AAS 96-015

THE MARS GLOBAL SURVEYOR SPACECRAFT TEST LABORATORY

Keyur Patel

The spacecraft testbed is a crucial part of the spacecraft development and operations phase. Given the aggressive Schedule of the Mars Global Surveyor spacecraft, tbc Spacecraft Test laboratory will play a crucial part during the MGS development phase in support of flight software testing, fault protection testing, sequence testing, and Other areas to meet the November, 1996 launch date. Although it is basal on the Mars observer Verification Test laboratory, the MGS Spacecraft Test Laboratory required a number of hardware and software modifications to accommodate the new spacecraft design, incorporate the lessons learned during flight, and incorporate the findings of the subsequent failure investigations. The result is a modified testbed ready to support spacecraft integration and test and mission operations to return to the red planet. The latest results in the use of the Spacecraft Test Laboratory will be presented.

INTRODUCTION

The Mars observer spacecraft was launched in November 1992 to return to the red planet to Complete the database started by Mariners and Viking spacecraft. On August **21,** 1993, just prior to mars orbit insertion, the Mars Observer (MO) spacecraft was lost, The]0SS Of th C Mars Observer spacecraft on August 21, 1993 eft a void in the martian planetary science database. To recover the loss of MO data and to continue Mars exploration NASA approved he Mars Global Surveyor (MGS) spacecraft.

The MO Verification Test laboratory (the MO testbed) was initially delivered to JPL in June 1992 for operational USC. It was extensively used for sequence validation, flight software and fault protection testing, capability demonstartions, process verification, and anomaly investigation. After the loss of MO the VT]. was in use over six months to

support the MO failure review boards and played a significant role n simulating the different scenarios to determine the probable failures. A number of mollifications to increase fidelity or correct errors, in both hardware and software, were required on the VTL to support this testing.

In July 1994 Lockheed Martin Astronautics (LMA), Denver, was selected as the MGS system contractor. in this capacity LMA is responsible for the spat.ccrfat design, integration, test, and operations. To support this activity the MO VTL was shipped from JPL and installed at LMA in August 1994. The approach taken in adapting the MO VTL to MGS STL was to only modify what is necessary to reflect the MGS design. This resulted in few modifications to the VTL and left the rest of the proven VTL system intact.

MISS1ON DESCRIPTION

The MGS spacecraft is schedu cd to be launched from Cape Canaveral on a Delta II/PAM-D launch vehicle combination in November of 1996. After launch, the Earth-Mars cruise period lasts for 10 months and the spacecraft arrives at Mars in september of 1997 and performs the Mars orbit insertion burn to place itself in 48 hour elliptical orbit around Mars. By March of 1998, aerobraking will place the spacecraft in its final orientation at a 2:00 PM Sun-sync horous orbit at an altitude of 378 Km above Mars. The mapping phase will begin in April 1998 and continue for a fall martian year until April 2000. During the mapping phase the nadir pointed instruments will collect global data on Mars. After the completion of the primary mission the MGS spacecraft will provide relay links for future landed elements.

OVERVIEW

As its predessor the STL is a projec resource test facility which will provide the capability to verify/test the spacecraft hardware and f ight soft ware interaction under diffrent conditions. The primary use of the facility is for:

- Flight Sequence Testing
- Flight Software Testing
- Anomaly Investigation
- Test and Training Exercises

- Process Development and Verification
- Spacecraft Development Testing

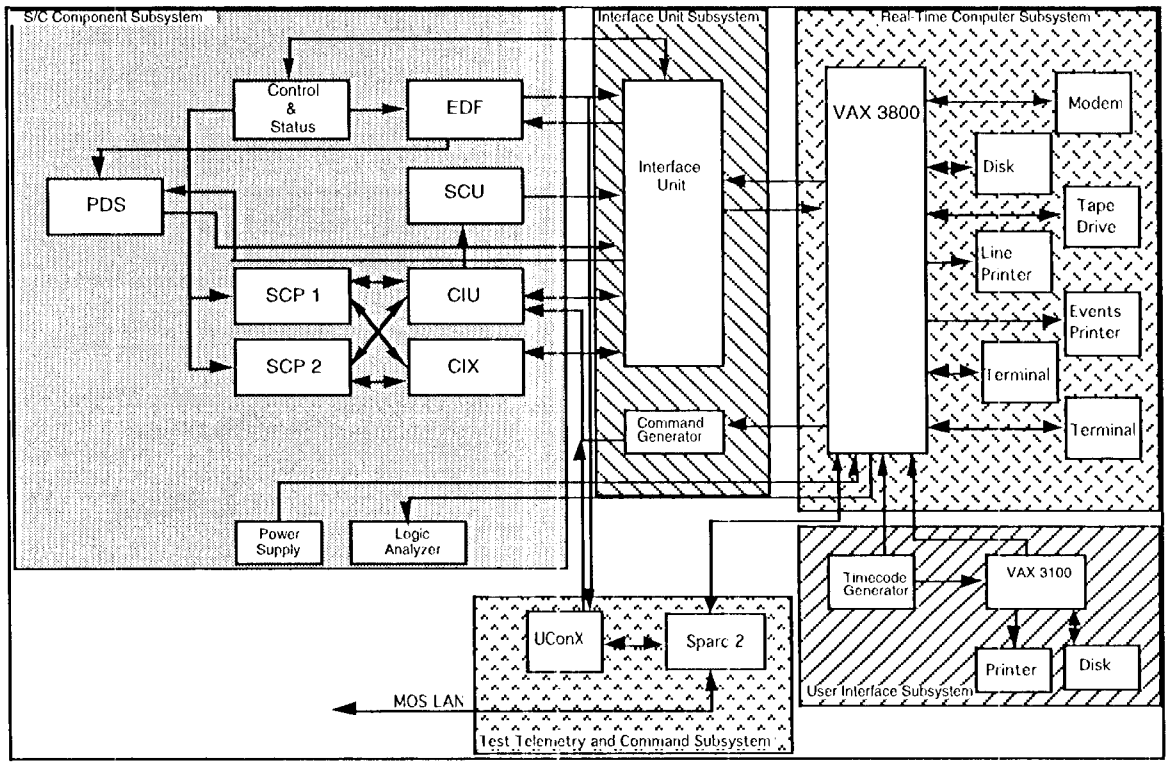

Figure 1 S'1'1, Block Diagram

In the post launch enviornment the STL will be used in support of the first 4 items.

The MO VTL had a number of capabilities that Were extremenly beneficial for simulation execution and simplified simulation operations. All of these are available in the S"1'1.. Some of the more i report ant capablities arc:

- Pause and Continue
- Checkpoint and Restart
- Timejump
- Memory Peck and Poke
- Fault Injection

After the loss of MO and during the subsequent investigations a number of problems were discovered and corrected in the VTL system.

The most significant problem concerned the generation of the 10Hz clock signal by the simulation clock. When certain failures were introduced into a simulation, concerning the CIU clock dividing chain, the resulting effects were inerror due to the VTL suppling an external 10I1z signal to the simulation. This external signal is required for the pause/continue function to work. The VTL was modified to function in two modes. 'l'he first mode will let the VTL function in its normal mode. The second switched tbc VTL during the simulation to the internal 10Hz signal but the user lost the capability to exercise a number of VTL capabilities. 'l'his two mode operation capablity is maintained for the STL. The other problem discovered was the lack of a shadowing algorithm for the sun sensor model. This was never corrected for VTL, but it will be corrected for S"1'1.. The rest of the problems wer minor and were corrected on VTL or will be corrected on ST].,

As the VTL, the S'1'1. (Figure 1.) is comprised of the Spacecraft Component Subsystem (SCSS), the Interface Unit Subsystem (l(JS), the Real-'l'imc Computer Subsystem (l<"l'CS), and the user interface subsystem (lJ]S). The new addition to the S'1'1, is the Test and Telemetry and Command Subsystem ('l'T'ACS).

The SCSS consists of the. non-flight components of the MGS spacecraft Command and Data Handling Subsystem (C&DH). The non-flight S/C elements arc functionally identical to flight product. These elements will be built using the flight drawings and similar parts with exceptions in packaging, testing and the use of commercial grade components. The entire C&DH subsystem is not represented with the hardware boxes, some of it is emulated in software in the RTCS. It also includes a power and control unit, test instrumentation, and the MGS flight software.

The SCSS consists of essential spacecraft components to provide sufficient fidelity for the simulation. The spacecraft hardware represented in the STL arc the Standard Controls Processor (SCP), the Engineering Data Formatter (EDF), the Computer Interface Unit (CIU) and the Computer Interface Unit Extender (CIX), and the Signal Conditioning Unit (SCU). The SCP is the Marconi chip set utilizing the MIL-STD-1750A instruction set with 128K RAM words and 22K of prom for the safemode software. The EDF performs the collection, distribution, and formatting for the spacecraft engineering da a and has 32K of RAM and 22K prom. The CIU and CIX function as the input/output interface to the external devices on the spacecraft. The SC' J consists of the relays required for the

power switching on the spacecarft. All the above boxes in the S"1'1. arc fully redundant except for the SCU. The modifications to these boxes arc described in Reference 1 .

The SCSS also includes the power, control and status box(es) which provide the means to monitor and safely provide power to the non-flight hardware. The test instrumentation provides a means of capturing critical spacecraft signals, mainly the SCP and CIU/CIX traffic, logging thcm for a programmable period, and providing them to the Real-Time Computer Subsystem

The RTCS contains two separate soft Ware components which interface to each other and to the other S'1'1. subsystems. The real-time support software component shall provide for telemetry downl ink, command uplink and interface to the user subsystem for simulation control. The models software component shall contain the S/C models not represented in hardware and the models required to perform c1 o s c cI loop simulation,

The RTCS consists of a MicroVax 3800 with 32 Mbyte of internal memory. It also contains the peripheral equipment for display a n d archival of 1/0 traffic, telemetry, uplink data and event messages. The peripheral equipment includes a 1200 (3x400) Mbyte disk drive memory, a VT340 color graphics terminal, a line printer, an events printer, a digital tape system which can be used for file transfer with JPL, a custom S'1'1. simulation control panel, console, a modem for electronic file transfer to/from JPL, and an Ethernet interface.

The IUS consist of the hardware necessary to interface the non-flight hardware components to the RTCS. It includes custom interface circuitry, simulation control electronics, and a command generation electronics clement, a master clock source (RX()) and a simulation clock which starts and stops with the simulation. The interface electronics will be able to receive spacecraft output data and control signals from the CIU and will be able to return simulated spacecraft input. Thc interface electronics will aJso provide 256 discrete digital and 384 analog simulated telemetry data and control signals to the EDF from the Models Software (MODS) in the Real-Time Computer. in addition, simulation control electronics receives clock and control information from th C spacecraft components anc will be able to receive Engineering Transfer Frame (ETF) telemetry at al S/C rates and transfer ETF's to the Real-Time Computer.

The command generation clement of the IUS is used to send S/C commands and data from the Real-Time computer Subsystem a n cl t r a n s f e r them to the CIU in demodulated DSN message format at al 1 specified MGS spacecraft rates and in all spacecraft modes.

The UIS i s used to initialize tests a n d simulation control, t o perform post-processing on logged simulation data, to display and print information and t o uplink S/C commands. 'l'he UIS workstation computer is a MicroVax 3100, with 16 Mbyte of internal memory, 312 (3x 104) Mbyte disk drive memory, a color monitor, and a page printer.

The TTACS which was initially tested on the MO VTL, after its loss, is used to send spacecraft commands and process downlink telemetry. It is also the same system which will be used during the ATLO phase of t h e MGS spacecraft development. The TTACS consists of a Sun Spare 2 workstation, with 32 Mbyte of internal memory, 2 Gbytes of disk drive memory, a digital tape system, a UConX interface (serial to ethernet conversions), ethernet interface and a color monitor. The TTACS a c a n be considered as a black box which interfaces with the STL to provide uplink and down] ink capabilities. In this regard it will ıot be discussed in this paper.

The three main computer software configuration tems utilized i n the RTCS and the UIS arc the Real-'l'imc Support Software (RTSS), t h c Command Telemetry and User Interface Software (CTUS), and the Models Software (MODS).

Real-Time Support Software

The 1<"1'SS contains 11 software components which interface between t h c host computer and the S"1'1. hardware. The 11 functions arc:

- Simulation Control
- Telemetry Collection a n d Logging
- SCP I/O Traffic Logging a n d Dispatching
- Command Uplink
- Time Executive
- Power Supply Controller
- Message Output Control
- Computer Process Interface (includes nter-compu er communications)

- startup
- Model Profile Logging
- Command Sequence Verification

Figure 2 shows the 1<'1'SS data block diagram and the high level relationships between the RTSS modules.

'l'he Simulation Control (SIMCTL) maintains execution control of the simulation in the ST].. It provides the user the capabilities to pause and continue the simulation with 110 restrictions, to perform a checkpoint and to restart form a previous checkpoint, to perform timejumps, and to interrogate and change SCP memory. SIMCTL processes simulation execution commands, data capture commands, and provides direct access to the SCP via the Interface Unit using special software (called SCP Augmented Software (AUGSW)) executing in the SCPs. The AUGSW is not part of the MGS flight software but resides in the SCPs. The only modification that needs to be made to flight software is to redirect the Level O interrupt to point to AUGSW instead of the flight sortware's default ISR. When flight software is executing the AUGSW is inactive, but when a pause is requested, initiating the nonmaskable level O interrupt, AUGSW becomes active.

Telemetry Collection and logging module inputs and logs telemetry and STL status data. The input data consists of spacecraft EDF telemetry data and STL status information collected. Telemetry Collection and Logging is composed of two components. The first, Telemetry Collection, inputs telemetry and STL data and passes the data to the CTUS. The second, Telemetry Logging, logs EDF telemetry frames and STL status frames to disk.

Command Uplink Processing outputs uplink data to the Command Generator and inputs data from the spacecraft which is used to verify the uplink and the spacecraft reaction to the commands. The input command data is from the CTUS and the JPL supplied Spacecraft Message File (SCMF) The verification of the execution of spacecraft commands is accomplished by checking a J]']. supplied Predicted Effects File (PEF) using downlinked command sequence numbers anti telemetry data, mode] state changes, and SCP 1/0 event data.

The Time Executive module (TEXC) maintains all STL time bases and provides precisely timed activation of software components that have strict timing requirements.

The Power supply Controller module monitors the Power Supply, saving its current data to shared memory for access by the Tel emetry Collection CSC.

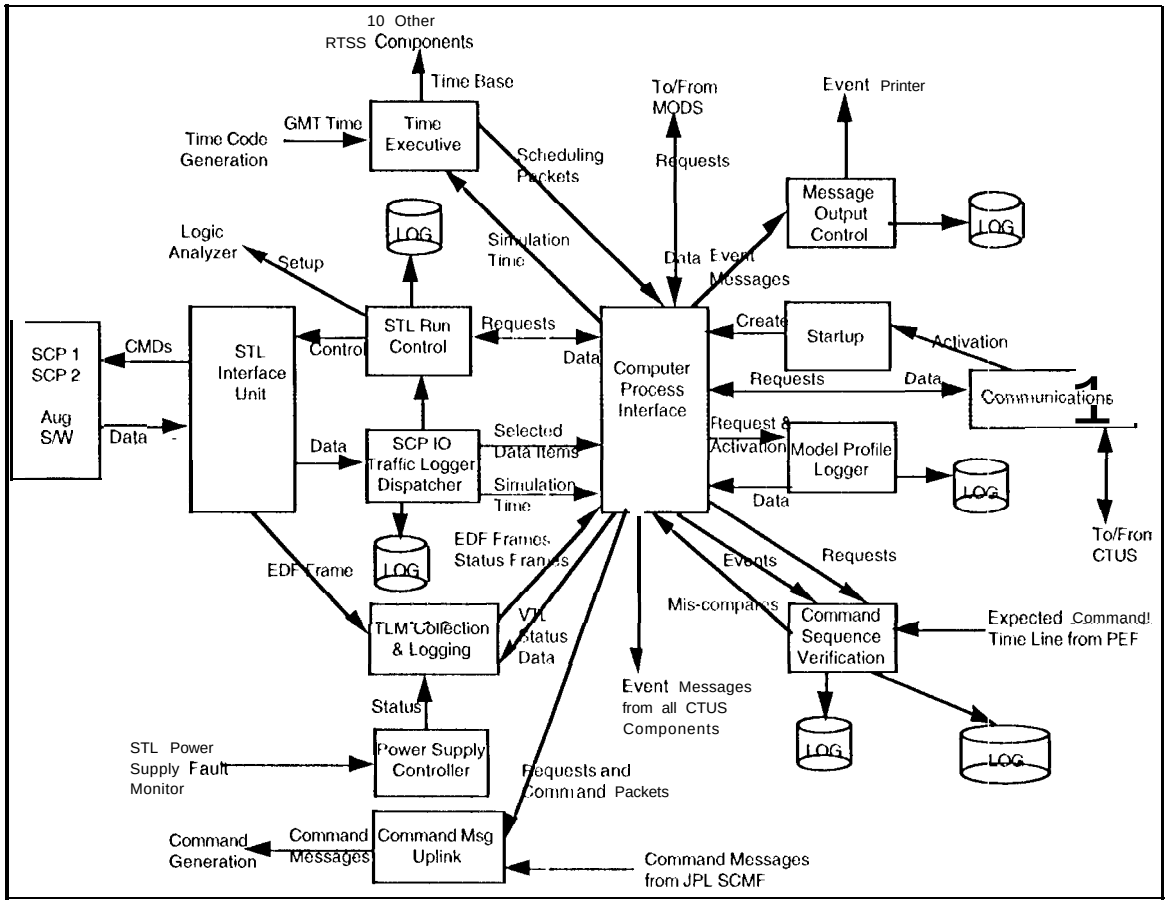

Figure 2 Real-'l'ime Support Software Data Flow Diagram

The Message Output module provides the mechanism for other RTSS components to issue event messages which notify the operator of errors and other significant events that occur during simulation runs. All event messages generated arc logged on a disk file and arc output to the events printer.

The Computer Process Interface module (CPI F) provides the RTSS CSCI with the ability 10 transfer packets between any of its processes or with another module which may be running on the same or different computers. It also provides the interface point for the transfer of packets between the RTSS and the CTUS and MODS residing on the Host

Computer. Simulation file selection requests from the CTUS are processed by this process. The Host Management File is maintained by CPIF, which describes the Host computer's files

The RTSS Startup module (STRT) provides the single point from which software components residing on the Host computer are invoked, including the creation of queues to provide the mechanism by which these components communicate The MODS is also activated by this process. STRT is responsible for initializing databases and starting software components in an orderly fashion, for whose execution must first be a prerequisite to another software component's execution.

The Model Profile logging module (MPLOG) logs MODS model data and transfers model data to the CTUS. MPLOG creates a table of log items and saves this table in the Model Profile Log File. MPLOG logs the data periodically to disk during a simulation run. The logging interval is set by the user and can be as fast as 0.5 seconds to 60 seconds.

Command Telemetry and User Interface

The Command, Telemetry, and User Interface software package, CTUS, contains the necessary components to perform the functions listed below.

- Database Management and Generation
- STL System Startup
- Computer Process Interface (includes inter-computer communications)
- Time Executive
- Displays
- Prints
- STL System Control
- Telemetry Processing
- Spacecraft Memory Management
- Spacecraft Commanding Off-line Support Software
- Command list Generation
- Post-Processing

Figure 3 shows the CTUS data block diagram and the high level relationships between the CTUS modules.

The database management module (DBMG) provides a user friendly environment to facilitate the creation and modification for the database files used by other CTUS. The database software package is Ingress. Databases are maintained for the following:

- Spacecraft Commanding
- Telemetry Processing
- Limit Checking
- CIU/CIX Configuration
- Model Parameters

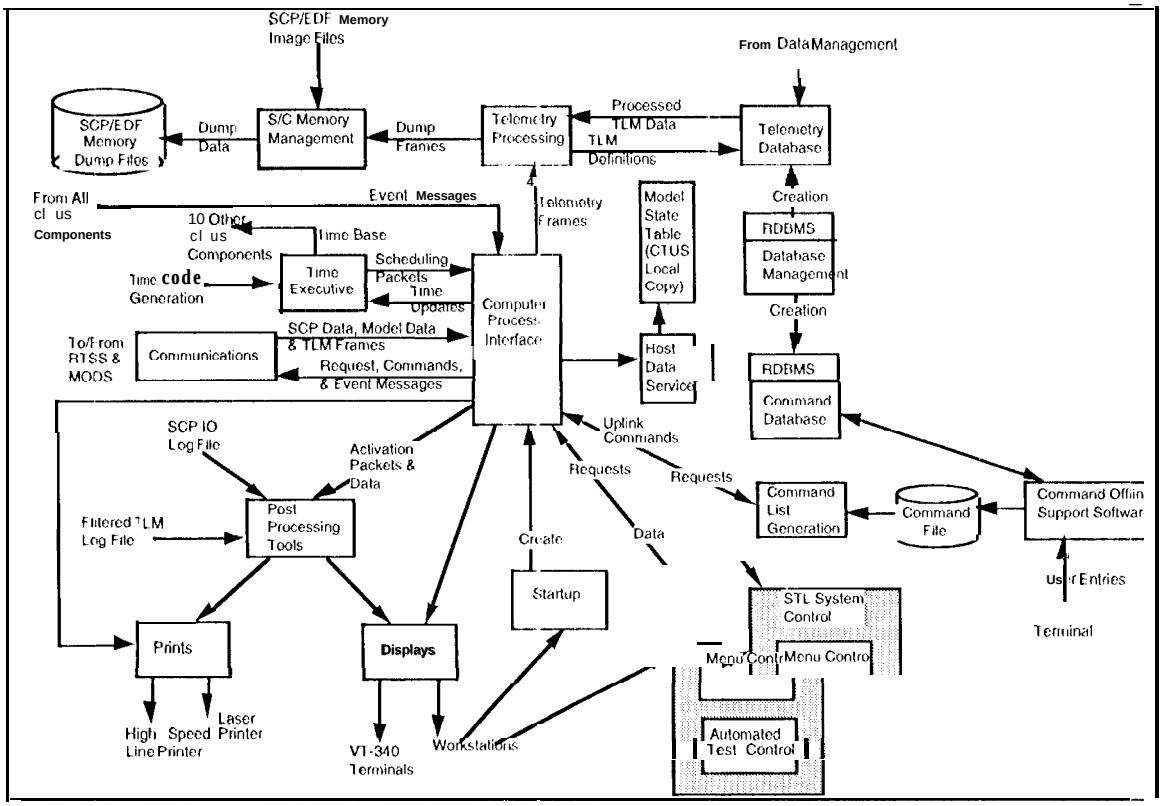

Figure 3 Command Telemetry User Interface Data Flow Diagram

Database Management function provides the CTUS with the capability of storing and retrieving information in the CTUS run-time database. User input for a particular database creation (e.g. Telemetry, Command, etc.) is through a menu environment anti executed off-line at a time when the STL is not performing a simulation or on another workstation.

The CTUS Startup module (Sri'Rrl') provides the single point from which software components residing on the workstation computer arc invoked, including the creation of queues to provide the mechanism by which these Components communicate. STRT is responsible for initializing databases and starting software components in an orderly fashion, for whose execution must first be a prerequisite to another software component's execution. STRT is also responsible for the orderly shutdown of the STL software components executing on the Workstation. An S"1'1. simulation is started by simply typing "GO_VTL" (in tribute to MO) and the simulation is terminated by tying "STOP_VTL".

The Computer Process Interface module (CPIF) provides the CTUS with the ability to transfer packets between any of its processes which may be running on the same or different computers. It also provides the interface point for the transfer of packets between the CTUS to the RTSS and MODS.

The Time Executive module maintains all STL time bases and provides precisely timed activation of software components that have strict timing requirements.

The display module provides for the display of simulation run-time data and post-processing data generated from the models, EDF outputs, and SCP 1/0 activity. The display types consist of the Telemetry Page Display (for VT-340 terminals) and the 1/0 Traffic Display.

The print module provides for the hardcopy output of simulation run-time data and post-processing data generated by the models, EDF outputs, SCP 1/0 activity, and S'1'1. run-time control.

'l'he Automated Test Control function allows automatic, time-ordered execution of some STL functions which arc also menu selectable. Due to the automation of discrete steps, the ACTL provides the ability to repeat the identical simulation with generally the same timing on the STL. The STL run-time simulation is controlled by prepared test schedules consisting of commands and associated time tags for ST]. run-time control, model parameter modification and fault injection, and spacecraft commanding. 'l'his is accomplished through Automated Test Procedures (ATPs). An ATP is a free-format ASCII text file created by the lest operator using the VAX/VMS text editor or any computer prior to

starting the test run. This was the prefered method on MO for sequence testing and flight software regression and baseline testing.

The telemetry processing module (TLMP) configures and initializes the ST1. to receive this telemetry data from the host computer via the CPIF. TLMP checks the received data against user-defined limits, decommutates telemetry minor frames into individual telemetry points and stores [his data in the CTUS database for USC by other CTUS modules. TLMP also logs to disk each telemetry point which exceeds a pre-selected threshold (filter).

The spacecraft memory management module (SMEM) stores the C SCP and EDF memory image and formats the data for later comparisons to command uplink memory loads. SMEM inputs a second file containing information of SCP code and data for masking areas during memory compare. SMEM gathers telemetry dump data, SCP or EDF, passed from the TLMP module and stores the data into files on disk. The collection is started when a memory dun) p is iniatiated from the SCP/EDF and terminates When the dump end address has been reached 01 the dump is terminated by another ground command.

The SpaCCCraft Commanding function provides the capability for building, validating, dispatching and uplinking both individual and block-mode commands including memory loads, as designated via In c n u and automatic test procedures, to the spacecraft (eq ui pment). The 11 ser creates ASCII command lists using the VAX/VMS text editor or imports a file in the proper format. commands and data words in the file are specified via database-dcfiJ]cd mnemonics. Command execuiion times and command verification request flags are also specified in the file. The Spacecraft Commanding function, in order to accomplish the creation and transmission control of command sequences to the spacecraft, contains software components which execute outside the S'1'1. runtime environment as Well as components which execute on different computers and different software Components, CTUS and RTSS. The runtime CTUS components form binary command messages from command list text files, whereas the RTSS component is responsible for sending these command bit sequences to the command generator at the specified times for uplink to the spacecraft. The RTSS also checks telemetry data to obtain command verification status during command sequences. The offline CTUS components allows the user to build SCP script command files for eventual uplink.

Models

The models software (MODS) provides the closed loop simulation response for the spacecraft SCP commands to meet the simulation needs. The MODS CSCI executes within the STL RTCS. MODS provides simulation support for all phases and modes of the MGS mission, providing simplified models where physical effects arc not detected or responded to by the flight software. The interfaces between the models arc shown in Figure 3. The functions performed by the MODS CSC arc:

- Simulation of the spacecraft hardware within the AACS, C&DH, Payload, Power and Propulsion subsystems
- Simulation of the solar and planetary positions
- Simulation of the space.craft dynamics and power
- Generation of simulated telemetry data
- Logging of requested events and data
- Execution control
- Fault injection

MODS willsimulate the spacecraft subsystems and their individual hardware devices including the processing of the SCP device commands, generation of its telemetry points, creation of the SCP inputs, and determining any changes in spacecraft state. The device models provided in the S'1'1. are:

- Reaction Wheel Assembly (RWA)
- Inertial Measurement Unit (IMU)
- Celestial Sensor Assembly
- Mars horizon Sensor Assembly
- Four Pi Steradian Sun Sensor Assembly
- Solar Array Gimbal Drive Electronics
- High Gain Antenna Gimbal Drive Electronics
- Propulsion System
- Thermal system
- Power system
- Data Handling
- Launch System

The non-device models in the S'1'1. are:

- Enviornmental Model

- Dynamic System Model
- Ephemeris Model Subsystem

Figure 4 shows the MODS data block diagram and the high level relationships between the MODS modules.

'l'he RWA model simulates the RWA hardware for the X, Y, Z, and Skew RWAs. The simulation includes modeling the receipt of commands from the SCP; generation of the wheel torque inputs to the Dynamics Model; output to the SCP of wheel speeds, directions, power on/off and power-limitc(i status; simulate. d EDF telemetry values; and the introduction of simulated hardware failures.

The IMU model simulates the gyroscopes, accelerometers, electronics, and CSA interface of the MGS inertial measurement unit. The IMU model takes the angular rates and acceleration of the spacecraft , translates that information into corresponding simulated IMU sensor data, and outputs this data to the CIU/SCP via the IU. The IMU mode] also formats the star transit data from the Celestial Sensor Assembly (CSA) model for output to the SCPs

The CSA model simulates the detection of starlight entering thc FOV of the slits in the Celestial Sensor Assembly. For each simulated star transit, the CSA model outputs the star transit time and slit number, This model simulates both the operation of the hardware and the actual star image crossings. This model was modified on MO to include a full star catalog capability,

The MHSA model simulates the operation of the four quadrants and detector assemblies of the MHSA according to spacecraft attitude, position, ephemeris, and the position of Mars to provide information about the detection of the Mars horizon for primary attitude control reference during the Mars mapping phase. This model also simulates the electronics required to provide digital attitude error data to the SCP.

The SSA model simulates the operation of the two-axis sun sensor assemblies to determine the measurement of the sun vector with respect to the spacecraft. The SSA model provides a measurement Of the Sun vector in body coordinates. The model simulates the redundant assemblies on the spacecraft by operating in one of two modes. It can accept a command from the ground (via the SCP) to select the detector or it can operate in automatic mode. The model simulates the

generation of an analog ATA (Automatic Threshold Adjust) signal proportional to the intensity of the incident sunlight and sends cl at a from the detector with the highest ATA. For the computer detector selection mode, the model simulates the selection of a sensor.

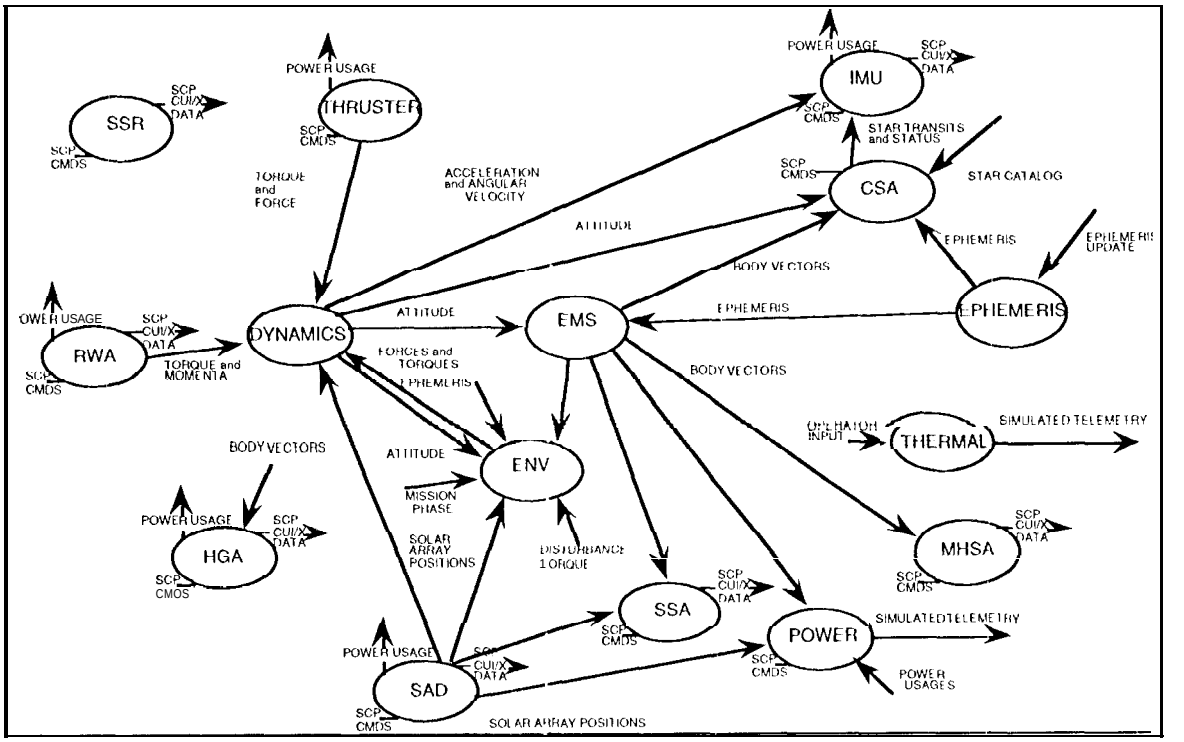

Figure 4 Models Data Flow Diagram

The SAD 111odel simulates the current and status from the Solar Array and provides a measurement of az im uth/elevation speed, direction, and position. This model functions also includes the receipt, processing and validation of SCP commands that affect the gimbal drive electronics and rotation of the solar arrays. The motor drive m odel 1 ed by the SAD subsystem is capable of operating in two command modes: Rate and Step. The operating mode is selected by the input comma 11 d word. The SAD model also computes the position of each solar array with respect to the spacecaft

The HGA model simulates the current and status of the High Gain Antenna during slewing and provides a measurement of azimuth/elevation speed, direction, and position. This model functions also includes the receipt, processing and validation of SCP commands that affect the gimbal drive electronics and position of the }] GA. The

motor drive modelled by the HGA model subsystem is capable of operating in two command modes: Rate and Step. The operating m ode is selected by the input command word. It also calculates the HGA position with respect to the spacecraft and the antenna view of t h c Earth.

The thruster mode] simulates the operation of the propulsion system. This includes the receiving and processing of SCP commands for the pym valves, latch valves, thrusters, and the main engine. It will produce the associated telemetry values, and allow for the CTUS t o simulate hardware failures. In addit on, this model also calculates t h c forces and torques resulting from the operation of the bi-propellant a n d mono-

The thermal model provides u p and down temperature ramps for each of the Differential Thermal Controller (DTC) heater-controlled elements on the spacecraft monitored by the flight software. The ramping is user-control]ccl so that flight software redundancy management of the DTC heaters can be tested. Additionally, this model provides up and down temperature ramps for the NTO and MM]] tanks (whose heaters are actively controlled by the flight software). All temperature points within the simulated bus telemetry may b c controlled by the user through telemetry overrides and take precedence over temperatures calculated by the thermal control model.

The power model establishes the spacecraft power balance based on commanded loads and system configuration. It also accepts a n d validates the power commands from the SCP and outputs data a n d telemetry, simulating the regulation strategy of the battery and s h u n t activity if excess power is available. This model determines the operating point of the solar array, and calculates the total power 1 oad being used by spacecraft equipment SUCh as the DTRs, RWAs, GDE, hrusters, the anticipated load from the heaters, as w C]] as payload nstrument on/off commands.

in the spacecraft, the XSU provides switched digital signal a nd iming interfaces in the C&DH. The switched interfaces permit connecting the outputs of the PDS, EDE, and 7'0S to three Electronic Un its and four Tape Units, and to two transponders and also permit the connecting of playback outputs on the recorders to the transponders. The XSU mode] will processes the commands that control the state of the XSU.

The SSR model simulates the record, position, and playback commands. Power consumption is also modeled.

The MOT model receives, processes, and validates the SCP commands that control the MOTs, 'l'W'l'A, and RF power amplifiers. 1 t also maintains the status of the commanded equipment.

The PDS model receives, processes, and validates the SCP commands that control the on/off status of the PDS and the scientific instruments. This model is turned off when tests arc being conducted with the PDS hardwrac integrated.

The launch system model consists of interfaces with the ground support equipment (GSE), the Titan 111, Transfer Orbit Stage ('1'0S) and the MO adapter, Within the launch subsystem, modelling will be limited to fixed output parameters as a function of commanded state.

The enviornmental torque model computes a Fourier S-term series approximating the net environmental torque. The coefficients for the series arc chosen to approximate the total of the gravity gradient torque, solar radiation pressure torque, and aerodynamic drag torques. Two sets of coefficients arc available: one set for mapping and another set for cruise. The angle used for evaluating the Fourier series in cruise mode is the angle from the X axis to the sun about the -Y axis. in mapping mode the series is evaluated using the angle from periapsis. The series evaluation may be overridden by an user-sJ>ccificd environmental torque (in the body frame) for a specified number of execution cycles.

The dynamics system models the dynamical state of the spacecraft. It takes into account all external torques and forces, inert ias, mass, RWA torques and momentum, and the system momentum. It utilizes a fourth order Runge Kutta integration of the state equation to compute the dynamic state of the spacecraft.

'l'he spacecraft configuration model maintains the spacecraft structural and mass model as it varies during the mission. This model determines the spacecraft mass and propellant mass used as well as the location of the spacecraft center of mass and the inertia tensor. It computes the rotation into the new principal axis system from the spacecraft body system of coordinates.

The spacecraft state mode] defines the dynamical state of the spacecraft in inertial space. in particular, it determines the angular velocity, momemtum, position vector, and the orientation vector of the spacecraft in its orbit in inertial space. This mode] is also an orbit propagator that will include external forces affecting the orbit of the spacecraft. It provides the dynamic inputs into the lMU which reflect the motion and acceleration of the spacecraft in its orbit and the simulated spacecraft orientation in inertial space for the attitude determination sensor models as well as the environmental torque models.

The ephemeris mode] subsystem defines the position of Mars, Earth and the Sun in inertial coordinates. These unit vectors arc then transformed into body-fixed coordinates using the inertia] to body transformation based on the derived attitude of the spacecraft. Three coefficient matrices arc used to calculate the Mars, Earth and Sun vector. These matrices are based on approximations of the motion of Mars, Earth and sun, and they provide accurate results for a short period of time. A new matrix containing ephemeris data must be processed by this mode] subsystem whenever new ephemeris data is commanded to the SCP. No SCP commands are received by this mode] subsystem. No data or telemetry is output by this mode] subsystem. The ephemeris model subsystem outputs data to other models for usc in their calculations.

The ENV model subsystem simulates the external torques and forces on the spacecraft that are a result of solar pressure, gravity gradient and aerodynamic drag. A truncated Fourier Series is used to predict these disturbances. No SCP commands are received by this 111odel subsystem. No telemetry is output by this model subsystem. The ENV model sybsytem outputs data to other models for use in their calculations.

NTERFA CE UNIT

The Interface Unit Subsystem (IUS) performs the 1/0 buffering, reordering and timetagging of information transfer from the Host Computer to the SCSS equipment and from the SCSS to the Host. 'l'he IU also contains a simulated RXO which provides the clock source for both the SCSS and system simulation Clock.

The IUS is a system of racks in a number of cabinets which contain the equipment that interfaces the SCSS to the Host Computer. The IUS is composed of VMEbus (IEEE 1014) systems each of which is controlled by a vendor supplied VME industrial controller (PME 68-14). TWO systems arc contained in the Simulation Clock Chassis and one is contained in each of the three interface chassis, The purpose of the PME 68-14 is to control the transfer of data between the RTCS and the SCSS. The other circuit cards that comprise the five Interface Unit VMEbus systems arc a combination of vendor supplied and custom designed cards. All interface to the VMEbus according to IEEE 1014 interface specification.

The data provided by the RTCS for input to the SCPs has to be transferred 10 the **IUs.** Since this data must be in place in the appropriate IUS buffer when the input request is received, the RTCS will generate this data faster than real-time. This is accomplished by means of an advanced 10 Hz, interrupt which the IUS sends to the mode els software in the RTCS. The advanced 10Hz interrupt Sent from the IU to the Host Computer is used by MODS to start its process to generate the data required to provide to the SCSS for the next interrupt cycle.

The IUS chassis provides simulated discrete and analog telemetry to the EDF in the SCSS. All telemetry points arc updated by a single message from RTCS. The message is received via a DMA transfer from a Host Computer and is temporarily stored in the chassis controller card message buffer. The elements of the message arc distributed to their individual destinations by the chassis controller. The discrete points arc transferred 16 at a time in 16 transfers. The 384 analog values arc transferred one at a time to the. 384 analog inputs of the EDF. Each value is maintained in its buffer until it is updated.

The IUS also transfers 1/0 event messages and CIU/CIX discrete buffer output from the SCSS to the RTCS. The occurrence of an I/O **CVCH** t is sensed by direct connections to the SCP 1/0 buses. The bus information is buffered and multiplexed by circuits within the NFSC Cabinet and then transferred via high speed "1''1'1. line drivers and matching receivers on the Address Decoder Circuit within Interface Chassis 2 in the Interface Unit. All connections to the SCP 1/0 busses arc buffered with high impedance line receivers to prevent loading. The CIU/CIX busses of each SCP arc multiplexed to reduce the cabling to IU

The Simulation Clock Chassis (SIMCLK), in the IUS, contains the interface for the Downlink Telemetry from the EDF, the serial interfaces

to and from the EDF and the CIU/CIX, discrete outputs to the CIU PIBs, and the Simulation Clock circuitry. The simulated RXO provides a 5.12 Mhz RXO signal to both the CIU and the SIMCLK. The CIU takes the signal and divides it down to provide the SCSS components with the appropriate signals. Meanwhile the SIMCLK takes the same signal and divides it clown to provide the clock sources needed in the STL.

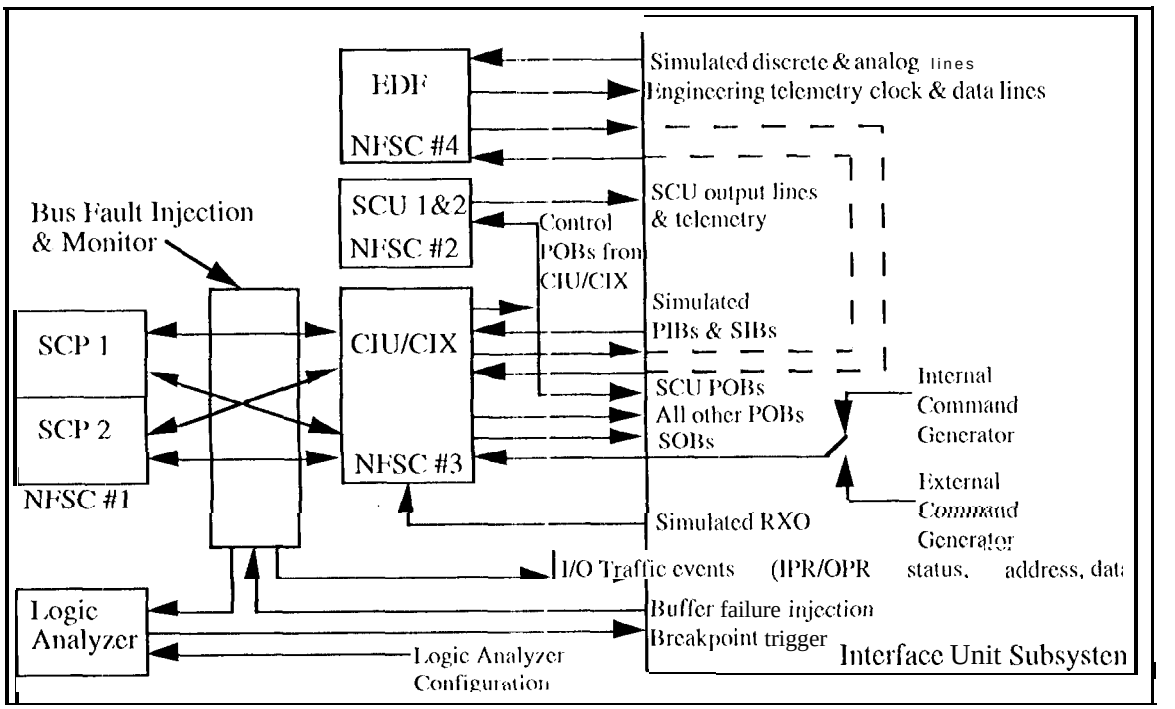

Figure 5 Interface Unit Subsystem High Level Diagram

The command generator (CG), in the IUS, accepts spacecraft command data from the host computer and converts it to a uplinkable message (DSN format). It generates and appends the checksum as well as the acquisition, head, tail and idle sequences. The CG supports all uplink bit rates required for MGS including the 5000 bps GSE rate.

CONCLUSION

The MO VTL is on its way to be successfully converted to the MGS STL system. With the significant inhertance from MO the MGS Project bad a working testbed early in the development process. This provided early support for flight software testing to minimize the impact to the

spacecraft integration schedule. As the Project enters its different phases in the development and operations process the STL will bc ready,

ACKNOWLEDGMENTS

The work described in this paper has been performed by the Jet l'repulsion laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Adminiatration.

1 would also like to thank Alan Cheuvornt at Lockheed M art in Astronautics, Denver in providing support in preparation of this paper.

REFERENCES

1. R. 1.. Bunker, The MGS Avionics System Architecture: Exploring t h c Limits of Inheritance,
2. The Mars Global Surveyor Mission, JPL 400-551
3. MGS Spacecraft Test Laboratory System Requirements Specification, JPL D-1 2346
4. MGS Spacecraft Test Laboratory Design Specification, JPL D-1 3002