1111 66 N95- 17581

31986

p. 7

TEST/SCORE/REPORT: SIMULATION TECHNIQUES FOR AUTOMATING THE TEST PROCESS

Barbara H. Hageman
Integral Systems, Inc.

Clayton B. Sigman
National Aeronautics and Space Administration
Goddard Space Flight Center

John T. Koslosky
National Aeronautics and Space Administration
Goddard Space Flight Center

ABSTRACT

A Test/Score/Report capability is currently being developed for the Transportable Payload Operations Control Center (TPOCC) Advanced Spacecraft Simulator (TASS) system which will automate testing of the Goddard Space Flight Center (GSFC) Payload Operations Control Center (POCC) and Mission Operations Center (MOC) software in three areas: telemetry decommutation, spacecraft command processing, and spacecraft memory load and dump processing. Automated computer control of the acceptance test process is one of the primary goals of a test team. With the proper simulation tools and user interface, the task of acceptance testing, regression testing, and repeatability of specific test procedures of a ground data system can be a simpler task. Ideally, the goal for complete automation would be to plug the operational deliverable into the simulator, press the start button, execute the test procedure, accumulate and analyze the data, score the results, and report the results to the test team along with a go/no go recommendation to the test team. In practice, this may not be possible because of inadequate test tools, pressures of schedule, limited resources, etc. Most tests are accomplished using a certain degree of automation and test procedures that are labor intensive. This paper discusses some simulation techniques that can improve the automation of the test process.

The TASS system ***test***s the POCC/MOC software and provides a ***score*** based on the test results. The TASS system displays statistics on the success of the POCC/MOC system processing in each of the three areas as well as event messages pertaining to the Test/Score/Report processing. The TASS system also provides formatted ***report***s documenting each step performed during the tests and the results of each step. A prototype of the Test/Score/Report capability is available and currently being used to test some POCC/MOC software deliveries. When this capability is fully operational it should greatly reduce the time necessary to test a POCC/MOC software delivery, as well as improve the quality of the test process.

1. INTRODUCTION

1.1 TASS Background

The Transportable Payload Operations Control Center (TPOCC) Advanced Spacecraft Simulator (TASS) system has been designed to support the development, test, and operational aspects of Payload Operations Control Center (POCC) and Mission Operations Center (MOC) software deliverables. TASS is designed to test the majority of POCC/MOC low-level requirements. The TASS system simulates spacecraft telemetry and command

PRECEDING PAGE BLANK NOT FILMED

PAGE 1102 INTENTIONALLY BLANK

functions. TASS takes advantage of the TPOCC architecture by using the backup POCC/MOC system configuration hardware for the simulator, or TASS can be separately hosted on a streamlined version of the POCC/MOC. This eliminates the need to schedule hardware or Nascom lines during various test configurations. In essence, the user has a simulator on call at all times.

TASS has the capability to simulate the Nascom link protocols required to support satellites and generate simulated spacecraft telemetry streams using the POCC's/MOC's operational data base (ODB). TASS validates spacecraft commands and alters the real-time telemetry stream in response to those commands. The user can alter the telemetry stream either by data base mnemonic or by specifying individual bits in the telemetry frame or packet. Similar telemetry display pages at both the simulator workstation and the POCC/MOC workstation help identify telemetry processing irregularities. As part of the system design, software hooks are available so more complexity can be added by providing various dynamic models for the telemetry generating function.

In the POCC/MOC test environment, the TASS system provides a means for saving and restoring predefined test scenarios and results, telemetry stream contents, and data structures to allow the user to accurately repeat specific tests, retest with known data, or continue testing from a given point in the test scenario. These features allow the user to perform regression tests on new software deliverables in the shortest possible time.

TASS records all received Nascom blocks and all received spacecraft commands in history files that can be viewed for detailed analysis through the use of an offline utility program. All system events, errors, operator input, procedure input recorded in the event log; and spacecraft memory images that are saved can be viewed by using the offline utility programs. After completing the test, the user generates test reports using the report generation subsystem. These reports can later be used to evaluate the test results during the analysis process.

Unique implementations of spacecraft memory load and dump capabilities are provided as well as an NCC communications protocol when TDRSS support is required.

1.2 TASS System Design

A typical POCC/MOC system string is used to host the TASS software. The hardware configuration to support TASS consists of two computers connected by Ethernet and associated peripherals as shown in Figure 1. These computers are a real-time front-end computer or processor (FEP) in a Versa Module European (VME) bus enclosure and a general-purpose computer or workstation. The real-time FEP is used to process spacecraft commands and to build and transmit telemetry streams. The Hewlett-Packard HP 9000-715 workstation allows the user to configure, control, and monitor the FEP from one or more user terminals.

TASS makes extensive use of the same TPOCC reusable software that the POCC/MOC developers use, mainly the user interface (display and TPOCC Systems Test and Operations Language (TSTOL)) and the Nascom interface. The display system is based on X Windows and fully adheres to the industry-standard OSF/Motif principles. TSTOL is the user script language which is used to control the TPOCC application system (either the POCC/MOC system or the TASS system). TSTOL is also used to develop operational scenarios and test procedures. Presently, TPOCC reusable software comprises approximately 78% of the TASS system. Another 16% of TASS is reusable from mission to mission, such that only about 6% of TASS needs to be newly developed with each added POCC/MOC mission.

1.3 Control Center Configuration

Because of the methodology chosen for the overall ground system design, no special equipment or system configuration is required for TASS. TASS uses the POCC/MOC backup system string and communicates with the primary POCC/MOC system string thru the local TPOCC switch. In a test configuration, the TASS input/output data flow at the switch interface looks like the Nascom interface to the primary POCC/MOC system string. This

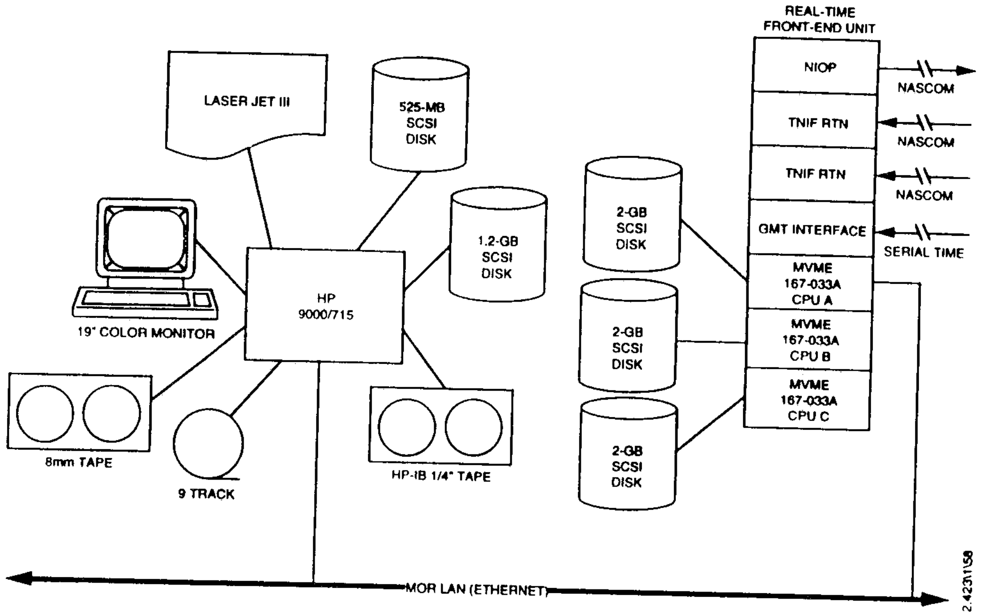


Figure 1. TASS Hardware Configuration

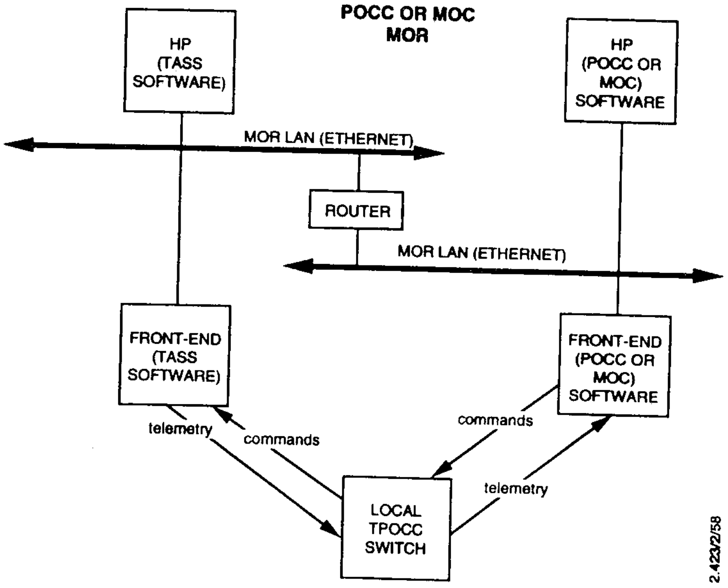


Figure 2. Control Center Configuration

architecture is shown in Figure 2.

The TASS system accepts spacecraft commands from the POCC/MOC and transmits telemetry to the POCC/MOC via Nascom connections on both FEPs. The workstations show displays generated by the TASS and POCC/MOC system.

2. GROUND DATA SYSTEM TESTING

2.1 Software Delivery Test Process

GSFC Control Center Systems Branch is responsible for testing the software deliverables for the POCC/MOC systems in the TPOCC environment. TPOCC based POCCs/ MOCs support the WIND, POLAR, SAMPEX, FAST, SWAS, SOHO, XTE, TRMM, and ACE missions. Testing of these POCC/MOC systems consists of unit testing, integration testing, and finally acceptance testing.

Unit testing is performed throughout the software implementation phase by the POCC/ MOC developers. Unit testing is completed prior to delivery of the unit for system integration.

Integration testing is performed before delivery to the test team. This testing verifies integration of TPOCC generic software and POCC/MOC unit software into the POCC/ MOC system and is performed by an integration manager who is supported by the development team.

Acceptance testing is performed by the test team before delivery to GSFC according to a comprehensive test plan and procedures. This testing verifies the functional and performance requirements and is completed prior to delivery of the system to operations/user community.

2.2 Automating the Test Process

To achieve the goal of automating the test process, several test methodologies have been prototyped. The most promising concept is Test/Score/Report. TASS and POCC/MOC simularities in system architecture, user interface, script language, and project data base files are some elements that support this system concept approach to automate the test process.

3. TEST/SCORE/REPORT

3.1 Automated Testing in Three Areas

The Test/Score/Report capability currently being developed will automate testing of the POCC/MOC software in three areas: telemetry decommutation, spacecraft command processing, and spacecraft memory load and dump processing. TASS takes advantage of the ground system attributes in designing the Test/Score/Report capabilities. Both the TASS and the POCC/MOC systems are using some of the same reusable building blocks of TPOCC software and running on the same hardware architecture. By using this approach, TASS can easily add features which enhance the automated test process.

Figure 3 shows the data and control flows between the two systems. The TASS system simply establishes a socket connection with the POCC/MOC system in order to make requests for data and to receive the data. This connection is transparent to the POCC/MOC system and requires no new software be written on the POCC/MOC side. The TASS system also reads the POCC's/MOC's system variable dump file and the ground image file which both reside on the workstation's disk. The system variable dump file contains all of the telemetry parameters located in the operational data base as well as counters and status information. This file is needed for initialization purposes before requests for data can be made. The ground image file is used to validate spacecraft memory load and dump processing.

Telemetry decommutation is tested in two ways. The first way is by comparing the values of telemetry parameters decommutated by the POCC/MOC against the telemetry parameters commutated by TASS. Ideally, the decommutated values should match the commutated values. The second way is by comparing the limit specifications previously set with the status words of decommutated telemetry parameters. Every telemetry parameter located in the operational data base is automat-

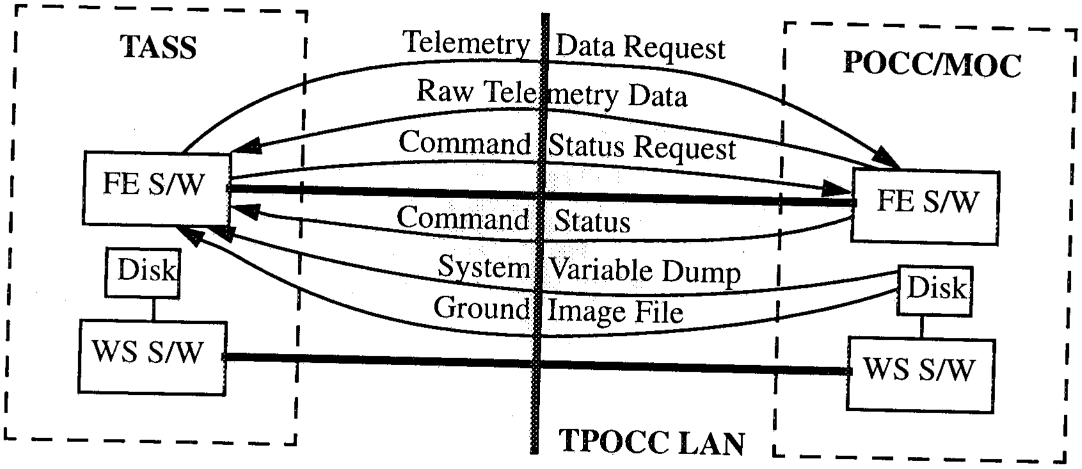


Figure 3. Data/Control Flow Between TASS and the POCC/MOC

ically checked. Each discrepancy is displayed as an event message which gives the value of the decommutated telemetry parameter and the value of the commutated telemetry parameter. Summary event messages for the two telemetry decommutation tests state the total number of telemetry parameters which decommutated correctly, the total number of telemetry parameters checked, and the percentage of which were correctly decommutated.

For spacecraft commanding, the test process validates the various fields in the Nascom block header for all spacecraft command blocks received by TASS and validates the individual spacecraft commands received in valid command blocks. TASS also checks whether the POCC/MOC verified the commands after they were executed by TASS. This is accomplished by making data requests to the POCC/MOC for values of command status parameters and counters. Summary event messages are displayed which give the number of valid command blocks, the number of valid commands, the number of commands verified by the POCC/MOC, and the percentages for each of the above tests.

The spacecraft memory load and dump processing is tested by comparing the spacecraft image maintained by TASS against the ground image file maintained by the POCC/MOC. This test is performed after memory load data is sent to TASS from the POCC/MOC via a spacecraft command and after TASS transmits a memory dump to the POCC/MOC via the telemetry stream. Ideally, the memory values maintained by TASS should match values in the POCC/MOC ground image file. A summary event message informs the user of the number of bytes that miscompared, the total number of bytes in the spacecraft image, and the percentage of bytes which had the same value.

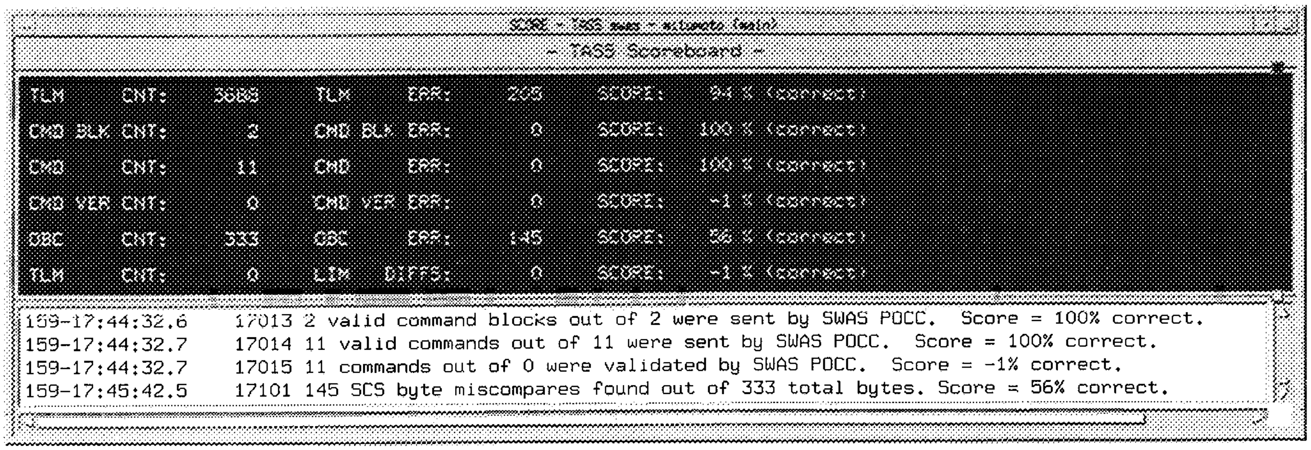
3.2 Scoring the Ground Data System

The initial scoring method will be in terms of percentages and raw counts. More experience in testing and interpreting the test results will be required to develop a better scoring methodology.

3.3 Reporting the Test Results

The TASS system displays statistics on the success of the POCC/MOC in each of the three areas as well as event messages pertaining to the Test/Score/Report processing. The Test/Score/Report display page shown in Figure 4 is broken up into two sections. The top section gives the summary counts and scores for the various tests. The bottom section is a scrolling region which displays all event messages generated during Test/Score/Report processing.

The TASS system also provides formatted reports documenting each step performed during the tests and the results of each step. The user can issue reports for each type of test (i.e., telemetry, limits, command, memory loads/dumps). Report options include showing all



```
SCORE - TASS swas - [illegible] (main)
- TASS Scoreboard -
TLM     CNT:     3688    TLM      ERR:     205     SCORE:     94 % (correct)
CMD BLK CNT:        2    CMD BLK  ERR:       0     SCORE:    100 % (correct)
CMD     CNT:       11    CMD      ERR:       0     SCORE:    100 % (correct)
CMD VER CNT:        0    CMD VER  ERR:       0     SCORE:     -1 % (correct)
OBC     CNT:      333    OBC      ERR:     145     SCORE:     56 % (correct)
TLM     CNT:        0    LIM    DIFFS:       0     SCORE:     -1 % (correct)
159-17:44:32.6     17013 2 valid command blocks out of 2 were sent by SWAS POCC.  Score = 100% correct.
159-17:44:32.7     17014 11 valid commands out of 11 were sent by SWAS POCC.  Score = 100% correct.
159-17:44:32.7     17015 11 commands out of 0 were validated by SWAS POCC.  Score = -1% correct.
159-17:45:42.5     17101 145 SCS byte miscompares found out of 333 total bytes. Score = 56% correct.
```

Figure 4. Test/Score/Report Display Page

parameters tested or just those parameters which miscompared or were in error. For telemetry tests, the report shows the telemetry mnemonic, the commutated value, and the decommutated value. For limit tests, the report shows the telemetry mnemonic, the commutated value, the limit specification and the status word of the decommutated parameter, and the decommutated value. For command tests, the report shows numbers of: Nascom block header errors, command errors, and commands verified. Finally, for memory load/dump tests, the report shows the memory address, TASS's spacecraft image value, and the POCC's/MOC's ground image value.

4. PRESENT STATUS

A prototype of the Test/Score/Report capability is available and currently being used to test the WIND, POLAR, SAMPEX, FAST, SWAS, and SOHO POCC software deliveries and the XTE MOC software delivery. This prototype includes the "test" and "score" features described in this paper. The "report" features (other then the Test/Score/Report display page) are currently being developed by the TASS development team and are planned to be released by early next year. The TASS development team is actively working with the POCC/MOC test team and software developers to obtain feedback on the Test/Score/Report prototype.

5. FUTURE DIRECTIONS

Basically we have completed our prototyping stage of this project. We have been successful in implementing this methodology in several projects as mentioned in the previous section. Future objectives are to automate more of the test process by: 1) including additional subsystem testing such as attitude, events, packet extraction, history, NCC, and database testing; 2) improving the scoring methodology; and 3) providing more functionality and options for the report process.

REFERENCES

NASA GSFC, (June 1994). *Transportable Payload Operations Control Center (TPOCC) Advanced Spacecraft Simulator (TASS) System Requirements Document (Revision 4).*

NASA GSFC, (May 1994). *Transportable Payload Operations Control Center (TPOCC) Advanced Spacecraft Simulator (TASS) System User's Guide for Release 7.*

NASA GSFC, (May 1994). *Transportable Payload Operations Control Center (TPOCC) Advanced Spacecraft Simulator (TASS) Detailed Design Specification for Release 7.*

NASA GSFC, (February 1994). *Transportable Payload Operations Control Center (TPOCC) Detailed Design Specification for Release 10.*

Acknowledgements -- We wish to extend special thanks to the following personnel for ideas and review of the report: Carroll Dudley (NASA GSFC), Luan Luu (Integral Systems, Inc.), Nancy McCluer (Integral Systems, Inc.), and Darlene Riddle (Integral Systems, Inc.).