a	n/a

Level 4 Req Spec	Requirement Description	Design Element	Build
V2-31130-03 [DEL]	MPS shall accept directives that control the editing, reporting, and conversion of resident telemetry files into CCSDS packet formats.	n/a	n/a
V2-31140-01	MPS shall accept directives that control the creation, modification, and transmission of OMDs. The complete requirements are contained within the Operations Management Data SWCI Requirements specification.	ETS/LRS	2
V2-31200-00	MPS shall be capable of timestamping and logging all simulator event messages.	Mps_User_IF.c	2
V2-31200-01	MPS shall display simulator event messages.	Mps_User_IF.c	1
V2-31200-02	MPS shall display received and previously transmitted OMD messages.	ETS/LRS	2
V2-31200-03	MPS shall display resident telemetry files (generated by the high rate system).	Mps_User_IF.c	2
V2-31200-04	MPS shall be capable of delogging all operator directives and simulator event messages in real time to the screen.	Mps_User_IF.c	1
V2-31200-05	MPS shall provide at least one status display to the operator, showing key information about the state and configuration of the simulator.	Mps_User_IF.c	2
V2-31200-06	MPS shall provide the capability to display to the operator command blocks, frames, and packets received.	Mps_User_IF.c	1
V2-31200-07	MPS shall provide the capability to display to the operator telemetry blocks, frames, and packets transmitted.	Mps_User_IF.c	1
V2-31200-08	MPS shall update status, data quality, and accounting information once every 10 seconds, at a minimum.	Mps_User_IF.c	2
V2-31200-09	MPS shall be capable of providing dumps of received or generated test data on electronic and physical media.	Mps_User_IF.c	2
V2-32100-01	MPS shall execute directives that identify simulation mode.	TO_tlmmode.c	1
V2-32100-02	MPS shall execute directives that set spacecraft time and GMT.	TI_PUBLIC	1
V2-32110-01	MPS shall execute directives that configure command processing for either serial or blocked mode, when running as a spacecraft simulator.	CI_setmd.c	1

Level 4 Req Spec	Requirement Description	Design Element	Build
V2-32110-02	MPS shall execute command directives that enable or disable any element of the command validation process.	CI_cmdval.c	2
V2-32110-03	MPS shall execute directives that set the expected values within the block header.	CI_cmdhdr.c	2
V2-32110-04	MPS shall execute command directives that override the CLCW.	CI_clcw.c	2
V2-32120-01	MPS shall execute telemetry directives that configure the transmission of telemetry.	TO_tlmcfg.c	1
V2-32120-02	MPS shall execute telemetry directives that start and stop telemetry transmission for two physical channels when MPS is running in spacecraft simulator mode.	TO_start.c TO_stop.c	1
V2-32120-03	MPS shall execute telemetry directives that start and stop transmission of EDOS formatted data through EBnet.	TO_start.c TO_stop.c	2
V2-32120-04	MPS shall execute telemetry directives that configure telemetry generation for either serial or blocked mode when running as a spacecraft simulator.	TO_tlmcfg.c	1
V2-32120-05	MPS shall execute telemetry directives that set values in the telemetry block header when MPS is configured for Nascom blocked mode processing.	TO_tlmhdr.c	2
V2-32120-06 [DEL]	MPS shall execute telemetry directives that set sample rates for all real-time APIDs generated by the AM-1 spacecraft.	n/a	n/a
V2-32120-07	MPS shall execute telemetry directives that set the value of any telemetry parameter by either mnemonic or position within a packet.	MT_settlm.c	2
V2-32120-08	MPS shall execute telemetry directives that set the value of any location in the MPS-simulated spacecraft memory.	MT_settlm.c	2
V2-32120-09 [MOD]	MPS shall execute telemetry directives that request the value of any telemetry parameter in raw units.	MT_gettlm.c	2
V2-32120-10	MPS shall execute telemetry directives that request the contents of any telemetry packet.	MT_getpkt.c	2
V2-32120-11	MPS shall execute telemetry directives that request the value of any location or block of locations in spacecraft memory.	SM_PUBLIC	2

Level 4 Req Spec	Requirement Description	Design Element	Build
V2-32120-12	MPS shall execute telemetry directives that insert bit errors in telemetry frames, virtual channel data units (VCDUs), and packets.	TO_tlmerr.c	2
V2-32120-13	MPS shall execute telemetry directives that turn on and off R/S error generation.	TO_enablers.c	2
V2-32120-14	MPS shall execute telemetry directives that force telemetry dropouts of one or more VCDUs on a given VC either once or in a repeated pattern.	TO_drop_vc.c	2
V2-32120-15 [MOD]	MPS shall set initial telemetry parameter values as identified in a previously defined telemetry initialization table.	MT_settlm.c	2
V2-32120-16 [DEL]	MPS shall execute telemetry directives that control the MPS Solid State Recorder.	n/a	n/a
V2-32130-01	MPS shall execute modeling directives that enable or disable selected event modeling.	MO_active.c	2
V2-32130-02	MPS shall execute modeling directives that set the spacecraft time and the GMT.	TI_PUBLIC	1
V2-32130-03	MPS shall execute modeling directives that associate any telemetry parameter with any predefined model.	MO_active.c	2
V2-32130-04	MPS shall execute modeling directives that change between static, table, or algorithm models.	MO_active.c	2
V2-32210-01	MPS shall use the universal time coordinated (UTC) format for time-of-day -related data, as required by the test.	TI_PUBLIC	1
V2-32210-02	MPS shall maintain spacecraft mission elapsed time (MET).	TI_PUBLIC	1
V2-32210-03	MPS shall be capable of receiving the serial time code as provided by NASA36 time distribution.	TI_PUBLIC	1
V2-32210-04	MPS shall be capable of initializing the MET to Greenwich mean time (GMT).	TI_PUBLIC	1
V2-32210-05	MPS shall set, adjust, and operate the spacecraft clock as commanded.	TI_PUBLIC	2
V2-32220-01	MPS shall calculate specified telemetry parameters based on a function or table associated with the parameter and spacecraft time elapsed since simulator startup or receipt of a specific command.	MO_Main.c	2
V2-32220-02	MPS shall be capable of changing the function or table used to model a dynamic telemetry parameter.	MO_active.c	2

Level 4 Req Spec	Requirement Description	Design Element	Build
V2-32310-01	The MPS, when acting as a spacecraft, shall comply with the command data formats and protocols specified in the TGT to EDOS interface document.	CI_init.c	1
V2-32310-02	The MPS, when acting as a DSN, shall comply with the command data formats and protocols specified in the EBnet to EDOS interface document for the DSN to EDOS interface.	CI_init.c	1
V2-32310-03	The MPS, when acting as a GN, shall comply with the command data formats and protocols specified in the EBnet to EDOS interface document for the GN to EDOS interface.	CI_init.c	1
V2-32310-04	The MPS, when acting as a WOTS, shall comply with the command data formats and protocols specified in the EBnet to EDOS interface document for the WOTS to EDOS interface.	CI_init.c	1
V2-32310-05	The MPS interface with EOC shall comply with the interface formats and protocols specified in the EOC to EDOS interface document.	CI_init.c	2
V2-32310-06	MPS shall be capable of receiving command data in serial form from the Radio Frequency (RF) SOC.	CI_Init.c	1
V2-32310-07	MPS shall be capable of receiving command data in serial form from EDOS at rates specified in EOS AM-1 ICD.	CI_Init.c	2
V2-32310-08	MPS shall be capable of receiving and validating DSN, GN, or WOTS 4800-bit Nascom blocks containing commands from EDOS at rates specified in EOS AM-1 ICD.	CI_Bld_Tf.c	1
V2-32310-09	MPS shall be capable of receiving command data as UPD command blocks from the EOC via EBnet.	CI_Init.c	2
V2-32310-10	MPS shall perform polynomial error protection (PEP) checks and report PEP errors when detected.	CI_Tf_Check.c	2
V2-32310-11	MPS shall provide the capability to respond to all C&DH spacecraft commands as defined in the AM-1 PDB.	CI_Main.c	2
V2-32310-12 [DEL]	MPS shall be capable of transmitting command echo blocks (TBD).	n/a	n/a

Level 4 Req Spec	Requirement Description	Design Element	Build
V2-32310-13	MPS shall be capable of receiving command data blocks from EOC.	CI_Main.c	2
V2-32310-14	MPS shall be capable of logging up to 8 MB of received commands during a testing session.	CI_Tf_Check.c	2
V2-32320-01	MPS shall simulate spacecraft command acceptance according to the COP-1 protocol.	CI_Tf_Check.c	1
V2-32320-02	MPS shall provide the capability to read and interpret flags in all headers.	CI_Bld_Tf.c	1
V2-32320-03	MPS shall provide the capability to validate the all headers of received data	CI_Tf_Check.c	1
V2-32320-04	MPS shall provide the capability to monitor and display command processing status.	CI_Tf_Check.c	2
V2-32320-05	MPS shall store received commands for posttest review subject to specified storage capacities.	CI_Tf_Check.c	2
V2-32320-06	MPS shall simulate spacecraft command validation processing using information from the PDB.	MC_Xqt.c	2
V2-32320-07	MPS shall generate a simulator event message whenever a valid command is received.	MC_Xqt.c	1
V2-32320-08	MPS shall generate a simulator event message whenever a command error is detected.	CI_Tf_Check.c	1
V2-32330-01	MPS shall simulate execution of stored absolute time commands (ATC) and relative time command (RTC) sequences.	SC_Main.c	2
V2-32330-02	MPS shall process stored commands, sequentially, without regard to spacecraft clock values.	SC_Main.c	2
V2-32340-01	MPS shall provide the capability to verify the received EOS spacecraft commands by updating the command link control word (CLCW) based on the receipt of a valid command.	MC_Xqt.c	1
V2-32340-02	MPS shall receive spacecraft memory and table loads from the EOC and shall store the load data in the simulated memory.	SM_Tload.c, SM_Mload.c	2
V2-32340-03	MPS shall perform a CRC validation in the load data and shall set a pass/fail indicator in telemetry.	SM_Tload.c	2
V2-32340-04	MPS shall process commands that request or configure for a Spacecraft Controls Computer memory dump.	SM_Main.c	2
V2-32340-05	MPS shall process commands that set or adjust the spacecraft clock and GMT.	TI_PUBLIC	1

Level 4 Req Spec	Requirement Description	Design Element	Build
V2-32340-06	MPS shall provide the capability of setting telemetry end-item verifiers on receipt of valid commands.	MC_Xqt.c	2
V2-32410-01	MPS shall provide the capability to simulate EOS AM-1 low-rate spacecraft return-link data.	MT_Main.c, GT_Main.c	1
V2-32410-02	MPS shall provide the capability to generate simulated time codes in the packet headers.	GT_Main.c	1
V2-32410-03	MPS shall provide the capability to generate and transmit up to 512 unique APIDs.	MT_Main.c	2
V2-32410-04	MPS shall format telemetry parameters into packets as specified in the AM-1 PDB packet definitions which include the definition of housekeeping, health and safety, and diagnostic dump packets.	GT_Main.c	2
V2-32410-05	MPS shall provide the capability to insert fill data into generated test data.	TO_Vcdu.c	1
V2-32410-06	MPS shall send out telemetry packets at specified intervals in spacecraft time or in response to specific events. The specified intervals (sample rates) shall exist for each AM-1 packet ID and shall be modifiable by the operator.	GT_Main.c	2
V2-32410-07	MPS shall be capable of inserting errors into any field in a telemetry packet header.	TO_tlmhdr.c	2
V2-32410-08	MPS shall be capable of simulating SCC memory dumps. MPS shall build packets based on the contents of the simulated SCC memory.	GT_Main.c	2
V2-32420-01	MPS shall provide the capability to accept AM-1 housekeeping data by electronic transmission and by physical media.	GT_Main.c	2
V2-32420-02	MPS shall be capable of generating low-rate telemetry based on external AM-1 housekeeping data.	GT_Main.c	2
V2-32420-03	MPS shall generate housekeeping packets based on spacecraft activity log entries extracted from user-provided files.	GT_Main.c	2
V2-32430-00	MPS shall provide capability to transmit up to two streams of telemetry.	GT_Main.c	1
V2-32430-01	MPS shall generate EDUs and EDOS data headers based on the User Datagram Protocol (UDP) format defined in the EDOS External ICD Data Format Control Document (Reference 5).	TO_Edu.c	2

Level 4 Req Spec	Requirement Description	Design Element	Build
V2-32430-02	MPS shall allow modification of any field within the EDOS data header.	TO_tlmhdr.c	2
V2-32430-03	MPS shall provide the capability of transmitting the CLCW in the form of EDUs to EOC through EBnet.	TO_Edu.c	2
V2-32430-04	MPS shall provide for the storage of EDUs which will be later used during the testing session for rate buffered file transmission, provided the user selected the rate buffered file telemetry generation mode.	TO_Edu.c	2
V2-32430-05	MPS shall transmit EDUs on an as built basis.	TO_Edu.c	2
V2-32430-06	MPS shall append an end of file record at the end of each rate buffered file.	TO_Edu.c	2
V2-32430-07	MPS shall provide the capability to transmit EDUs using the UDP.	TO_Edu.c	2
V2-32430-08 [MOD]	MPS shall provide the capability to transmit rate buffered files via KFTP either at the end of a simulated contact or request.	TO_Edu.c	2
V2-32430-09	The MPS interface with the EOC shall comply with the interface formats and protocols specified in the EOC to EDOS interface document.	TO_Edu.c	2
V2-32430-10	The MPS shall provide the capability to store up to 8MB of transmitted EDUs.	TO_Edu.c	2
V2-32440-01	The MPS when acting as a spacecraft shall comply with the telemetry data formats and protocols specified in the TGT to EDOS interface document.	TO_Vcdu.c	1
V2-32440-02	The MPS, when acting as a DSN, shall comply with the telemetry data formats and protocols specified in the EBnet to EDOS interface document for the DSN to EDOS interface.	TO_Vcdu.c	1
V2-32440-03	The MPS, when acting as a GN, shall comply with the telemetry data formats and protocols specified in the EBnet to EDOS interface document for the GN to EDOS interface.	TO_Vcdu.c	1
V2-32440-04	The MPS, when acting as a WOTS, shall comply with the telemetry data formats and protocols specified in the EBnet to EDOS interface document for the WOTS to EDOS interface.	TO_Vcdu.c	1

Level 4 Req Spec	Requirement Description	Design Element	Build
V2-32440-05	MPS shall build two independent CADU streams from the received packets. The contents of each CADU shall conform to formats and rates identified in the EOS AM-1 ICD (Reference 4).	TO_Vcdu.c	1
V2-32440-06	MPS shall output CADU streams in SN, GN, DSN, and WOTS formats on two physical channels for all rate and mode combinations specified within the EOS AM-1 ICD (Reference 4) .	TO_Vcdu.c	1
V2-32440-07	MPS shall provide the capability to multiplex up to 31 VCDU-IDs in the same return link physical channel.	TO_Vcdu.c	2
V2-32440-08	MPS shall provide the capability to set any field in the CLCW located in the telemetry VCDU trailer.	TO_clcw.c	2
V2-32440-09	MPS shall be capable of inserting bit errors into telemetry VCDUs.	TO_vcdu.c	2
V2-32440-10	MPS shall be capable of skipping the transmission of one or more real-time VCDUs for a specific VC.	TO_vcdu.c	2
V2-32440-11 [DEL]	MPS shall provide for the storage of housekeeping telemetry to be used as playback data.	n/a	n/a
V2-32440-12 [MOD]	MPS shall simulate playback of recorded telemetry data by generating and transmitting a static H/K packet in response to a Dump H/K buffer command.	TO_vcdu.c	2
V2-32440-13	MPS shall be capable of transmitting telemetry in any of the following Nascom blocked formats: GN format DSN format WOTS format	TO_Vcdu.c	1
V2-32440-14	MPS shall be capable of transmitting telemetry in serial form.	TO_Vcdu.c	1
V2-32440-15	MPS shall be capable of inserting errors into telemetry blocks.	TO_tlmhdr.c	2
V2-32440-16	MPS shall be capable of performing Reed-Solomon encoding on VCDUs, using an interleave depth of 1.	TO_vcdu.c	1
V2-32440-17	MPS shall provide the capability to store up to 8 MB of transmitted CADUs.	TO_vcdu.c	2

Level 4 Req Spec	Requirement Description	Design Element	Build
V2-33100-01	MPS shall convert the ASCII-formatted PDB into a binary format which can be processed by the Simulate S/C C&DH and EDOS Processing SWCI.	Pdb_Main.c	2
V2-33100-02	MPS shall use the PDB to determine the APID number and length of valid AM-1 telemetry packets.	MT_Main.c	2
V2-33100-03	MPS shall use the PDB to determine the number and mnemonics of the telemetry parameters.	MT_Main.c	2
V2-33100-04	MPS shall use the PDB to define raw-to-EU conversions for telemetry parameters. MPS shall support both linear and polynomial conversions.	MT_Main.c	2
V2-33100-05	MPS shall use the PDB to determine valid AM-1 command formats.	CI_Main.c	2
V2-33100-06	MPS shall use the PDB to determine telemetry end-item verifiers for commands.	CI_Main.c	2
V2-33200-01 [DEL]	MPS shall convert ASCII-formatted modeling files into a binary format which can be processed by the Simulate S/C C&DH and EDOS Processing SWCI.	n/a	n/a
V2-33200-02 [MOD]	MPS shall provide the operator with the capability to access model tables, giving values for each AM-1 telemetry parameter at specific points in time.	MO_Main.c	2
V2-33200-03 [MOD]	MPS shall provide the operator with the capability to access model functions and coefficients.	MO_PUBLIC	2
V2-33300-01	MPS shall extract telemetry records from the external data file and shall convert the supplied data into AM-1 packets	GT_Main.c	2
V2-33300-02	MPS shall save all the converted AM-1 packets and shall make all packets available as input to MPS' telemetry generation process.	GT_Main.c	2
V2-34000-01	MPS shall provide the capability to generate and transmit OMDs through EBnet to the EOC, including Customer Operations Data Accounting (CODA) reports, at a frequency of every five seconds and at a rate of 50 kbps.	ETS/LRS	2
V2-34000-02	MPS shall provide the capability to store OMDs from the EOC during a testing session.	ETS/LRS	2

Level 4 Req Spec	Requirement Description	Design Element	Build
V2-34000-03	MPS shall generate and display OMD event messages, logging the transmission and receipt of OMD messages.	ETS/LRS	2
V2-34000-04	MPS shall log the transmission and receipt of OMD event messages to disk.	ETS/LRS	2
V2-41100-01	The MPS shall operate in an operational ambient temperature ranging from 50 degrees to 80 degrees Fahrenheit and ambient humidity ranging from 20 percent to 70 percent (noncondensing).	n/a	1
V2-41100-02	The MPS shall require a physical space of no more than 10 feet by 20 feet.	n/a	1
V2-41100-03	The MPS shall require approximately 10 kilowatts for power.	n/a	1
V2-41200-01	The MPS shall be ready for operational use within 20 minutes of power on.	n/a	1
V2-41300-01	The MPS shall comply with the set of display guidelines specified in DSTL-92-007, *Human-Computer Interface Guidelines*, August, 1992.	Notebook PC monitor	1
V2-41400-01	The MPS shall comply with security provisions specified in the *NASA Automated Information Security Handbook*, NHB 2410.9A.	MVME 187-003B	2
V2-41400-02	The MPS shall comply with the *NASA Communications (Nascom) Access Protection Policy and Guidelines*.	MVME 187-003B	2
V2-51000-01	The MPS front end shall be composed of a workstation with a graphics display terminal, and an American Standard keyboard.	Notebook PC	1
V2-51000-02	The MPS front end configuration shall provide an interface to an Ethernet network.	MVME 187-003B	1
V2-51000-03	The MPS front end configuration shall provide magnetic disk storage.	1 GB hard drive	1
V2-52000-01	The MPS shall provide a MC68040-based microprocessor and an Ethernet controller.	MVME 167-34B	1
V2-52000-02 [MOD]	The MPS shall provide input/output cards which receive commands and transmit telemetry as serial bitstreams or as 4800-bit blocks. One IO card will be required for each of the 2 EOS AM-1 S-band channels simulated by the MPS.	HSIO card(s)	2
V2-52000-03 [DEL]	The MPS shall provide for hardware-based Reed-Solomon (R/S) encoding and cyclic redundancy checksum (CRC) encoding.	n/a	n/a

Level 4 Req Spec	Requirement Description	Design Element	Build
V2-52000-04	The MPS shall provide an internal NASA 36-bit timecode source.	PC synchronizable time generator card	1
V2-52000-05	The MPS shall provide a dynamic random access memory (DRAM) module that will be used for the storage of data created by the MPS during tests.	64 MB DRAM module	1
V2-52000-06	The MPS shall provide a hard disk drive with a minimum of 1 GB that will store the program bootstrap, executable files, and other simulation environment files, such as the Project Data Base and scenario files used during tests.	1 GB hard drive	1
V2-52000-07	The MPS shall provide a physical media storage device that can be used to support the exchange of small amounts of information with external systems and for system backups and data logging.	21 GB hard drives, 8mm tape, 3.5" floppy disk drive	1
V2-52000-08	The MPS shall be portable.	n/a	1
V2-53100-01	The MPS shall provide two output serial ports and one input serial port. All serial ports shall comply with the RS-422/449 standard with differential data and clock signals.	HSIO card(s)	1
V2-53100-02 [MOD]	The MPS serial interface shall support synchronous, non-return to zero (NRZL and NRZM) formatted data.	HSIO card(s)	1
V2-53100-03	The MPS shall transmit up to two streams of CADUs through the serial interface. The maximum data output rate is 256 Kbps for one stream and 16 Kbps for the other.	HSIO card(s)	1
V2-53100-04	The MPS shall receive spacecraft commands in CLTU bitstream through the serial interface at rates from 125 bps to 10 Kbps.	HSIO card(s)	1
V2-53100-05	The MPS shall transmit up to two streams of CADUs in Nascom blocked formats through the serial interface. The maximum data output rate is 512 Kbps for one stream and 16 Kbps for the other.	HSIO card(s)	1
V2-53100-06	The MPS shall receive spacecraft commands in CLTU bitstream in Nascom blocked formats through the serial interface at 2 Kbps.	HSIO card(s)	1

Level 4 Req Spec	Requirement Description	Design Element	Build
V2-53200-01	The MPS shall provide an Ethernet interface that conforms to 10BaseT of the IEEE 802.3 standard.	MVME 187-003B, MVME 167-34B	1
V2-53200-02	The MPS shall interface with the EOC through the Ethernet interface using the Internet Protocol (IP) suite, including TCP/IP and UDP/IP.	MVME 187-003B, MVME 167-34B	2
V2-53200-03	The MPS shall receive CLTUs in command data blocks from the EOC and output EDUs (packets and CLCWs) to the EOC through the Ethernet interface. All data transfers through this Ethernet interface are based on UDP/IP protocol.	MVME 167-34B	2
V2-53200-04	The MPS shall transmit rate-buffered data files to the EOC through the Ethernet interface. All data transfers through this Ethernet interface are based on a FTP application and TCP/IP protocol.	MVME 167-34B	2
V2-53200-05	The interface between the ETS front end and simulator platform shall use FTP and TCP/IP. The information transferred through this interface includes the following: • User directives from the front end to the simulator platform • Status from the simulator platform to the front end • Event messages from the simulator platform to the front end • Test data files from the front end to the simulator platform.	MVME 187-003B, MVME 167-34B	1
V2-53300-01	The MPS shall provide two NASA 36-bit timecode interfaces with a BNC connector for each interface. The interfaces will be used to ingest external timecodes for time-stamping test data.	PC synchronizable time generator card	2
V2-53300-02	The MPS shall ingest external timecode in 1-kHz (kilo-Hertz) time interval and output timecode in either one-million second (PB1 format) or one micro-second (PB5J format) resolution internally.	PC synchronizable time generator card	2

Appendix B. MPS Files

The following are data files and descriptions that are either used or created by the MPS. The sizes of the PDB, Housekeeping and Activity Log files will be as provided to the MPS. The size estimates for all other files are based on a 3 hour simulation at a 32 Kbps transmission rate.

PDB file	This is the Project Database (PDB) file. It is an ASCII formatted file as described in the EOS AM-1 PDB Data Format Control Document (Reference 13).
Housekeeping file	This is a housekeeping telemetry file supplied by the HRS to the MPS. (Format TBD)
Activity Log file	This is an activity log file supplied to the MPS which will contain data to be interleaved into the telemetry data stream. (Format TBD)
Event Log file	This file is created by the MPS and contains an event history of an MPS simulation. Events would include commands received and operator requests. This file will contain 80 character ASCII records. Estimated file size is 80K bytes.
Error Log file	This file is created by the MPS and contains all error messages generated by the MPS. This file will contain 80 character ASCII records. Estimated file size is 20K bytes.
Command Log	This file is created by the MPS and contains all valid commands received. Depending on the simulation mode, the command log will contain either command data blocks or commands as serial data streams of acquisition sequences and CLTUs. This file will be in hexadecimal format. Estimated file size is 10K bytes.
Telemetry Log	This file is created by the MPS and contains a hexadecimal log of transmitted telemetry data. Depending on the simulation mode, the telemetry log file will contain EDUs and CLCW packets or CADUs. There will be one log for each channel. Estimated file size is 40K each.
Rate Buffered file	This file is created by the MPS and will contain EDUs created from a simulated housekeeping playback. There will be one Rate Buffered file for each channel. Estimated file size is 4Mbytes each.

Appendix C. Abbreviations and Acronyms

AOS	Advanced Orbiting Systems
APID	application process identifier
ATC	absolute time command
BER	bit error rate
C&DH	command and data handling
CCSDS	Consultative Committee for Space Data Systems
CERES	clouds and Earth's radiant energy system
CI	configuration item
CLCW	command link control word
CLTU	command link transmission unit
Cmd	command
CODA	Customer Operations Data Accounting
COP	command operations procedure
COTS	commercial off-the-shelf
CRC	cyclic redundancy checksum
CSC	Computer Sciences Corporation
DAAC	Distributed Active Archive Center
DDS	Detailed Design Specification
DEST	destination
DFCD	data format control document
DSID	data stream identifier
DSN	Deep Space Network
EBnet	EOSDIS Backbone Network
EDOS	EOS Data and Operations System
EDU	EDOS data unit
EOC	EOS Operations Center
EOS	Earth Orbiting System

EOSDIS	EOS Data and Information System
ESDIS	Earth Science Data and Information System
ETS	EOSDIS Test System
EU	engineering unit
FARM	frame acceptance and reporting mechanism
FOT	Flight Operations Team
GMT	Greenwich mean time
GN	Ground Network
GSFC	Goddard Space Flight Center
GUI	graphical user interface
hex	hexadecimal
HK or H/K	housekeeping
HP	Hewlett-Packard
HRS	High-Rate System
HSIO	high speed input/output
HW	hardware
ICW	input control word
IP	internet protocol
KB	kilobyte
Kbps	kilobits per second
KFTP	file transfer protocol in Kerberos environment
LAN	local area network
LRS	Low-Rate System
MB	megabyte
MDM	multiplexer-demultiplexer
MET	mission elapsed time
mm	millimeter
MPS	multimode portable simulator
MTYP	message type

MVME	Motorola based VME
NASA	National Aeronautics and Space Administration
Nascom	NASA Communications
NEFC	next expected frame count
NRZ-L	non-return to zero-low
NRZ-M	non-return to zero-mark
OCW	output control word
OMD	operations management data
OMDSIM	OMD simulator
PC	personal computer
PCM	pulse code modulation
PDB	project database
PEP	polynomial error protection
PN	pseudorandom noise
PSS	portable spacecraft simulator
R/S	Reed-Solomon
RB	Rate Buffered
RF SOC	Radio Frequency Simulations Operations Center
RTC	relative time command
SC or S/C	Spacecraft
SCC	spacecraft controls computer
SN	Space Network
SOC	Simulations Operations Center
SRC	source
SRS	system requirements specification
SSDM	(CSC) SEAS System Development Methodology
SSR	solid state recorder
StP	Software through Pictures
SW	software

SWCI	Software Configuration Item
S/C	spacecraft
TBD	to be determined
TBS	to be supplied
TC	telecommand
TCP	transmission control protocol
TDRS	Tracking and Data Relay Satellite
TLM	telemetry
TTS	Test and Training Simulator
UDP	user datagram protocol
UIL	User Interface Language
UTC	universal time, coordinated
VC	virtual channel
VCDU	virtual channel data unit
VCID	Virtual Channel ID
VID	Vehicle ID
VME	versa module eurocard
WGS	Wallops ground system
WOTS	Wallops Orbital Tracking Station
WSGT	White Sands Ground Terminal
XTE	X-ray Timing Explorer
XTTS	XTE Test and Training Simulator