N95- 17568

111153

31973

p. 8

INTEGRATION OF A SATELLITE GROUND SUPPORT SYSTEM BASED ON ANALYSIS OF THE SATELLITE GROUND SUPPORT DOMAIN

R. D. Pendley, E. J. Scheidker, D. S. Levitt, C. R. Myers, and R. D. Werking

Computer Sciences Corporation
10110 Aerospace Road
Lanham-Seabrook, Maryland 20706

This analysis defines a complete set of ground support functions based on those practiced in real space flight operations during the on-orbit phase of a mission. These functions are mapped against ground support functions currently in use by NASA and DoD. Software components to provide these functions can be hosted on RISC-based work stations and integrated to provide a modular, integrated ground support system. Such modular systems can be configured to provide as much ground support functionality as desired. This approach to ground systems has been widely proposed and prototyped both by government institutions and commercial vendors. The combined set of ground support functions we describe can be used as a standard to evaluate candidate ground systems. This approach has also been used to develop a prototype of a modular, loosely-integrated ground support system, which is discussed briefly. A crucial benefit to a potential user is that all the components are flight-qualified, thus giving high confidence in their accuracy and reliability.

Introduction

The satellite ground support domain comprises all ground-based (as opposed to onboard) activities needed to operate an orbiting spacecraft, including the bus and payload. It does not include such activities as, for example, instrument data reduction from a scientific satellite, image production from a weather satellite, or message traffic management from a communications satellite; although the ground support domain does cover capturing and making available the data required by such end-user processes. This domain also includes the integration of payload plans and commands into the overall plan for mission support. The activities supported by functions in this domain also differ during the prelaunch, launch, early mission, on-orbit, and end-of-life phases of a mission. In this paper we undertake to define a complete set of spacecraft support functions that span the satellite ground support domain during on-orbit operations for one or more spacecraft.

The principal motivation for this analysis is the belief that satellite ground control systems, traditionally implemented on central processor systems based on mainframe or mini-computers, can be hosted on client-server or other architectures, based on high-performance work-stations linked in networks. Such systems have been proposed within government organizations such as NASA and the Defense Department, and by numerous commercial firms.

By looking at the functions covered by two of these proposed architectures and applying our own spaceflight support experience, we have derived a superset of functions that covers all the aspects of satellite flight support. This set of functions facilitates comparison among the numerous approaches to distributed, open-system architectures that have been proposed in the past four years. We also discuss a loosely integrated ground support system prototyped at CSC in an effort to understand how to move to a distributed, open-system architecture while taking maximum advantage of the enormous amount of existing flight-proven software developed for mainframe- and mini-computer-based ground systems.

Spaceflight Ground Support Functions

The ground support functions found in the two sources investigated for this paper are summarized in Table 1. The first column lists the functions summarized by A. R. Stottlemyer and his co-authors in a paper proposing distributed architectures for NASA ground systems

Table 1 - Two sets of satellite ground support functions

Stottlemyer *et al.*	DoD ISC HCI
1 Mission Design	1 Schedule Resources
1.1 Orbit requirements and design	2 Create Satellite Support Plan
1.2 Attitude requirements and design	3 Update Satellite Support Plan
2 Remove communications artifacts	4 Configure, Test, and Verify System
3 Spacecraft position and orientation	4.1 Verify Configuration
3.1 Orbit determination	4.2 Test End-to-end Configuration
3.2 Attitude determination	4.3 Configure for Operations
4 Analysis of spacecraft operations performance	5 Perform Satellite Support
4.1 Trend analysis	5.1 Acquisition of Signal
4.2 Command Response	5.2 Verify Tracking
5 Analysis of scientific instrument performance	5.3 Verify Correct Telemetry Stream
5.1 Data quality	5.4 Verify Frame Synchronization
5.2 Measurement quality	5.5 Verify Command Link
5.3 Calibration	5.6 Perform Planned Commanding
6 Operations planning	5.7 Verify Satellite State of Health
6.1 Spacecraft operations	5.8 Produce Output Products
6.2 Instrument operations	5.9 Complete and Verify Support Activities
6.3 Support environment operations	5.10 Log Activities
6.4 Supporting analysis	5.11 Terminate Pass
7 Spacecraft command and control	6 Deconfigure Resources
7.1 Command generation	6.1 Deconfigure Resources
7.2 Command validation	6.2 Verify Deconfiguration
7.3 Command issue	7 Orbit Data Collection and Verification
8 Scientific data analysis	7.1 Collect Orbit (Tracking) Data
8.1 Data preparation and management	7.2 Verify Data
8.2 Analysis algorithm management	8 Attitude Data Collection and Verification
8.3 Support for data access and manipulation	8.1 Collect Attitude Data
8.4 Product generation and distribution	8.2 Verify Data
9 Data acquisition and management	9 State of Health Data Collection
10 System resource management	9.1 Request State of Health Data
10.1 Physical resources	9.2 Collect State of Health Data
10.2 Operations staff	9.3 Process and Verify Data
11 Integration and test	10 Orbit Determination and Planning
	10.1 Predict Orbit
	10.2 Plan Orbit Maneuvers
	10.3 Maintain Orbit Model
	11 Attitude Determination and Planning
	11.1 Plan Attitude Determination
	11.2 Plan Attitude Maneuvers
	11.3 Maintain Attitude Model
	12 State of Health Determination and Planning
	12.1 Determine State of Health
	12.2 Plan State of Health Activities
	12.3 Maintain State of Health Model

(Stottlemyer *et al.*, 1993).The functions in the second column are taken from a Defense Department standard drafted by the Integrated Satellite Control (ISC) Human Computer Interface (HCI) Working Group (ISC HCI Working Group, 1993). NASA Goddard's Mission Operations

Directorate has also begun an extensive campaign to take advantage of workstation-based, distributed architectures for satellite ground support. However, this effort, called the Renaissance Initiative, is newly begun and it is therefore premature to include it in this analysis.

These two sets of ground support functions represent different views of satellite ground support. The Stottlemyer *et al.* paper was written primarily to explore the feasibility of system architectures and is not meant to be an exhaustive analysis of the ground support domain. Their paper nonetheless contains a list of eleven high-level ground support functions that we have broken into subcategories to facilitate comparison with other function sets. This architecture analysis was one of the drivers of the Renaissance initiative and in this analysis we use it as a snapshot of the NASA ground support function set.

The Defense Department function set is taken from an appendix of a standard drafted to define DoD's view of the optimum interface between humans and computers for satellite ground support. In writing this standard, these authors also found that they needed a generic set of satellite ground support functions, which appears in this appendix and which we have taken to represent a picture of DoD satellite ground support.

In defining our superset of ground support functions, we made the following assumptions:

- only on-orbit operations considered in this analysis
- payload (instruments, e.g.) operations and planning not included
- integration of payload commands and schedules received through external interface included
- no particular institutional organization assumed, but system resources can be physically separated

We created the superset of functions appearing in Table 2 on the next three pages by combining the two function sets in Table 1 and adding elements drawn from our own ground support experience. We have tried to generalize functions. For example, NASA places considerable importance on managing onboard flight recorders to maximize scientific data return. A more general function might be the optimum management of onboard resources, for which different operations teams might have varying goals such as maximum observation time or extended mission life. One purpose of our function 1.3.7, *integrate commands to form command load*, is to optimize the planned command load within such constraints.

To organize the listed functions, we set up the seven main categories and sixteen subcategories shown in the light grey areas of Table 2. These areas are collectors of identifiable functions, which are in turn mapped against the other function sets. To facilitate comparison with reference functions, we have mapped them into our categories, using broad interpretations. Note that Stottlemyer functions 1.1 and 1.2 are not included, because they are requirements definition, hence prelaunch and not part of the on-orbit phase. This arrangement can be modified by adding or deleting lower level functions. As we extend this analysis to other mission phases, such as launch or end-of life, it is reasonable to anticipate that the function set will need modification.

The major categories were chosen by analyzing the reference function sets and other models, seeking high-level function collectors that would span the entire domain of on-orbit flight operations and would be significant for all identifiable missions. These categories are discussed below.

Defining the *spacecraft state* (1) in terms of a physical model and its state representation is the basis of the spacecraft mission control systems developed by the Altair Aerospace Corporation (Wheal, 1993). We have called this part of the spacecraft state the *vehicle state* (1.1), defined by the collection of its telemetry values. To fully define the concept of the spacecraft state, we have added the concept of the *dynamic state* (1.2), reflecting the fundamental flight dynamics definition of state as a set of parameters defining

Table 2 - Superset of ground support functions mapped against previous sets

SPACECRAFT GROUND SUPPORT FUNCTIONS	Stottlemyer *et al.*	DoD ISG HCI
1 SPACECRAFT STATE **1.1 Vehicle State**		
1.1.1 Monitor Health and Safety	6, 6.1	5.8, 5.9
1.1.2 Monitor Payload Status	5, 5.1	5.8, 5.9
1.1.3 Verify Commands	4, 4.2	5.9
1.1.4 Monitor OBC Image		5.9
1.2 Dynamic State		
1.2.1 Real-time Orbit Determination	3, 3.1	5.9
1.2.2 Off-line Orbit Determination	3, 3.1	10.1
1.2.3 Verify Orbit Maneuvers	4, 4.2	5.9
1.2.4 Real-time Attitude Determination	3, 3.2	5.9
1.2.5 Off-line Attitude Determination	3, 3.2	11.1
1.2.6 Verify Attitude Maneuvers	4, 4.2	5.9
1.3 State Transitions		
1.3.1 Generate and Validate Commands to Alter Vehicle State	7, 7.1, 7.2	2, 3, 12.1
1.3.2 Generate and Validate Commands for Orbit Maneuver	7, 7.1, 7.2	2, 3, 10.2
1.3.3 Generate and Validate Commands for Attitude Maneuver	7, 7.1, 7.2	2, 3, 11.2
1.3.4 Generate and Validate Flight Software Change	7, 7.1, 7.2	2, 3, 12.1
1.3.5 Generate and Validate OBC Image Commands	7, 7.1, 7.2	2, 3, 12.1
1.3.6 Generate and Validate Payload Commands	7, 7.1, 7.2	2, 3, 12.1
1.3.7 Integrate Commands to Form Command Load	7	2, 3
1.3.8 Uplink Command Load	7, 7.3	5.6
1.3.9 Uplink Real-time Command	7, 7.3	5.6
2 MISSION AND SPACECRAFT OPERATIONS **2.1 Planning and Scheduling**		
2.1.1 Predict Orbit	6, 6.4	2, 3, 10.1
2.1.2 Predict Orbit Event	6, 6.4	2, 3
2.1.3 Plan Orbit Maneuver	6, 6.4	2, 3, 10.2
2.1.4 Predict Attitude	6, 6.4	2, 3, 11.1
2.1.5 Predict Attitude Event	6, 6.4	2, 3, 11.1
2.1.6 Plan Attitude Maneuver	6, 6.4	2, 3, 11.2
2.1.7 Plan Spacecraft Contact	6, 6.4	2, 3
2.1.8 Plan Vehicle Activity	6, 6.1	2, 3
2.1.9 Plan Payload Activity	6, 6.2	2, 3, 12.1
2.1.10 Plan Flight Software Change	6, 6.1	2, 3
2.1.11 Plan System Configuration	6, 6.3	1, 2, 3
2.1.12 Plan Remote Resource Activity	6, 6.3	1, 2, 3
2.1.13 Integrate and Optimize Plan and Schedule	6	2, 3
2.2 Logging and Reporting		
2.2.1 Log Events	8, 9	5.10
2.2.2 Access Logs	9	
2.2.3 Create Spacecraft State Report		
2.2.4 Generate Mission Operations Report		
2.2.5 Generate Communications Report		
2.2.6 Generate Data Management Report	8, 8.3, 9	

SPACECRAFT GROUND SUPPORT FUNCTIONS	Stottlemyer *et al.*	DoD ISG HCI
2.2.7 Generate System Operations Report		
2.2.8 Generate Calibration Report		
2.2.9 Generate Simulation Report		
2.2.10 Generate Integrated Reports and Digests		
3 SPACECRAFT COMMUNICATIONS		
3.1 Ground RF Support		
3.1.1 Predict Attitude-independent Contact Times	6, 6.4	2, 3
3.1.2 Predict Attitude-dependent Contact Times	6, 6.4	2, 3
3.1.3 Compute Acquisition Data	6, 6.4	2, 3
3.1.4 Model Antenna Position and Mask	6, 6.4	10.3
3.1.5 Acquire Signal		5.1
3.1.6 Terminate Contact		5.11
3.1.7 Determine Onboard Oscillator Frequency	2	
3.1.8 Perform Frequency Compensation	2	
3.1.9 Perform Range Correction	3.1	
3.2 Tracking Data Streams		
3.2.1 Capture Tracking Data	2, 9	5.2, 7.1
3.2.2 Verify Tracking Data	9	5.2, 7.2
3.2.3 Sort and Sequence Tracking Data	9	7.1
3.2.4 Check Tracking Data Quality	3.1, 9	7.2
3.3 Telemetry Data Streams		
3.3.1 Capture Real-time Telemetry Data	2, 9	5.3, 8.1, 9.2
3.3.2 Verify Real-time Telemetry Data	9	5.3, 8.2, 9.3
3.3.3 Sort and Sequence Real-time Telemetry Data	9	5.4, 8.1
3.3.4 Check Real-time Telemetry Data Quality	9	8.2
3.3.5 Capture Playback Telemetry Data	2, 8, 9	8.1, 9.2
3.3.6 Verify Playback Telemetry Data	8, 9	8.2, 9.3
3.3.7 Sort and Sequence Playback Telemetry Data	8, 9	8.1
3.3.8 Check Playback Telemetry Data Quality	3.2, 5, 5.1, 8	8.2
3.4 Command Streams		
3.4.1 Verify Command Link	2	5.5
3.4.2 Command Echo		
4 DATA MANAGEMENT		
4.1 Archive		
4.1.1 Archive Telemetry Data	8, 8.1, 9	5.3
4.1.2 Archive Tracking Data	9	5.2
4.1.3 Archive Command Data	9	
4.1.4 Archive OBC Image Data	9	
4.1.5 Archive Processed Data		
4.1.6 Archive System Configuration Data	9	
4.1.7 Archive Logs	9	
4.1.8 Archive Reports	9	
4.2 Retrieval		
4.2.1 Retrieve Telemetry Data	8, 8.1, 9	8.1, 9.1, 9.2
4.2.2 Retrieve Tracking Data	9	7.1
4.2.3 Retrieve Command Data	9	
4.2.4 Retrieve OBC Image Data	9	9.1, 9.2

SPACECRAFT GROUND SUPPORT FUNCTIONS	Stottlemyer *et al*	DoD ISG HCI
4.2.5 Retrieve Processed Data		
4.2.6 Retrieve System Configuration Data	9	
4.2.7 Retrieve Logs		
4.2.8 Archive Reports		
4.3 Data Analysis		
4.3.1 Data Trend Analysis	4, 4.1	12.1
4.3.2 Engineering Analysis	4, 4.1	12.1
4.4 Reference Databases		
4.4.1 Telemetry Database		12.3
4.4.2 Command Database		12.3
4.4.3 Spacecraft Ephemeris		10.3
4.4.4 Solar, Lunar, and Planetary Database		10.3, 11.3
4.4.5 Geophysical Database		10.3
4.4.6 Time Reference Database		
4.4.7 Star Catalog		11.3
4.4.8 Spacecraft Properties Database		10.3, 11.3,
4.4.9 Rules Database		
5 SYSTEM OPERATIONS **5.1 Configuration**		
5.1.1 Configure Local Resources	10, 10.1, 11	4.1, 4.2, 4.3
5.1.2 Configure Remote Resources	10, 10.1, 11	4.1, 4.2, 4.3
5.2 De-Configuration		
5.2.1 De-configure Local Resources	10, 10.1, 11	6.1, 6.2
5.2.2 De-configure Remote Resources	10, 10.1, 11	6.1, 6.2
5.3 Remote Resource Transactions		
5.3.1 Send Message to Remote Resource	8, 8.3, 10.1	4.3, 6.1
5.3.2 Receive Message from Remote Resource	8, 8.3, 10.1	4.1, 6.2
5.3.3 Send Data to Remote Resource	8, 8.4	4.2, 6.1
5.3.4 Receive Data from Remote Resource	8, 8.4	4.2, 6.3
6 CALIBRATION		
6.1 Calibrate Spacecraft State		12.3
6.2 Calibrate Telemetry Conversions	5, 5.3	12.3
6.3 Correct Spacecraft Properties and Model	4.2	10.3, 11.3
6.4 Correct for Biases and Misalignment	3.2, 5, 5.3	11.3, 12.3
6.5 Calibrate Propulsion System	4.2	10.3
6.6 Calibrate Tracking Data	3.1	
7 SIMULATION		
7.1 Simulate Telemetry	11	4.2
7.2 Simulate Tracking	11	4.2
7.3 Simulate Commands	11	4.2
7.4 Simulate OBC	11	4.2
7.5 Simulate Vehicle State	11	4.2
7.6 Simulate Dynamics State	11	4.2
7.7 Simulate System Resources	11	4.2

the spacecraft position, velocity, attitude, attitude rates, and additional parameters needed to determine its dynamics. Carrying this concept to its logical conclusion, the process of commanding becomes one of making *transitions* (1.3) between states. Note that the command generation defined in this category refers to generating commands for uplink, distinguished from the command planning that appears in the next category. We made this distinction because of the potential applicability of rules-based systems to generating and integrating safe, optimized command loads.

The concept of *mission and spacecraft operations* (2) appears in all the function sets. We have divided this area into two parts. *Planning and scheduling* (2.1) appear in both of the reference function sets. The *logging and reporting* (2.2) category is less well represented in the references. Here logging refers to making records of actions taken, plans executed, and events that have occurred. Reports are passed among flight team members and to outside parties, and are taken from logs, data, and analysis of data. In all the superset categories the low-level functions are stated as singular, but can be combined to make complex functions for multiple spacecraft. For example, planning an orbit maneuver might require optimizing fuel consumption, the target orbit, and tracking and communication opportunities, requiring iteration and integration of the individual functions.

Spacecraft communications (3) is taken from analysis of Goddard mission operations. *Ground RF support* (3.1) covers the functions needed to establish radio-frequency links between the spacecraft and ground controllers, including antenna modeling and signal management. Two types of data may be received: *tracking* (3.2), bearing position and velocity information, and *telemetry* (3.3), reflecting the vehicle state. Data flows to the spacecraft as *commands* (3.4), effecting state transitions.

Large volumes of data, particularly received from the spacecraft and resulting from processing, are characteristic of the ground support domain, making *data management* (4) essential. As in most application domains, this category includes *archive* (4.1), *retrieval* (4.2), and *analysis* (4.3) of data. We have additionally added *reference databases* (4.4) such as star catalogs, telemetry conversions, or rules for applied intelligence processing.

As found in both reference function sets, *system operations* (5) require functions of their own. NASA and DoD functions differ sharply in this area. For DoD spacecraft, a ground support system deals with multiple spacecraft, while for a NASA satellite there is generally a dedicated ground system. Using one system for several spacecraft makes *configuration* (5.1) and *de-configuration* (5.2) significant problems. A NASA flight operations team generally relies on ground resources physically remote from its control center, unlike DoD facilities that place all the resources in one place. Dealing with distant antennas or networks requires additional communication and data channels for *transactions with remote resources* (5.3).

We have added the category *calibration* (6) to reflect the need to tune the performance of the spacecraft and ground support system based on data from past performance. Calibration results appear in the reference databases of category 4.4.

There is some question whether *simulation* (7) is a part of flight operations, or a test-and-integration function only. We include it on the grounds that changes onboard the spacecraft, evolution of the mission objectives, and pursuit of operational efficiencies will make modification of the system and its configuration necessary, requiring testing throughout the mission. Also, some mission teams utilize simulated data for training, maneuver prediction, and operational activity modeling.

Integrated Ground Support System Prototype

In 1992, CSC began work on a prototype ground system proposed by R. D. Werking (Werking and Kulp, 1993), called the CSC Integrated Ground Support System (CIGSS). The goal was to demonstrate that the functionality needed for ground support could be placed on a RISC-based workstation under UNIX by taking maximum advantage of the large amount of existing ground support software. Components were to be re-hosted as necessary from other platforms and operating systems, and loosely integrated by creating file interfaces between pairs of programs. Components were to be drawn from the NASA Goddard software legacy, obtained from vendors, or developed if necessary.

A working prototype has been developed and demonstrated, showing the feasibility of this approach and giving some insights into the software and system engineering needed to exploit the large amount of existing ground software on workstations. For example, B. S. Groveman and his co-workers have rehosted FORTRAN programs from IBM mainframe computers, finding the transition of computational modules straight-forward, but the creation of user interfaces more challenging. (Groveman *et al.*, 1994).

The functions originally proposed for this system were command and control, health and safety monitoring, flight dynamics, mission planning and scheduling, and payload data management functions. However, in looking at how to combine candidate components, we soon found it necessary to have a function set that enabled us to describe what a particular set of components could do in combination. This experience led us to create the superset of ground support functions.

Conclusions

We expect that future ground systems will be integrated from existing components, certainly with some modification and tailoring, but rarely developed through the traditional lifecycle. Long-time spacefaring agencies such as NASA and DoD possess enormous legacies of expensively acquired, flight-tested software, and an ever-growing number of commercial vendors are offering products for spacecraft ground support. The result is a range of choices for nearly all the functions needed for a ground support system, albeit in complicated combinations needing some form of evaluation and validation.

We have, therefore, developed a generic set of ground support functions to guide evaluation of the functionality of components and to assist in choosing an appropriate set to integrate. With these goals in mind, we intend to extend this exercise in four ways. First, the ground support domain is large and complex, and its boundaries are not sharp, so we expect to adjust our functions as we continue its analysis. Second, we intend to cover other mission phases. Third, we intend to evaluate different operations concepts and user interfaces as a way to minimize operations costs. Finally, the function set would make a far better evaluation tool if it has quantitative performance indices, which we plan to determine through our continued evaluation of legacy software and COTS products.

References

Groveman, B. S., Liang, E. Y., Starbuck, R. A., Tamkin, G. S., & Boland, D. E. (June 1994). A Recommended Approach to Rehosting IBM-Mainframe FORTRAN Software Systems to UNIX Workstations, *Fourth Annual CSC Technology Conference.* Atlanta, GA.

Integrated Satellite Control Human Computer Interface Working Group, Department of Defense (August 1993). *Human Computer Interface Standard, Draft 1.0, Appendix A1.*

Stottlemyer, A. R., Jaworski, A., & Costa, S. R. (October 1993). New Approaches to NASA Ground Data Systems, *Proceedings of the Forty-fourth International Astronautical Congress, Q.4.404.* Graz, Austria.

Werking, R. D., & Kulp, D. R. (September 1993). Developing the CSC Integrated Ground Support System, *Poster presented at the Seventh Annual AIAA/Utah State University on Small Satellites.* Logan, UT.

Wheal, C. A. (1993). Application of State Space Modeling Techniques to Satellite Operations, *Altair Aerospace Corporation.* Bowie, MD.