NASA CONTRACTOR REPORT 177337

Study of Application of Space Telescope Science Operations Software for SIRTF Use

(NASA-CR-177337) STUDY OF APPLICATION OF SPACE TELESCOPE SCIENCE OPERATIONS SOFTWARE FOR SIRTF USE (TRW, Inc., Sunnyvale, Calif.) 81 p HC A05/MF A01 CSCL 03A
N85-18080
Unclas
G3/18 17761

F. Dignam
E. Stetson
W. Allendoerfer

CONTRACT NAS2- 11938
February 1985

NASA

NASA CONTRACTOR REPORT 177337

Study of Application of Space Telescope Science Operations Software for SIRTF Use

F. Dignam
E. Stetson
W. Allendoerfer

TRW
Sunnyvale Laboratory
Sunnyvale, California

Prepared for
Ames Research Center
under Contract NAS2-11938

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

1. INTRODUCTION

1.1 Purpose

In the "SIRTF Free Flyer, Phase A System Concept Description" (Document no. PD-1006, May 3, 1984), NASA/Ames describes a Space Infra-Red Telescope Facility (SIRTF) concept of operations. Space Telescope (ST) is a similar NASA research astronomical observatory, having most of the same operational functions as SIRTF. In this context, TRW and NASA/Ames agreed to examine the applicability of software TRW has developed for the ST Science Operations Ground System (SOGS) for use in SIRTF. This final report is a result of that study effort.

The study was organized into tasks. The purpose of Task 1 was to evaluate the design and development of the ST science operations software and compile a history of lessons learned, both positive and negative, that would benefit SIRTF. Task 2 consisted of assessing the applicability of operational ST SOGS software for use in SIRTF and the degree of modification necessary for that conversion.

1.2 Summary of Results

The design and development of the ST SOGS project, like most large efforts, encountered a number of problems. All of these were manageable and the program appears to be headed for a successful completion. Analysis of this history has resulted in 49 specific recommendations for SIRTF. The recommendations are organized into three categories (requirements, Science Instruments, and contract phasing) and are described in section two.

SIRTF, to the level of detail specified thus far by NASA, is compatible with the environment, concept of operations and functions of ST SOGS. Our study results indicate that nearly half of the software design and source code might be used for SIRTF. This assessment is dependent on the following important assumptions. Transportability of this software requires, at minimum, a compatible DEC VAX-based hardware architecture and VMS operating system, system support software similar to that developed for SOGS, and continued evolution of the SIRTF operations concept and requirements in a manner which is compatible with the ST SOGS operation. These assessments of transportability are described in section 3.0.

1.3 SOGS System Description

A basic understanding of the structure and design of the ST Science Operations Ground System is necessary for the discussions presented in this report. Hence, a brief overview is presented here.

The Space Telescope (ST) is a large, versatile, high-resolution telescope with a complement of five scientific instruments, including two cameras, two spectrographs, and a photometer. The ST Fine Guidance System (FGS) is also used to make astrometric measurements. In addition to permitting observations at wavelengths inaccessible from the ground, the absence of atmosphere allows observation in the visible region of the spectrum to be made at the full resolution of the telescope. ST will be operated in orbit as an astronomical facility and will provide observational capabilities well beyond those of existing ground-based telescopes.

The ST will be carried into orbit by a Space Shuttle from Kennedy Space Center. It will be inserted into a circular orbit having a nominal 500 KM altitude and a 28.5 degree inclination. While the ST is in orbit, it can be revisited by the Shuttle for maintenance by astronauts and, if necessary, boosted into a higher orbit. Orbital maintenance can be performed by replacing spacecraft subsystem components or complete science instruments. The ST can be retrieved and returned to the ground for major servicing, instrument updating, and telescope refurbishment.

Figure 1.3-1 depicts the communication paths and major ground elements of the ST Program. The data transferred to and from the ST are sent via the TDRSS satellites, the TDRSS White Sands Ground Station, and via DOMSAT to the NASCOM facility at GSFC.

As shown, the ST Mission Operations Ground System (MOGS) consists of two facilities dedicated to the ST Program, namely the Space Telescope Operations Control Center (STOCC) and the ST Science Institute Facility (ST ScIF), as well as support from other GSFC facilities. The STOCC is located at GSFC and consists of the Payload Operations Control Center (POCC) and the Science Support Center (SCC). The ST ScIF is located on the grounds of Johns Hopkins University, Baltimore, Maryland.

Figure 1.3-2 depicts the functions performed by SOGS and its interfaces with other elements of the MOGS. The POCC, as shown, is responsible for all spacecraft scheduling and operations, including the command message function, health and safety check, and spacecraft telemetry processing. The POCC sends real-time and tape recorded science data and science instrument telemetry to SOGS. SOGS sends the POCC the science schedule for processing into spacecraft and instrument commands and requests for real-time operations.

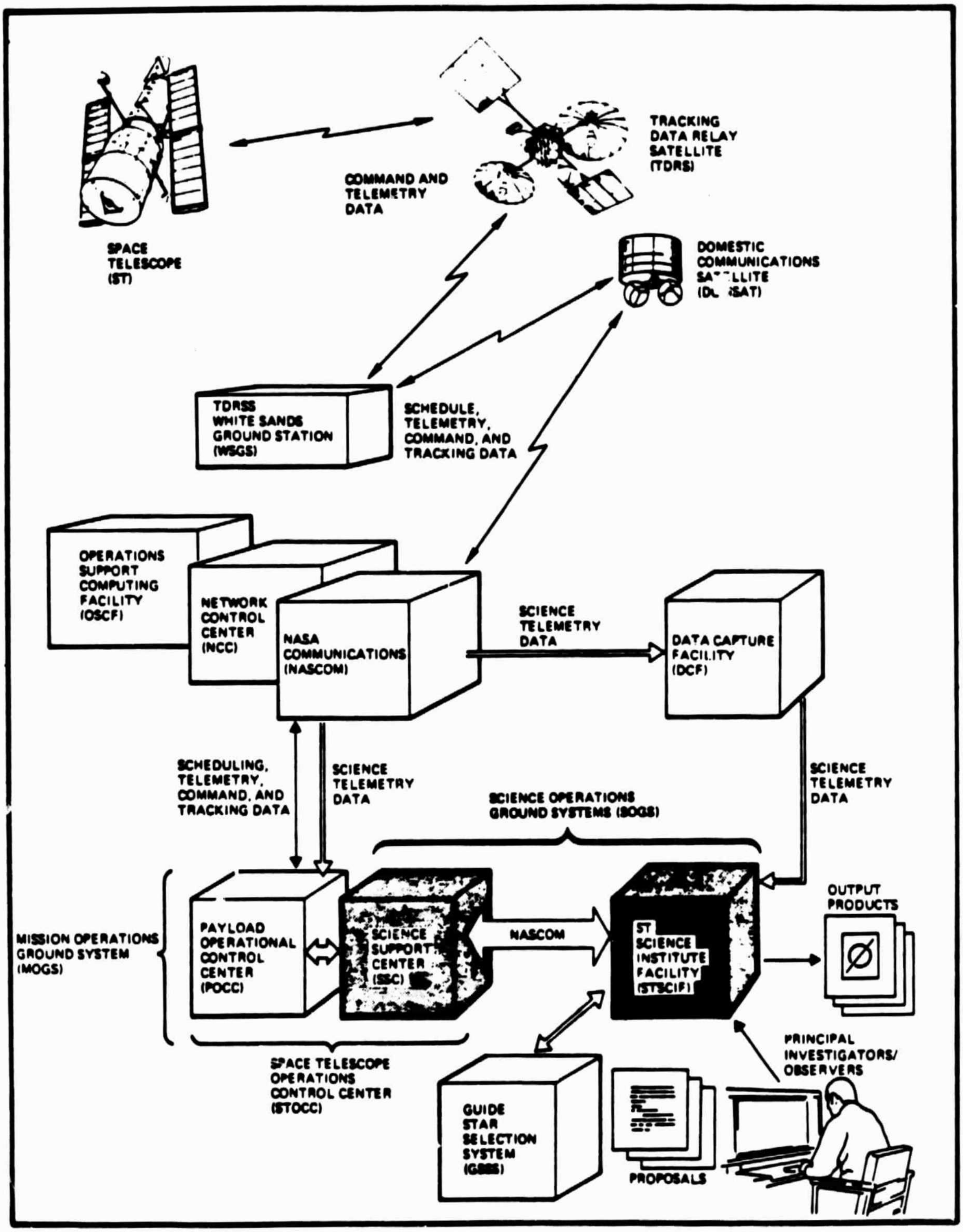


FIGURE 1.3-1 SPACE TELESCOPE SYSTEM

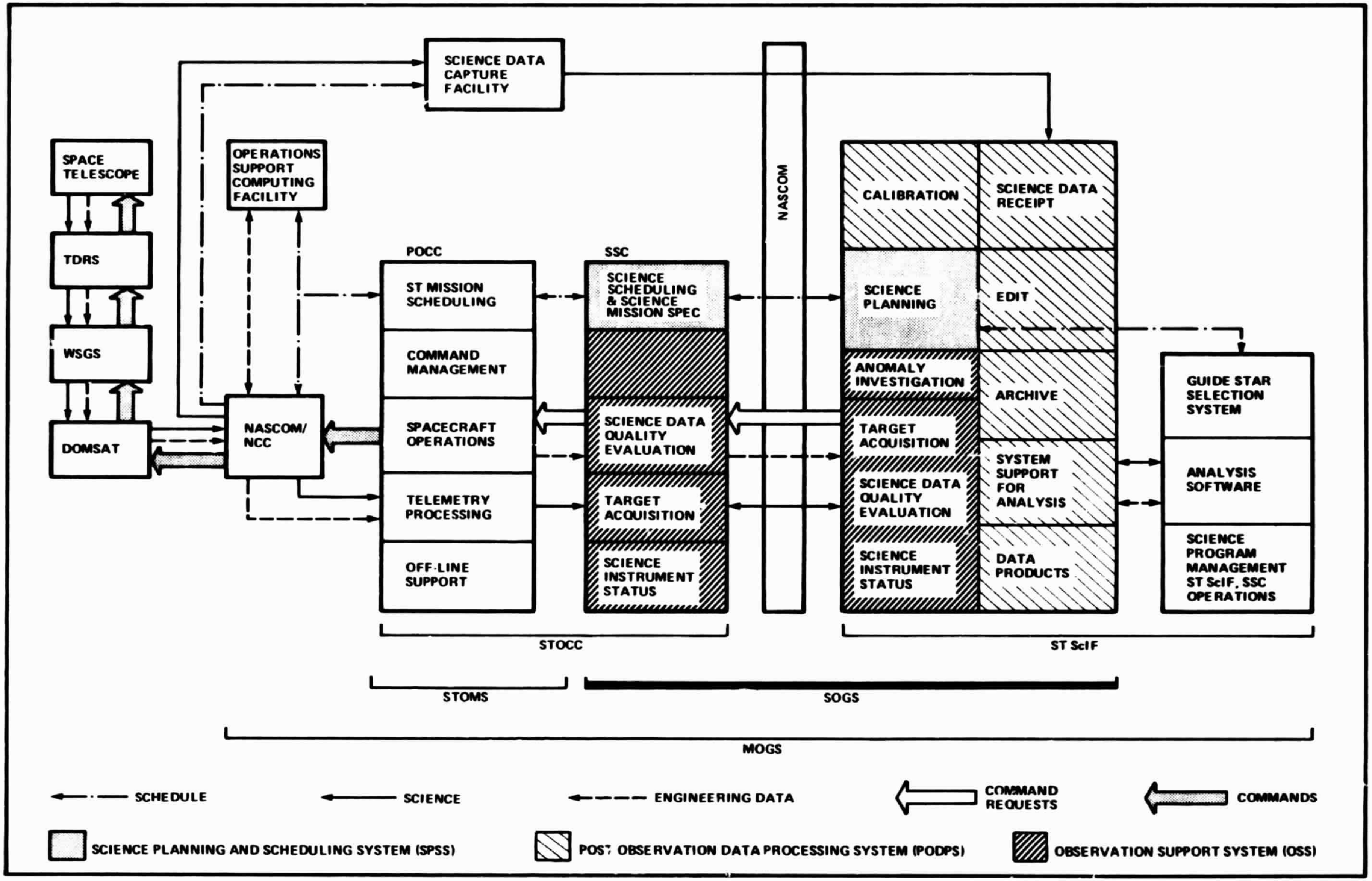


FIGURE 1.3-2 ST MISSION OPERATIONS GROUND SYSTEM (MOGS)

The GSFC Data Capture Facility (DCF) receives, checks, reformats, sorts, and stores the science data from the science instruments on the ST. At scheduled intervals, the DCF sends this science data to SOGS.

The ST Science Institute (ST ScI) is responsible for the science program, including the operation and management of the SOGS after delivery. This organization is also responsible for the development of the Guide Star Selection System (GSSS), which chooses the stars on which the Fine Guidance System will align.

In science planning, SOGS requests guide stars and astrometric reference stars for each target in the science plan. The ST ScI is also developing Science Data Analysis Software (SDAS), which will be integrated and executed within the SOGS Post Observation Data Processing System (PODPS) environment.

The major functions of SOGS required to support the overall mission of the Space Telescope can be characterized as follows:

a. Provide the equipment and software to plan and schedule the utilization of the science instruments.

b. Provide the equipment and software to monitor, command, and control the science instruments in real time through observations of the science data.

c. Provide the equipment and software to catalog, sort, calibrate, and archive all the science data and provide output products.

d. Provide the equipment and software utilities to support science data analysis.

The following subsections first present the general design of the SOGS system and then an overview of its operations.

1.3.1 Overview of the System Requirements

The design of the SOGS system is responsive to the requirements as stated in DRD-SOGS-SE-Ø6-1 (reference 1.4b).

SOGS can be divided into four major components: hardware, software, interfaces, and operations. The following paragraphs provide an introduction to each of these components.

1.3.2 Hardware

SOGS is a distributed data processing system, implemented as shown in Figure 1.3-3. The system employs Digital Equipment Corporation (DEC) VAX 11/78Ø computers with two at the SSC and four at the ST ScI. These computers are assignable to the SOGS application systems, with Science Scheduling and Observation

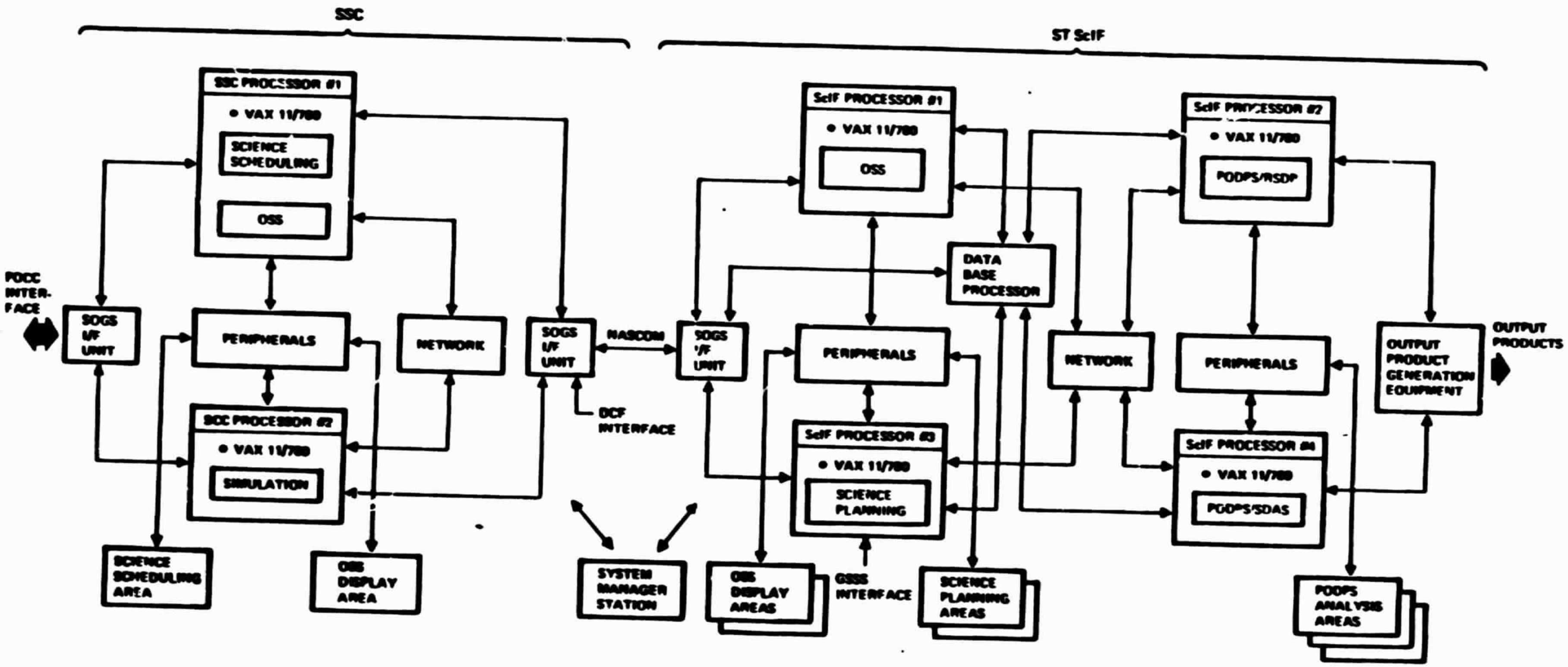


FIGURE 1.3-3 ST SOGS HARDWARE ARCHITECTURE

Support at the SSC and Science Planning, Observation Support, and Post-Observation Data Processing at the ST ScIF.

The computer resources are sized and the system designed so that operations can continue with a failed CPU or peripheral or when a CPU is taken off-line for maintenance or software development. The OSS, PODPS, and SPSS have minimum availabilities of Ø.9985, Ø.975, and Ø.975, respectively.

Each of the SOGS systems has several workstations composed of alphanumeric terminals and graphics and/or image displays. Hardcopy output is required at some stations. The PODPS, in addition, has film, plot, and tape outputs and sufficient tape drives to generate and read the science data archives.

1.3.3 Software

The SOGS software is organized into four functional elements: Support Software (SS); Science Planning and Scheduling System (SPSS); Observation Support System (OSS); and Post-Observation Data Processing System (PODPS).

1.3.3.1 Support Software

The SS provides SOGS with the system and support functions needed by the three applications systems (SPSS, OSS and PODPS). The SS provides augmentations to the DEC VMS operating system for initialization, termination, recovery, etc. Since SOGS is implemented in a distributed data processing system, the SS provides the needed network and system management software. Communication software needed to interface with peripheral and external data, as well as common image and graphics routines, are included. The SS provides a command language interface, supporting both familiar and unfamiliar users with menu, command, and procedure modes, as described in Appendix B of the SDAS ICD (reference 1.4f). SS also includes a data simulator which will provide simulated data for the integration, test, and acceptance of SOGS.

1.3.3.2 Science Planning and Scheduling System (SPSS)

The SPSS includes all of the SOGS functions which result in the preparation of complete specifications for ST science operations. These functions provide for a series of processes by which astronomy proposals are transformed into viable ST instrument activities. The primary thrust of SPSS is the efficient organization of the mission and the utilization of instrument capabilities to achieve as many scientific objectives as possible.

Science planning and scheduling accomplishes the following:

a. Support interpretation of proposals and evaluation of the appropriateness and feasibility of the requested observations.

b. Identify constraints which limit the ability to schedule each observation set.

c. Schedule individual observation activities and establish the time frames for their conduct.

The nominal result is a description of monthly science objectives for the ST, resolved to the weekly level.

The science scheduling function begins its efforts on a month's activities approximately 60 days before the beginning of that month. It accepts as input the monthly science plan and organizes the observations into day-to-day and orbit-to-orbit sequences and places them within a framework established by ST orbit geometry and event timings determined by orbit geometry. It establishes observation spacings and permits SI subsystem maintenance and calibration activity spacings which fulfill science requirements and accommodate orbit-imposed constraints.

The Science Mission Specification (SMS) is generated and transferred to the ST POCC mission scheduling activity as a constraint-free, full description of the science activities and associated requirements for supporting spacecraft operations.

This description is used by the POCC in the generation of the ST mission schedule. In the development of the Science Mission Specification, proposed science activities which result in constraint or restriction violations are identified and conflicts are resolved.

1.3.3.3 Observation Support System (OSS)

The OSS is that portion of SOGS which handles real-time science and science engineering data. Just as the ST POCC controls the operations of the spacecraft, the OSS is responsible for the real-time control of science observations. The major OSS science control functions are Target Acquisition and Verification, Science Data Quality/Utility Evaluation, Science Instrument (SI) Status Monitoring, and Command Request Support. Each major activity is briefly described below.

One of the basic advantages of the ST is its capability to observe sources of extreme faintness and to work in spectral regions inaccessible to ground-based observatories. The characteristics, locations, and even the existence of some potential targets for the ST are, therefore, uncertain. As a result, the baseline design of the ST and SIs include modes for fixed acquisition and ground-assisted acquisition.

For ground-assisted acquisition, the instrument transmits preliminary data to the ground (images for the cameras and acquisition field or "pseudo images" for the nonimaging SIs).

The OSS receives, processes, displays, and permits user interaction with these data for the purpose of observer verification and selection of the target of interest. The OSS provides the capabilities to transfer pointing offset requests and allowable instrument configuration changes to the PORTS for transmittal to the ST.

Just as there is uncertainty in the locations of some of the targets for ST, so too will there be some uncertainty in the characteristics of these targets. Rather than gather data on targets which are not of interest, or which are of interest but for which incorrect SI parameter settings have been planned, the OSS provides observers with the operational flexibility to select from pre-planned observing options. The OSS receives, in near real time, a subset of SI science data frames and permits the observer to evaluate the quality and utility of these data. To support this evaluation process, the OSS provides the capability for display and selected processing of these data.

An additional major functional requirement of the OSS is the capability to monitor SI performance and status. This function is distinct from SI health and safety monitoring, which is a POCC requirement. The OSS is able to receive SI engineering data from the POCC and process and display these data. The purpose of these data is to enable trained OSS operators to assist the general observer in assessing the performance of a given SI as it may impact the current observation.

1.3.3.4 Post-Observation Data Processing System (PODPS)

The ST ScI PODPS is that portion of SOGS which receives, edits, calibrates, and archives ST science data and supports user analysis. Prior to arrival at the ST ScIF, all data will have had the transmission codes removed by the NASA data receipt function at the Data Capture Facility (DCF). The PODPS also has the capability to receive, catalog, and store additional data from other SOGS elements such as definitive orbit data, astrometry science data, SI engineering data, and other ancillary ST data.

The ST Science editing process prepares the data for archiving and for calibration. Calibration is the processing applied to SI science data to remove systematic errors and instrument signatures. The calibration process utilizes SI-specific data bases and algorithms provided by the IDTs. Calibrated SI data are also placed in the archives. Astrometry science data, acquired from the output of the ST Fine Guidance Sensors, are also received and archived.

Both edited and calibrated data are made available by the PODPS for the analysis function. Analysis is the process of extracting scientific information from the data. The Science Data Analysis Software (SDAS) will be provided by the ST ScI, will have an interface compatible with the PODPS, and will operationally execute with PODPS. PODPS provides specific image, graphics, and other math processing utilities for use by SDAS.

The results of these functions are data products which will be made available to observers. Data output product generation equipment is provided as part of the PODPS for generating magnetic tapes, photographic products, and plots.

1.3.4 Interfaces

The SOGS design accommodates computerized interfaces to three other ST systems: POCC, GSSS, and the DCF as well as providing a communications link between separate SOGS facilities located at GSFC and Johns Hopkins University. In addition, SOGS provides for media interfaces containing support material: input proposals from astronomers, input orbit data tapes from JPL, and output products in the form of plots, tapes, and pictures.

1.3.5 Operational Support

The SOGS operational environment consists of vendor and applications software, a high-level command language, and user and product work areas. The application software is both real-time data driven and interactive. Users at workstations perform their tasks utilizing the COMET language. Operations personnel, on a planned schedule, tend to the archives and output products generated by SOGS. A System Manager, through a single console, interfaces with the computer at both sites. The OSS system at the SSC, to support the real-time data collection function, is the only SOGS system requiring 24-hour operations.

1.4 Reference Documentation

a. STE-13-NAS5-27001 "Functional Specification for the Space Telescope Science Operations Ground System", Change 15, 19 October 1984.
b. DRD-SOGS-SE-06-1 "Design Manual, Requirements Section", first issue April 29, 1982; Change 7, 23 October 1984.
c. SMO-1000 "Space Telescope Mission Operations Functional Requirements", Revision C , January 1984.
d. Statement of Work for Application of Space Telescope Science Operations Software for SIRTF Use, 1985.
e. SO-07 "Science Operations Concepts", first issue April 6, 1982; current Revision B2, 23 November 1983.
f. DRD-SOGS-CM-04, Science Data Analysis Software to Post Observation Data Processing System Interface Control Document, Revision B, 21 April 1984.
g. "SIRTF Observational Operations Concept", PD 1005, February 3, 1984.

1.5 Glossary

1.5.1 Definition of Terms

calibration	processing of science data to remove instrument signatures
constraints	operational limitation imposed on the use of the hardware that must not be violated in either planning or operations. This includes features or characteristics of the hardware inherent to the design which, if violated, could cause physical damage
engineering data	telemetry data stream containing instrument and S/C data not including the astronomical data gathered
program	overall effort to build, fly, and operate Space Telescope
project	one of the many contracts in support of ST Program
restrictions	operational limitation imposed on use of the hardware that may be violated if the trade-off between the desired operation or data and the resulting risk of system degradation is operationally acceptable and authorized by the Mission Operations Manager
science data	telemetry data stream containing the astronomical data
segment	one of the components of the ST system (e.g., SOGS)
system	the entire ST, from S/C through ground segments

1.5.2 Acronyms

ARC	Ames Research Center, NASA
CDR	Critical Design Review
COMET	Command Executive Translator, interactive command language
DCDS	Distributed Computing Design System
DCF	Data Capture Facility
DOMSAT	Domestic Communications Satellite
FGS	Fine Guidance Sensor
GSFC	Goddard Space Flight Center, NASA
GSSS	Guide Star Selection System
ICD	Interface Control Document
IDT	Instrument Development Team or Investigation Definition Team (same team, different names)
MMI	Man-Machine Interface
MOC	Mission Operations Contractor
MOGS	Mission Operations Ground System
NASCOM	NASA Communications Network

OSS	Observation Support System
PASS	POCC Application Software Support
PDB	Project Data Base
PDR	Preliminary Design Review
POCC	Payload Operations Control Center
PODPS	Post Observation Data Processing System
PORTS	Preliminary Operations Requirements and Test Support
PRR	Preliminary Requirements Review
RFP	Request for Proposal
RID	Review Item Discrepancy
SDAS	Science Data Analysis Software
SI	Science Instrument
SIRTF	Space Infrared Telescope Facility
SMS	Science Mission Specification
SOGS	Science Operations Ground System
SOW	Statement of Work
SPSS	Science Planning and Scheduling System
SS	Support Software
SSC	Science Support Center
ST	Space Telescope
STOCC	Space Telescope Operations Control Center
ST ScI	Space Telescope Science Institute
ST ScIF	Space Telescope Science Institute Facility
SYSREM	System Requirements Engineering Methodology
TDRSS	Tracking and Data Relay Satellite System
VAP	Verification and Acceptance Program

2. TASK 1 - LESSONS LEARNED

2.1 Introduction

2.1.1 Purpose

The purpose of Section 2 of this report is to document the findings of Task 1 of the SIRTF Study SOW (ref. 1.4d). Efforts carried out under this task were:

a. A review of the SOGS design and development history with the intent of identifying lessons learned, both positive and negative, that might be applicable to SIRTF, and,

b. The documenting of these lessons, including recommendations for further studies.

2.1.2 Scope

This analysis is limited in scope to what TRW has learned through our ST SOGS Project. While it focuses primarily on SOGS, it also presents lessons learned from how other parts of the ST Program impacted the SOGS effort. There are undoubtedly other lessons that other parts of the ST Program have learned that have not impacted SOGS.

It is the nature of a lessons learned report to emphasize the problems encountered and what was done or should have been done about them and to minimize the many elements of the effort that have been trouble free. As of this writing, the SOGS system has completed its first factory acceptance tests and is preparing for the first delivery to NASA GSFC. While problems were encountered early in the program, SOGS is considered by both TRW and the customer to be headed for a successful completion.

Many of the lessons learned and documented below were learned by observing them as being successfully applied to ST SOGS; others were discovered and adopted later in the program to recover from some problem; and still others, in retrospect, would have helped the program reach its successful completion more easily.

2.1.3 Report Organization

The lessons learned have been divided into three categories:

a. Requirements: a discussion of the content, timeliness, and proper allocation of the system and segment requirements, including each SOGS application subsystem as well as the external interfaces and the impact of these on the SOGS development;

b. Science Instruments: a consideration of the Science Instruments (SIs), their data streams, and how their designs impacted SOGS; and finally,

c. Contract Phasing: an analysis of when various segments of the ST Program were started relative to each other and the impact of this phasing on SOGS.

Where an early study might help avoid a potential problem being discussed, it is identified in line with the discussion and recommendations. The order of presentation of recommendations was driven by the organization used, and thus the numbering system implies no priority or significance.

In many of the recommendations that follow, a working group or central committee is suggested as a way to avoid a particular problem. On the ST Program, such working groups were usually a meeting of equals where problems were surfaced and discussed, but rarely solved without strong government intervention.

1. RECOMMENDATION: Working Groups, Technical Interface Meetings, and other multi-segment committees need to have strong leadership with the incentive, authority, and resources to surface problems, arrive at solutions, and direct implementation of the solutions.

2.2 Requirements

2.2.1 General

The SOGS Functional Specification (ref. 1.4a) is the contractual document that established the basic capabilities to be implemented by SOGS. These were expanded, analyzed, and documented as testable requirements by TRW in the first several months of the contract. In the period that these were being written, TRW was also interviewing the various Instrument Development Teams (IDTs) to help us understand their instruments, their required instrument ground processing algorithms, and their expectations of SOGS. The interviews led to a realization that many desired or required capabilities were not in the Functional Specification and, thus, were not in the TRW baseline for SOGS. In addition, the ST SCI published an Operations Concept Document (ref. 1.4e) identifying numerous missing but required capabilities. A composite of these missing capabilities was documented in Section 10 of the first version of the SOGS Requirements Specification (ref. 1.4b). This section caused significant discussion at the Preliminary Requirements Review (PRR) and led to an unusually large number of Review Item Discrepancies (RIDs), but it identified holes very early and led to contract changes that implemented the critical functions.

2. RECOMMENDATION: In early requirements documents and reviews, encourage the open discussion of what is needed but not currently in the contractual baseline. Analyze and direct implementation of required new functions before design starts.

Part of the large number of RIDs was due to redundant and inappropriate (to SOGS, not the ST Program) comments. At subsequent reviews, the total number of RIDs submitted to TRW was significantly reduced by prescreening the comments through a committee consisting of GSFC and user representatives.

3. RECOMMENDATION: Review comments submitted at customer and user reviews should be filtered by a committee of system-level users and system engineers to redirect comments not directed at the segment under review and to combine similar comments into one general comment for segment evaluation and response.

Throughout the life of any program, new and overlooked capabilities will be identified. For example, while it has been recognized for a long time, the ST Program has relatively recently addressed the fact that the current implementation for observing moving targets will not meet its minimum scientific requirements. The problem was identified, a working group of impacted segments was formed, and a resolution was arrived at with quick direction for the various contractors to proceed with implementation. Other, less global changes were not processed with this speed and clear direction. This not only delayed the implementation of the new feature, but frequently slowed progress in the ongoing design due to the uncertainty of what was in, what was out, and the impact of expected additions.

4. RECOMMENDATION: An identified missing capability should be addressed quickly by a group consisting of all (even peripherally) involved parties (contractors, government, and users); this group must have the authority, resources, and incentive to arrive at a decision and direct its implementation.

5. RECOMMENDATION: The contracting agency should perform a risk analysis to estimate the amount of reserve required to cover potential changes. This amount, which might be as high as 25% or more of the contract values (SOGS alone has approximately doubled in cost), should be held within the government program office for allocation to different segments as required.

As needed extensions and modifications were identified and TRW was directed to implement them, it was frequently difficult to integrate the new work into the existing schedule and to continue towards fixed milestones. Internally, TRW had some phasing of development, but externally there was initially only a single

Preliminary Requirements Review (PRR), a single Preliminary Design Review (PDR), a single Critical Design Review (CDR), a single acceptance test, and a single delivery. By necessity, phased reviews and deliveries have replaced these single milestones, but a preplanned recognition that the development should be phased would have made development and change implementation smoother.

6. RECOMMENDATION: Segment development should be divided into phases, each phase having a separately scheduled set of review, test, and delivery milestones (along with a single top-level, segment-wide review). The phases should be established early in the segment contract (if not before the award) and should be structured such that high risk elements are developed in early phases, required building blocks (e.g., support and operating system software) are developed in early phases, and areas where change is most likely deferred to later phases. As add-on work is required, it should be allocated to future phases rather than impacting current phases whenever possible.

The ST Program was necessarily broken up into several segments; SOGS is just one segment of the ground system. The allocation of capabilities to segments was aimed at keeping interfaces and overlap to a minimum and was successful at this (with a few exceptions discussed below). It was then assumed that each segment could proceed to work from their functional specification and interface documents. Unfortunately, many requirement, design, and scheduling decisions that any individual segment makes are dependent on many of the other segments' requirements, designs, or schedules. For example, observation branching (where an observer can decide in real time between several preplanned alternative observations) was not consistently specified or designed between ST segments. Branching involves SOGS, PASS, PORTS, and the spacecraft on-board computers, but the problem was not addressed from the top scientific objective down to requirements for each segment and, while it now has the required functionality, it is not implemented in the simplest manner it could have been and will not operate as readily as one might desire.

7. RECOMMENDATION: A single top-level system engineering function must be provided (either by the government or a system engineering contractor) to monitor the capabilities being implemented by each segment, to assure correct allocation, interpretation, and implementation, and to assure that the total program is satisfying its scientific goals. This function must span the ground, the spacecraft, and the science instruments. The organization performing this function must have the authority, resources, and incentive to initiate corrective actions in a timely manner.

8. RECOMMENDATION: Each segment should be given system-wide visibility into the capabilities being built in other segments as well as their current status. Attendance by representatives of each segment at technical reviews of all interfacing segments should be encouraged (if not required), and monthly status/progress reports should be distributed to all segments.

In addition to proper allocation of the capabilities, performance requirements must be carefully allocated from the top scientific objectives down. For example, if a real-time optical filter change decision is to be allowed (in OSS), response time allocations should include: time for the initial observation, time to downlink data, time to format data for display to the observer, time for ground analysis algorithms, time for analyst thought, time for developing commands, time for command uplink, and time for filter reconfiguration. If all of these times are not allocated from the top down prior to the start of design, the end result could be long delays in performing such a simple function.

To complement these performance requirements, average and worst case data bandwidths, volumes, and frequencies, or at least realistic ranges, are needed early in the program. The combination of performance and data volumes and rates is an early driver in developing a ground system hardware and software architecture; without these, worst case assumptions will be used that frequently lead to an over-designed system.

9. RECOMMENDATION: Performance and data volume requirements must be determined early and allocated from the top down to all segments.

One way to limit the impact of not being able to accurately estimate maximum data volumes is to put surge buffers on the front end of the ground system where possible. The ST Program put the Data Capture Facility (DCF) in front of SOGS to capture the science data from ST and buffer it to deliver it to SOGS at something closer to an average rather than maximum rate. This allowed SOGS to design its complex algorithms and archiving processing to the lower average numbers. SOGS could have performed the buffering itself to provide relief to the algorithms if the receipt-to-archives time limit were stated as a long term average rather than a short term requirement. The front end to OSS (the PORTS) has absolutely no buffering capability for real time data, not even a few micro-seconds. This forced OSS to be designed for a worst case loading plus a no-delay acceptance of real-time data.

10. RECOMMENDATION: Small (on the order of seconds) data surge buffering in systems upstream from the science ground segment relieves the real-time data receipt problem and is recommended. Longer term buffering should be implemented on non-real-time portions of the system to allow algorithm and archiving design to less than worst case data volumes; these buffers can be implemented in a front-end segment (such as the DCF) or as a front-end within the science ground segment.

There is a significant amount of overlap in the functions of the PORTS and the DCF with respect to science data. For example, both receive identical data from NASCOM and both perform bit reversal of tape playback data before passing the data to SOGS. The DCF additionally provides temporary data archiving (until PODPS has completed its archive function), data sorting, and bit error correction (the data sorting function provides similar capabilities to those performed in OSS).

11. RECOMMENDATION: Allocate data preparation functions such as sorting, framing, error correction, and bit reversal to a single sement to avoid redundant development.

The requirement allocation process is a large and complex problem for a program the size of SIRTF and must be approached using a disciplined methodology. The methodology must assist in insuring a complete and consistent set of top-level functional and performance requirements and verify a consistent set of inputs, processing, and outputs throughout the system. It should establish a requirements data base that is easy to maintain and modify in a controlled environment. The methodology should then support allocating the requirements among the various SIRTF segments (instruments, spacecraft, POCC, science ground processing, operators, scientific users, etc.). It must retain these allocations in the data base to assist in performing tradeoff analyses (e.g., should this function be performed on-board or on the ground? What is the impact of tightening this top level performance requirement?) and tracking changes and additions. An initial version of this data base would support the generation of the segment functional specifications for the RFPs and communicate to all segments what functions are being performed by other segments as well as their own. This data base should be developed early in the program and maintained by a top level system engineering organization throughout the program life. A more detailed description of such a methodology developed by TRW is presented in Appendix A.

12. RECOMMENDATION: The system-level functional and performance requirements should be specified and allocated to segments using a disciplined methodology and stored in a permanent data base for ease of future change and traceability.

The experience on SOGS (as well as most other interactive systems TRW has developed) has been that the user community could not initially express what they wanted in terms of user interaction and functionality. It is very difficult to visualize how a system will operate from a series of written requirements. The users will "know the right way when they see it" and also "know what they don't want when they see that". This sort of an environment leads to many changes as the system and the users mature, frequently causing design or code modifications. An early effort to understand how the users want to use the system will lead both to a more complete set of functional and operational requirements and a more satisfactory interactive system.

13. RECOMMENDATION: Work with the end users to understand their needs and develop a prototype system prior to formal generation of system requirements to help focus the needs of the users. This should at least prototype the user interface with the system, but might also extend to some of the internal design issues. The results of this analysis should be documented in the operations concept and requirements documents and agreed to prior to the initiation of segment design.

SOGS was developed around a single interactive command language (COMET). This command language has to support proposal entry, planning scheduling, real-time analysis, real-time instrument commanding, archive and product generation, and interactive analysis using standard and user-supplied custom tools. These functions are operated by a very broad range of operators with different skills, different training, and different needs. It is not clear that any single command language can service all of these requirements well.

14. RECOMMENDATION: Trade the cost effectiveness of developing a single command language and training everybody in the use of that single language against the advantages of several command languages, each customized to the particular requirements of one or more subsegments. This trade should be supported with MMI prototyping of the various alternatives.

The following subsections address requirements peculiar to each of the SOGS applications subsystems.

2.2.2 Science Planning and Scheduling System (SPSS)

The requirements for Science Planning and Scheduling have been much harder to define than the other applications subsystems of SOGS. There is no single answer to what SPSS should do or how it should do it; everyone has their own opinion about issues such as: How much is done automatically by computer algorithms? How much interaction is the human allowed? What are the parameters to be optimized in a schedule (maximum on-target time, maximum science data taken, maximum number of different observers allocated time, etc.)? Which instrument parameters should be selected by the proposer? the human scheduler? the automatic algorithms? How much optimization is required relative to the cost in development and operational resource utilization? Questions such as these were not answered early in the SOGS program. In fact, they were not even asked until designs were starting to solidify along the lines of TRW's interpretation of the Functional Specification (ref. 1.4a). When the answers came back different from TRW's assumptions, there was considerable redirection and modification.

15. RECOMMENDATION: Before completing the requirements analysis, the scheduling variables and parameters must be defined. This includes inputs (proposals), the vehicle and instrument options, instrument, spacecraft, and science constraints, and restrictions as well as the level of detail and content of the output of the scheduler. In addition, the operational concept of how scheduling is to be performed must be clearly specified.

The ST Program initially assumed a direct interface between the astronomer's proposal and the scheduling system of SPSS. Recently there has been a realization that this overlooks the differences between what the proposer needs to submit and what the scheduling system requires to perform scheduling. For example, information in the proposal will justify the scientific merit of the observation to support the technical evaluation committee, but SPSS has no need for this data; on the other side, SPSS needs to know how many alignments, exposures, observations, and observation sets (all different levels of abstraction from the actual collection of science data) are required plus an assessment of whether the data gathering can be spread across several orbits, interleaved with other instruments, etc., but the proposer usually does not care about these details. The ST Program is now going to a two step process.

16. RECOMMENDATION: An interactive front end capability should be provided to support proposal entry and selection, plus translation and detailed expansion into the data required for the scheduling process.

Scheduling is very sensitive to the operational concept to be used as well as the instrument and spacecraft features. Block time versus interleaved observations is a prime example of this. Even small changes in the operational concept can mean a different algorithm would be best. New algorithms need to be tested and easily inserted into the system as the operational concept matures and as experience is gained.

17. RECOMMENDATION: The actual scheduling algorithm should be separated from the main line calculation of orbits and constraint and restriction validation. If this is done, new scheduling algorithms can be slid into the system easily and a well-defined interface for externally generated schedules exists.

18. RECOMMENDATION: An early testbed, in which various automatic algorithms are prototyped prior to selection of which to use for SIRTF, would assure the best algorithms would be selected and also help drive telescope operational concepts or actual hardware design.

The divisions of responsibility between PASS, GSSS, and SPSS led to some of the problems associated with the SPSS definition. In several cases, functions were being performed in more than one segment, others in no segment, and still others not in the "best" (most cost-effective or efficient) segment.

Command generation and management is a very different discipline from scheduling. It requires different expertise and is frequently developed separately. The command system must worry about setting the appropriate bits, tracking the current configuration, managing on-board memories and communications, etc. Scheduling deals with orbits and science (bright light avoidance, occultations, etc.) plus instrument timing issues and certain constraints and restrictions. A clear-cut interface between these two functions would avoid redundant software and place the development of a single expertise in a single place. The ST Program has not gone in this direction, both due to the original split between SPSS and PASS and the complexity of the SIs requiring detailed SI knowledge to perform even coarse scheduling.

19. RECOMMENDATION: Command generation and management should be kept separate from scheduling, either as two separate subsystems of a single segment or in separate segments.

20. RECOMMENDATION: The SIs should be designed such that a detailed knowledge of on-board timing, memory, and commanding requirements is not needed to perform scheduling tasks.

Scheduling constraints and restriction validation is another set of requirements requiring system-wide analysis and allocation. Certain violations are more easily detected in the science scheduling, others are only known to the POCC. A careful definition and allocation will avoid overlap and holes in this area. In addition, constraints and restrictions were stated in such a way that they could sometimes not be directly checked or controlled by SPSS. For example, a restriction to not exceed a specified temperature cannot be allocated to SPSS, but one specifying that a heater level not be commanded to a value dependent on duration of instrument use could be.

21. RECOMMENDATION: The spacecraft, instrument, and science contraints and restrictions must be identified, documented, and baselined early and treated as system-level requirements. Each item should be allocated to the segment(s) responsible for validating that it is not violated with an indication of the impact if it is violated. Each item should be stated in a form (or forms) which the designated segment (or segments) can check and control.

ST is one of the first really complex spacecraft to make use of TDRSS operationally. As such, many operational considerations are not yet understood and will not be until some ST experience has been gained. For example, there is no good estimate of what percentage of the requests for links to ST will be accepted and rejected and how long in advance these decisions will be made.

22. RECOMMENDATION: Watch the ST experience carefully in its operational experience in using TDRSS and try to respond to problems encountered there by adjustments in the SIRTF operational concept.

2.2.3 Observation Support System (OSS)

The functions to be provided by OSS were well defined in the Functional Specification (ref. 1.4a). TRW has basically proceeded to build what was originally specified. There was some early discussion over how much analysis was to be allowed at an OSS console: was OSS just to support real-time interaction to adjust the instruments and pointing or was it also to support fast turn-around analysis? and how much analysis was required to support the required real-time decision? Due to the real-time nature of OSS, the allowed interactions and analysis were kept to the minimums originally specified, with any further analysis left for post-processing in PODPS.

23. RECOMMENDATION: Provide only fast, simple tools and algorithms in support of real-time analysis; the more complex and more accurate analyses should be performed off-line. The scope of real-time decisions should be structured and limited so as not to require complex analyses.

24. RECOMMENDATION: An early decision must be made as to whether "joy sticking" (interactive pointing of the telescope from the ground) will be allowed. This drives the capabilities required in OSS, particularly

in the area of commanding. (It also impacts the decision of block time versus interleaved observations in SPSS and perhaps even the orbit and ground communication techniques.)

On the ST Program, SPSS preplans the bulk of the SI configuration changes. This requires that current configuration information always be available to SPSS. Two real-time events make this knowledge difficult. These are the OSS capabilities to select from a set of preplanned alternatives (branching) and to change certain instrument parameters and configurations (such as the real-time optical filter selection). The ST Program has avoided many interface and scheduling problems by making the following a requirement:

25. RECOMMENDATION: SPSS must assure that all branches from a decision point come back to a single point with the SI in a single configuration no matter which path was taken. Either OSS or SPSS must also assure that the SI is returned to a known state if a real-time configuration change has been commanded or allowed during a session.

In some cases, the information required to make an intelligent real-time decision is not easily available to OSS. For example, some current instrument configurations cannot be determined directly from the engineering or science data streams. One of the instruments does not even report absolute optical filter positions, just deltas from the last position. To know the current filter position thus requires that a full model of current instrument configurations be maintained by OSS at all times, whether commanded by OSS, preplanned by SPSS, or commanded directly from the POCC. Such a model was finally determined to be too expensive to implement in OSS, so this information is not available to the real-time decision maker.

26. RECOMMENDATION: An early study should determine all the information required by the real-time analyst to make real-time decisions, and that data should be directly available in the engineering or science data streams. This study should be a driver for the spacecraft, instrument, and ground segment design.

There was a significant emphasis placed on OSS determining, automatically, when its commands were not correctly carried out by the spacecraft and when expected data was not downlinked (data accountability). Neither of these checks is supported by any real-time corrective action from OSS, since all activities are preplanned in SPSS. Command execution validation is really a command management and health function more properly allocated to the POCC; data accountability is just as effective in the off-line PODPS system since no real-time recovery is possible.

27. RECOMMENDATION: Allocate only those functions requiring real-time response from the science observer to the scientific real-time workstation (OSS). Command and health functions should be allocated to the POCC, data integrity can be deferred to the post-processing phase.

28. RECOMMENDATION: A trade should be performed to decide if real-time error recovery is needed. This will drive the scheduling decision of block time versus interleaved observations as well as what functions OSS must supply and what data OSS must be provided.

The engineering data supplied by ST is received and decommutated by the PORTS. A subset of the data is then tagged to indicate what parameter it represents and shipped to OSS. This greatly increases the data volume and bandwidth between PORTS and SOGS (by a factor of approximately 6-to-1). It also removes certain time information associated with the position of the raw data in the telemetry stream. The concept was to save SOGS from having to repeat the decommutation task that PORTS had to do for its health and safety analysis as well as supply only a required subset of the information to SOGS.

29. RECOMMENDATION: Perform an early analysis of the engineering data required by the science ground system (it may well be all of it). Trade the savings in redundant processing against the increased bandwidth requirements and the loss of (or complexity of retaining the) positional information.

2.2.4 Post-Observation Data Processing System (PODPS)

One of the major purposes of PODPS is to produce accurately calibrated files of science data. For certain modes of certain instruments, this requires knowing the time, spacecraft position, and/or spacecraft pointing at the time the data was taken. ST does not supply this information with the data; only predicted time, position, and pointing are available. This leads to less accurate corrections.

30. RECOMMENDATION: Time, position, and pointing information should be supplied with the science data or in an easily correlatable engineering data stream.

There has been a significant amount of discussion over what types of products (tapes, plots, film, and text) should be produced by PODPS under what circumstances. The lowest development cost approach is to define a single product for each mode of each instrument that is always produced. The more flexible approach is to produce all products selected from a predetermined list at proposal time. The first approach insures a consistent product archive, the archive researcher then knows for all instrument configurations what to expect in the files; the second approach is better for producing the customized products the original observer requires for research and analysis.

31. RECOMMENDATION: Both a default product for the archives and customized products requested in the proposal should be automatically produced for each data set received by PODPS.

There are many different calibration algorithms that can be applied to correct a given data set. Some will maintain geometric fidelity while sacrificing radiometric, some sacrifice geometric while maintaining radiometric, still others will comprise between several "best" corrections. Depending on what the observer needs for the particular experiment, the data correction requirements may be very different. Yet the data archives should contain a consistent set of data, all calibrated in some nominal way so researchers can compare data from different observations.

32. RECOMMENDATION: Default data correction algorithms should be used for the data to be placed in the archives based on instrument and configuration; the algorithm may not be "best" for any particular data feature, but must not destroy significant amounts of information. In addition, proposers should be able to select from a predetermined list of algorithms at proposal submission the algorithm they wish applied for their particular products. These same algorithms should also be available interactively for archive researchers who need other than the default corrections.

2.2.5 Project Data Base

The ST Project Data Base (PDB) was designed to be a centrally located data base of shared data. Each segment was tasked to specify files of data required in the PDB and the Mission Operations Contractor (MOC) was to populate it. Unfortunately, with so many segments involved, the files specified were not consistent as to format or content and frequently overlapped. Definitions were coordinated too late to support timely population. To test SOGS, TRW had to populate all portions of the data base we required, even those items and files specified by other contractors; presumably other segments are doing the same, causing a significant amount of redundant effort. A central project data base is an excellent idea, but it must be controlled by a single organization who takes responsibility for it, insures that only data appropriate to such a data base is allowed in it (e.g., not screen formats for the POCC terminals as is the case in the ST PDB), and assumes responsibility for populating the data base in support of the various segments' development schedules.

33. RECOMMENDATION: A central project data base should be developed, containing data items shared between segments and data needed by one segment but only available from another. Data peculiar to an individual segment should not reside in this data base. A single agent should be responsible for developing and populating this data base, with support from all segments.

There was a tendency on the ST Program to defer hard problems and problems no one wanted to address at the time to the PDB without consideration of whether the PDB was the solution or not, and then treating the problem as solved. For example, no one knew how to define what a "command set" was, how fine a function such a set would initiate, etc., so it was relegated to the PDB. But early knowledge of the level and content of "command sets" would have helped the SPSS design effort.

34. RECOMMENDATION: When an item is assigned to the PDB, it should immediately be defined and added to the data base. This timely response will assure the feasibility of the technique and force early identification and resolution of problems.

2.2.6 External Interfaces

SOGS has five external hardware interfaces (PASS, PORTS, GSSS, DCF, and NASCOM) and one external software interface (to the SDAS software), plus the project data base interface described in the previous section. In addition, it interfaces with operators, astronomer-users, proposals, calibration reference files, products, and (indirectly) with the science instruments. Only the five hardware interfaces and the software interface have true baselined Interface Control Documents (ICDs). Most of the others have been documented in memos, unbaselined operations concepts documents, or not documented at all.

35. RECOMMENDATION: All external interfaces should be documented in formal baselined ICDs by the completion of preliminary design. This includes interfaces with proposals, operators, users, calibration reference files, products, and instruments as well as hardware and software interfaces with other segments. Preliminary design should not be considered complete until these ICDs are in place.

The interfaces that were documented did not converse to agreed-to ICDs in a timely manner. This has been primarily caused by each contractor protecting the scope of his contract by refusing to accept interfaces requiring increased processing and trying to force the work onto the other side of the interface. For example, SOGS was requested (and finally agreed) to handle the entire proposal data base, including all data required by PASS and GSSS, even though that data is not needed by SOGS. An additional factor was agreements being made by two interfacing segments without consideration of the impact on a third segment. For example, an agreement between SOGS and GSSS could well impact PASS.

36. RECOMMENDATION: An interface working group should be formed and meet regularly to discuss all interfaces. This group must have the authority, incentive, and resources to make decisions and direct contractors to add work as required. All segments should be represented on this working group, from the spacecraft and instruments down to the various ground segments and the users.

Assuming the interfaces are finally well defined and all segments are implementing to baselined ICDs, there is still a significant integration risk due to misinterpretations and holes in the ICDs. Risk could be significantly reduced if segments could somehow test their interfaces early in development.

37. RECOMMENDATION: Each external ICD specifying a data interface should require a redundant tape interface for use both in testing and (where possible) operational failure modes. Each segment should be required to deliver output tapes to their interfacing segments periodically and conduct acceptance tests using such tapes as input. These deliveries could initially be generated by (hardware or software) simulators with later deliveries being produced by maturing versions of the deliverable segment. The recommendation covers not only interfaces between ground segments, but also the interfaces between the spacecraft (and instruments) and the ground. Conflicts between expectations and actual tape contents should be addressed by a central working group with the authority, incentive, and resources to make decisions and direct contractors to change their hardware or software designs as required.

2.3 Science Instruments

The five (plus FGS) ST Science Instruments were well into design prior to the award of the ground segments of the ST Program. They were each developed independently, with only very general guidelines to insure consistency of design or implementation. As a result, while each instrument is designed well and meets its scientific objectives, the integrated package of instruments and spacecraft does not fit well (for example, instructions on some instruments perform such minute functions that a significant function requires many instructions; it is estimated that the on-board memory may only hold an average of two orbits worth of commands and in some cases it cannot hold the entire instruction set required to execute a single observation).

38. RECOMMENDATION: Science Instrument design standards and guidelines should be established early in the SI development contracts, agreed to by the SI development segments, and enforced by a central agent.

39. RECOMMENDATION: A system-wide design coordination and review function must be provided to take the responsibility of assuring that the instruments will work as an integrated package with the spacecraft and the ground segments.

The ST instrument data streams are not consistent in the format, content, or encoding of the downlinked science or engineering data. Some examples: most instruments send down absolute optical filter positions, one sends relative deltas from the last position; some parameters on some instruments are gray coded, most are simply binary values; the same types of parameters are sent down from different instruments in different positions, with different numbers of bits of significance and at different time intervals. While SOGS can handle this situation by processing each instrument separately, significant cost savings could be gained by having as much standard as possible, thus allowing shared code and reduced learning time in SOGS development.

40. RECOMMENDATION: The design of the instruments should be as consistent as possible. A single design review group must take responsibility for reviewing the different instrument designs, detecting the inconsistencies, analyzing the reasons for the inconsistencies, and directing design changes to force consistency if there is no driving reasons for the difference.

One major problem on SOGS has been the lack of accurate, timely, and consistent information concerning the science instruments, their operation, command and control, constraints and restrictions, and downlink data format, content, and interpretation. Until recently, there has been no central responsibility to document this sort of instrument information and keep it up to date; TRW had to glean it from (frequently out of date) documents, memos, flight software PDL, dumps of test data tapes recorded during VAP, or by IDT simulations, plus large numbers of meetings, phone calls, and correspondence with both the instrument developers and the flight software contractor.

It must be recognized that the instrument developers are not the source for all of this information; they understand the data as generated by their instrument, but they are not responsible for the on-board data formatting into packets or the insertion of non-SI peculiar data into the data streams (e.g., packet format codes, time, header information, etc.). All of this information needs to be combined into one source document per instrument.

The spacecraft, instrument, and science constraints and restrictions also need to be established early in the contract and documented. Consideration must not only be given to health and safety violations, but also to scientific value (i.e., you won't hurt the instrument, but also won't get much useful science with a particular command sequence).

41. RECOMMENDATION: A single, definitive document should exist for each instrument. This document should contain all information required to safely command, control, and interpret the downlink from that SI. These documents and the instruments they describe should be put under centralized configuration control early in the ground segment design period. The document format should be standardized for each instrument to insure consistent content and ease of use. The outline presented in Table 2.3-I is an augmentation of the SE-Ø1 document outline currently being used for the ST SIs.

TABLE 2.3-I. SCIENCE INSTRUMENT NOTEBOOK OUTLINE

A. Introduction (short) (probably plagerized from other documents), including a high level overview of instrument utilization and scientific objectives.

B. Instrument description augmented with diagrams, etc., including the scientific rationale for the various modes.

C. Detailed functional descriptions at the subsystem level. Functional block diagrams which include all telemetry and command points should be included.

1. Optics

   a. Design of optics, rationale behind design
   b. Possible light path configurations
   c. Summary of focus design (e.g. depth of focus, range of mechanism, etc.)
   d. Measured/computed transmissions, reflectivities, dispersions and other parameters.
   e. Particular difficulties, sensitivities (like alignments) and their effects on SI.
   f. Internal baffling and stray light control.

2. Mechanical

   a. Summary of structural design and rationale
   b. Summary of structural modeling
   c. Test results, if any
   d. Mechanisms (for each mechanism)

      (1) Design description and rationale
      (2) Drawings
      (3) Operation--how do they work, in electr-mechanical terms?
      (4) How long do they take to operate?
      (5) How are they commanded?
      (6) How are they kept track of? When are they out of spec?
      (7) Particular difficulties, idiosyncracies, anomalies, etc. and their effects
      (8) Potential failures and failsafe mechanisms

3. Thermal Design

   a. Summary of SI thermal design and rationale
   b. Possible thermal configuration states and their validity
   c. Summary of thermal modeling
   d. Summary of test results, if any

TABLE 2.3-I (continued)

e. Map of temperature monitor and heater locations
f. Specific thermally sensitive sub-elements (e.g. CCDs, TECs etc.) describe each in detail

   (1) Why they are sensitive?
   (2) How they are controlled?
   (3) How are they monitored? What are appropriate science and health limits?
   (4) Impact of lack/loss of control
   (5) Back-up procedures for returning to useful/safe thermal state
   (6) What are the relevant time scales?

4. Power

   a. Detailed description of power distribution system (with drawings indicating command and telemetry points in various SI states
   b. Power consumption of various boxes
   c. Allowable/expected power configurations
   d. Rules/constraints for switching power
   e. Provisions redundancy, methods for cross-switching
   f. Idiosyncracies found in test (if any)

5. Detectors

   a. Summary of how they work, including a reasonable description of the physics involved.
   b. Detailed drawings
   c. Description of processing done on signals, data, etc. near to the detector
   d. Detailed description of detector related electronics (amplifiers, HVPS, etc)

      (1) drawings
      (2) how they work
      (3) how we control and monitor them

   e. Sensitivities, calibration parameters, etc. (test results or analysis)
   f. Descriptions of settings to be made (thresholds, gains, etc.)

      (1) how to determine proper values
      (2) how to get proper values to SI
      (3) how to monitor these values

   g. Descriptions of techniques recommended for use in trend analysis to verify performance

TABLE 2.3-I (continued)

(1) using science data
(2) using engineering data

h. Typical calibrations of detectors

(1) specification of science requirements (e.g. S/N required, range of exposures required, etc.)
(2) typical procedures used for calibrations
(3) analysis required for calibration

i. Important things to know about not covered above

6. Commanding the SI

a. Description of nominal operating philosophy

(1) What gets done where, etc.

b. Commanding philosophy (direct)

(1) list of commands
(2) definition of what each one does including critical commands, pre-requisite commands, time criticality, and execution times
(3) how are the results reflected in TM (direct verification or observed results)

c. Commanding Philosophy (indirect)

(1) use of NSSC-I or DF224 S/W, Macro commands, etc.
(2) list of commands
(3) definition of what each one does including critical commands, pre-requisite commands, time criticality, and execution times
(4) how are the results reflected in TM (direct verification or observed results)

7. Data and Telemetry Processing - including non-instrument on-board processing (e.g. packetizing the data).

a. Overview of data processing, including on-board processing
b. Formats available for engineering and science data
c. Detailed description of data formats including any data encoding used
d. What are the intended uses of the different formats

TABLE 2.3-I (continued)

e. Programability of formats (if any)
f. Timing of data in telemetry relative to S/C clock

8. Microprocessors/NSSC-I S/W
(Note that this section should discuss both the microprocessors and NSSC-I S/W for those SIs which use both. In particular, there should be a clear discussion of the architecture and intent of design when both are used for an SI, so that it will be clear what is done where.)

a. Summary of the use of microprocessor
b. Diagrams, data flows, memory organization, command buffering, redundancy, etc.
c. Description of flight code and what it does
d. Listing of flight code (or reference doc.)
e. Table of input parameters and sources for each program, task, or subroutine
f. Table of output parameters and sources for each program, task, or subroutine
g. Detailed description of input/output formats--which bit is in which word in which whatever
h. What parameters are in a flight data base adjustable from the ground?
i. Operations

(1) How do we know its working properly operationally
(2) Diagnostics, self checks, etc.
(3) Fixing (e.g. patches to code using RAMs)

9. Etc. (Other systems not covered above) (e.g. calibration or flat field lamps)

D. Observation Planning

1. Description of determination of observing time (or sensitivity) (Note this section is to provide material previously expected in OP-Ø4 Section I)

a. Various sensitivities, throughputs, efficiencies needed for computations (may ref. curves in previous parts of doc.)
b. Available S/W for doing computations (if any)
c. Specific sensitivity models

(1) Target acquisition
(2) Science observations
(3) Internal calibrations
(4) External calibrations

TABLE 2.3-I (continued)

2. Description and command sequences for routine SI mode transitions

   a. Hold to Operate
   b. Operate to Hold
   c. Other standard modes
   d. Any internal mode changes (e.g. WF to PC) (include command sequences, data to take, ground analysis or interpretation, recommended special observations as part of sequence, etc.

3. Description and Command Sequences recommended for routine orbital use

   a. routine observations
   b. mode I, II target acquisitions
   c. SI parameter adjustment
   d. internal or external calibrations
   e. etc.

4. Observational good practices
5. Planned use of SSM features (e.g. mechanism motion or take data flags)

E. Calibration and Maintenance philosophy

1. Summary of Calibration/maintenance needs of instrument.
2. To n. for each type of calibration activity.

   a. Description of need for calibration and means for obtaining it.
   b. Expected frequency of occurrence.
   c. Use of internal or external calibrators.
   d. Observation sequence, including command sequences if not the same as normal observations.
   e. Analysis required.

F. Real-time SI monitoring

1. Activities which require R/T monitoring and possible intervention
2. Routine monitoring with engineering data

   a. High priority parameters to monitor, health and safety limits
   b. Parameters to be monitored for major reconfigurations (how can configuration be determined from telemetry?)
   c. Routine processing of engineering data (more than merely display); algorithms; science limits.
   d. Telemetry display pages (used for test/VAP).

TABLE 2.3-I (continued)

3. Routine monitoring with science data

   a. What would be "good practice" things for us to routinely do, in R/T, with science data to verify SI science health?

4. Known failure/degradation modes and how they affect science (from test experience)

G. Contingencies

1. Safing procedures
2. Possible/likely problems, expecially those which would require a quick reaction (e.g. frozen WFPC heat pipes)

H. Ground Processing Algorithms for target acquisition, quality assessment, and data calibration.

1. To n. (for each algorithm)

   a. Inputs (in science and engineering streams, ground data bases, and from operators and proposal)

   b. Outputs - specify bits of significance and any special formats

   c. Equations and sequence of execution

   d. Accuracies of calculations

   e. Error detection/recovery.

I. Operational constraints and restrictions for health, safety, and scientific quality. Define each item, the recommended model (if any), and the impact of violation.

42. RECOMMENDATION: Standardized terms, parameter naming conventions, bit numbering conventions, etc. should be enforced in all instrument and data-related documents. A single agent should be responsible for establishing these conventions and enforcing them across all documents.

43. RECOMMENDATION: One way to assure consistency of documentation would be to have a single contractor responsible for collecting all required information from the many sources for each SI and generating and maintaining the documents.

The ST Program baseline required that all information needed for science data processing be placed in the science stream, even if redundant with data in the engineering stream. This leads to easier processing, eliminating the need to correlate data from two separate streams. Since the instruments were designed prior to the algorithms, this rule limited the algorithms to the imagination of the original instrument designers. If an input was not in the science stream, the algorithm could not be implemented.

44. RECOMMENDATION: All observation-peculiar data required for science data processing should be in a single data stream. This requires that the science data processing algorithms be developed at least to the level of identifying the inputs prior to completion of instrument design. This data should include information specifying data framing, time, unique identifier for the frame, spacecraft pointing, and instrument-peculiar parameters.

Most of the modes of the ST instruments performed on-board data integration, downlinking the data only after the exposure was complete. This reduced the on-board tape recorder requirements, the downlink requirements, and the ground processing resource requirements (since no ground integration was required). Current SIRTF plans are to directly downlink all data and perform data integration in the ground segment.

45. RECOMMENDATION: The savings in instrument simplicity must be traded against the increased costs in spacecraft and ground resources in deciding whether to perform data integration on-board or on the ground.

Each ST science instrument developed its own command philosophy. Some are a single command word with different bits indicating the desired configuration, others have separate commands for each configuration change, others have high level macro commands that are expanded on-board into detailed instrument commands, others require table loads to perform reconfigurations, one has an exposure meter mode where there is no way to predict when the exposure will be over or how many exposures will occur in a fixed period of time (making very difficult scheduling and data accountability problems). All of this made scheduling and command generation a much more difficult task than they need have been.

46. RECOMMENDATION: The philosophy of the instrument commanding should be consistent. A single design review group must take responsibility for setting standards and reviewing the different instrument command philosophies, detecting the inconsistencies and standards violations, analyzing the reason for the discrepancies, and directing design changes to force consistency if there is no driving reason for the difference. Remaining inconsistencies should be well documented.

2.4 Contract Phasing

The ST segments were procured in a logical order based on development time and need date for each portion of the system. The long lead elements (such as the SIs and spacecraft) were started early, the shorter lead elements (such as SOGS) were started several years later. The advantage to this schedule is that contractors are not done earlier than their segment is needed or earlier than it can be integrated into the rest of the system, so they need not be carried during nonproductive periods. The disadvantage is that some segments are not represented in early decisions which impact them. For example, the science data formats are needlessly complex and inconsistent from a ground processing standpoint. Front end analysis by the ground segment contractor to support trade studies of on-board versus ground processing complexity at the SI design stages might have reduced the difficulty of the ground data reformatting without increasing the complexity of the SIs themselves.

47. RECOMMENDATION: Early in the SI design, the ground segment contracts should be started at a low level to validate the consistency, feasibility, and simplicity (from a ground processing point of view) of the SI designs. This could be done through an early award or during a competitive Phase B.

48. RECOMMENDATION: Early in the ground segment contracts, before requirements are baselined, operational flows and operations concepts should be developed and documented. This should incorporate the results of early prototyping of the user interfaces.

49. RECOMMENDATION: The final, responsible end user community should be represented from the very start of the program (in the ST Program, this would have been the ST ScI). In this way they will be involved in assuring the program meets its scientific objectives, they will feel ownership of the functional specifications levied on the various segments, and will understand the constraints under which the program is operating.

3. TASK 2 - SOGS TRANSFERABILITY

3.1 Introduction

3.1.1 Purpose

The purpose of section 3 is to document the findings of Task 2 of the SIRTF Study SOW (ref. 1.4 d). Efforts carried out under this task were:

a. A review of the SOGS design and development with the intent of identifying and assessing the feasibility of reusing SOGS software for SIRTF.

b. Document these findings including identification of the underlying assumptions and conditions upon which the transportability assessment is based.

3.1.2 Scope

This analysis is limited to TRW experience on the ST SOGS Project. The intent of this section is to analyze the SOGS software and determine the degree of transportability to the SIRTF program. The determinations were made separately for the transportability of the conceptual design and the FORTRAN code. Analysis of tranferability is limited in many instances by the preliminary nature of the SIRTF operations concept.

SOGS, like all large systems, is characterized by complex dependencies which exist between the operational concepts and components of hardware and software. This section will also identify those fundamental factors on which the degree of transportability is based.

Judgements on the degree of SOGS transportability at this stage of SIRTF are necessarily approximate. In an attempt to quantize the ease of conversion, ratings are defined as follows:

High: Easily transportable; 75% or more of the design or source code can be used with minor modification.

Medium: 50% of the design or code can be used with minor modifications. Major changes and/or new development for the balance.

Low: Only 25% or less can be converted with minor modification.

3.1.3 Report Organization

Discussion of the transferability of SOGS is divided into three sections:

1. Operations Concepts: Discussion of the similarities and differences between the SIRTF preliminary operations concept and SOGS.

2. Software: Conceptual Design and code transportability of system support and applications software.

3. Hardware: VAX network, Data Base Machines, user workstations and communication equiment.

3.1.4 Background

Discussion in this section deals with SOGS software at a functional level. It assumes an understanding of the system consistent with the overview presented in section 1.3 of this document. Terms and concepts are defined in that section which are expanded upon here such that they reflect on transportability to SIRTF.

3.2 Operational Concept

3.2.1 Introduction

This section will address the impact of the SIRTF operational concept on SOGS transportability. The operational concept (described in ref. 1.4g - SIRTF Observational Operations Concept) is still in the preliminary stages, but, generally is similar to that of ST SOGS. This section will, to the level of the SIRTF documentation, identify the similarities and differences and describe their influence, if any, on software transportability. SOGS performs science operations only and, as such, pertains only to a portion of the SIRTF operations concept. For example, SOGS is not designed to actually implement any events on the ST vehicle, but only schedules science activity requests to the POCC for command processing and transmission. Mission operations issues are non-SOGS and are not addressed here except as they touch on SOGS.

As stated, the SIRTF operations concept is not mature or complete. In some cases, issues with significant impact on the SOGS software are identified for consideration or further definition. These are contained in section 3.2.3.

3.2.2 Operational Concept Analysis

3.2.2.1 Similarities

The similarities of the SIRTF concept of operations to that of ST SOGS are general in nature and are listed here along with the corresponding SOGS capabilities in that area (where appropriate). Both spacecraft are free flying, long life vehicles delivered into orbit and revisited by the STS.

The communication paths of the systems are nearly identical. Both will use the TDRSS and NASCOM networks for two-way communication to the spacecraft. The NASA Data Capture Facility will collect the high rate science data on the ground.

Basic Science Planning and Scheduling functions planned for SIRTF operation exist in ST SOGS. SOGS supports an interactive system for science proposal entry and modification. It also provides routine and special planning information such as future target acquisition dates, avoidance angles, occultations and observation candidate and calendar lists. SOGS has Guide Star Selection System interface software to request (and receive) guide star data from the ST Science Institute catalog. An automated algorithm produces a time ordered sequence of observations which can be interactively modified. This schedule can then be processed to produce a complete set of command requests to be sent to the POCC for transmission to the spacecraft. These scheduling algorithms support interleaving of observations to maximize use of the Science Instruments.

SIRTF plans for real-time monitoring and control of operations are compatible with ST. Both are designed for primarily non real-time operations. SOGS supports a limited capability to display health and status engineering data for informational purposes only. A SOGS operator can request limited real-time commanding in so much that it affects on-going science activity. A "quicklook" capability on science data is available to evaluate observation effectiveness in both real time and tape playback modes.

Science data evaluation functions provided by SOGS operate in a manner similar to the SIRTF concept. In fact, the post observation processing capabilities of SOGS software exceed the current SIRTF specifications by a large degree. These are discussed in a later section. SOGS receives large volumes of science data from the DCF at scheduled intervals and provides several automatic processing functions. Among these are evaluation, calibration, archiving and producing output products for observers to analyze or take away. Data storage products include both engineering and Science Instrument measurements in raw and calibrated form.

3.2.2.2 <u>Differences</u>

The SOGS capabilities described above represent, at a very general level, the basis for which SOGS software could be useful for SIRTF. The extensive capabilities that SOGS could transfer to SIRTF are described briefly in section 3.3 but are at a level of detail not yet developed for SIRTF. The differences in the concept of operations that can be determined at this stage are more specific in nature than the similarities.

In particular, SIRTF has only three science instruments (ST has five), a polar, sun synchronous orbit (currently baselined although a low inclination, ST-type orbit is under study by NASA) and will operate a single Science Instrument at a time (ST permits simultaneous operation). These three factors will reduce the complexity and size required of the science planning and scheduling function in comparison to SOGS. SOGS scheduling software also makes no attempt to simulate the ST spacecraft operations as is planned for SIRTF.

Space Telescope has the ability to integrate the observed science data on board the spacecraft before sending it down to the ground. This provides a significant reduction in down link data volume. This feature is currently not planned for SIRTF and, couled with the potentially higher data rates inherent in infrared instruments, poses significant questions. See issue number 3 in section 3.2.3.

The ST operations center is colocated at Goddard, simplifying the real-time telemetry link from NASCOM to the POCC. The interface from the DCF to SOGS is more complicated. This interface was required to use the standard X.25 network communication protocol which is officially defined only to 9600 baud. Such a rate was impractical, so special equipment was developed to implement this interface. The resulting system uses installed ground circuits to transmit the packet formatted data at 1.544 mbps. With SIRTF, the low rate telemetry data can be transmitted from the GSFC to the west coast via DOMSAT. The X.25 protocol for the DCF interface will be too slow to use via satellite because it involves a system of fixed length data packets, checksum processing and intercomputer handshaking after a certain number of packets are received. Changes to these interfaces reduces the transportability of the data receipt software in SOGS.

The SIRTF operations concept calls for navigation and spacecraft pointing data to be used in science data evaluation and analysis. Currently no real navigation or pointing data is available in SOGS except updates at two day intervals to predicted orbit and position-in-orbit data for scheduling. The ST FGS produces pointing information, but only limited capability to analyze these data exists in the POCC, and its use in science processing and evaluation is unclear. See issue number 4 of section 3.2.3.

3.2.3 Operational Concept Issues

This section is intended to contain a description of unresolved or undefined concepts which have significant impact on the transportability of SOGS software. It represents a dynamic list of issues which merit further consideration and definition by NASA and hopefully will help focus attention on important areas in the ongoing development of SIRTF.

1. Clear definition of the use of SIRTF by end users, the scientific community, is needed. The NASA policy regarding this area has large cost implications for reusing SOGS or developing anew. Functions and equipment described below are named by the current SOGS equivalent and represent a starting point for evaluation of this area.

   a. Amount of on-site post-observation processing

      How much routine/automatic data processing, editing, calibrating, archiving and product generation capability will be provided by SIRTF? Will SIRTF provide equipment and software support for on-site analysis (SDAS)? Will these users be given use of the automatic analysis software (PODPS) as in SOGS?

   b. Facilities

      What equipment and facilities will be provided to support generation of publication quality products (e.g. image plotter, film writer, etc.)?

   c. Scientific Data Products

      Will SIRTF generate customized products or a range of standard format items (e.g. general observer tapes via Flexible Image Transportation System format)? Will these products be deliverable (i.e. take-away)?

   d. Stored Data Retrival

      Should archived data be on-line or stored on tape? Is digital stroage envisioned? Currently, GSFC is studying an ST Data Archival and Distribution System (ST/DADS) which might one day service other spacecraft. Interest by the SIRTF Project might result in a future interface.

2. Use of the ST Guide Star Selection System with SIRTF will make several thousand lines of code in SPSS transportable. However, SIRTF use of the GSSS catalog will necessitate changes to the ST Science Institute developed software that responds to requests for guide stars. Either space in a SIRTF computer (VAX) or an additional VAX will be needed to run the GSSS processing. Also, SIRTF will likely require different selection criteria and algorithms to choose the proper guide star pairs for the SIRTF instruments and FGS.

3. Some general thoughts on data rates: Careful consideration must be given to SIRTF data volume and the amount of data sent through the communications paths and ground processing. Infra-red instruments may generate larger amounts of science data than the ST instruments. Is spacecraft on-board integration of science data to be implemented on SIRTF as in ST? If not, availability of a limited resource such as the high rate TDRSS lines are an issue. Secondly, an increase of data volume and processing on the order of even 2 to 1 or 3 to 1 might overwhelm the SOGS capacity. For example, OSS currently has no real-time data buffering capability to handle surges in ST data volume. Also, SOGS is designed to process approximately 3 x 10 bits or less per day, with an occasional day near 10 bits (with several days of low volume to work this off). These numbers might be low when compared to SIRTF operation.

   TDRSS can have significant Bit Error Rates (BER), so ST uses 3:1 convolutional encoding to yield BER $\leq$ 2 in 10 . This means ST uses the full 3 Mbps line to achieve 1.024 Mbps data rate. Further, ST uses Reed-Solomon encoders onboard the spacecraft to (optionally) encode science data. This encoding is used in the DCF to detect and correct BER $\leq$ 1 in 10 . This imposes about 7% overhead on the data stream, resulting in a maximum rate on SSA of .93 x 1.024 Mbps.

   ST uses a transponder for MA and a separate transmitter for SSA. The planned MA 10 Kbps rate uplink for SIRTF implies signficant electrical power and/or a sufficiently narrow beam. ST uplink rates are 100 bps on MA and 1000 bps on SSA through one of two available omni antennae. At ST data rates, the antennae require a large volume of commands (generated in the POCC) to maintain pointing to TDRSS within the beam width. Beam width can be increased, but at the expense of spacecraft power. ST uses a standard transmitter to achieve the 1.024 Mbps rate on SSA, but the unit overheats in 20 minutes (approximately) and hence is likely not the best solution for SIRTF.

   Assumptions on TDRSS regular availability every SIRTF orbit revolution might be overly optimistic. It implies that SIRTF is operating like a survey type spacecraft and TDRSS is not suited for that, especially on SSA.

4. Besides the lack of pointing data available in post observation analysis, several other factors affect the complexity and usefulness of the science processing on the ground. What data is required and how is it provided to the science analysis and evaluation software? For example, time tags and data necessary

to calibrate a Science Instrument for each observation might be included in the science data stream. Should a method to easily correlate the science and engineering data be provided? Combining the two data streams or providing time tags in each are possibilities. Position-in-orbit data is not available in SOGS for weeks after an observation. Is there an alternative? See recommendation 29 in section 2.0.

3.3 Software

SOGS software is organized into four functional components: The Science Planning and Scheduling System (SPSS), Observation Support System (OSS), Post Observation Data Processing System (PODPS) and System Support. The functions of each of these components will be described and analyzed for transportability to SIRTF.

Transportability of this software is dependent on SIRTF providing a compatible environment. The aspects of this environment which must be similar are:

1. Basic hardware architecture: distributed network of VAX computers

2. VMS operating system, FORTRAN programming language

3. Centralized IDM 500 Data Base Management System

4. Work stations: intelligent image and graphic stations with localized keypad control. Standard alphanumeric terminals with menu and command line processing

5. Support software interfaces: network, man-machine interface, error procesing

6. Evolution of the operations concept

Details are included in later sections, but use of these components or compatible upgrades is implicit in the following assessments of software transportability.

3.3.1 Science Planning and Scheduling System

SPSS includes all the functions which support the preparation of complete specifications of ST science operations. 59,000 lines of executable FORTRAN code encompass a series of interactive processes to transform astronomy proposals into viable ST instrument activities. The four functional areas of SPSS and their associated lines of code (LOC) are shown in figure 3.3-1.

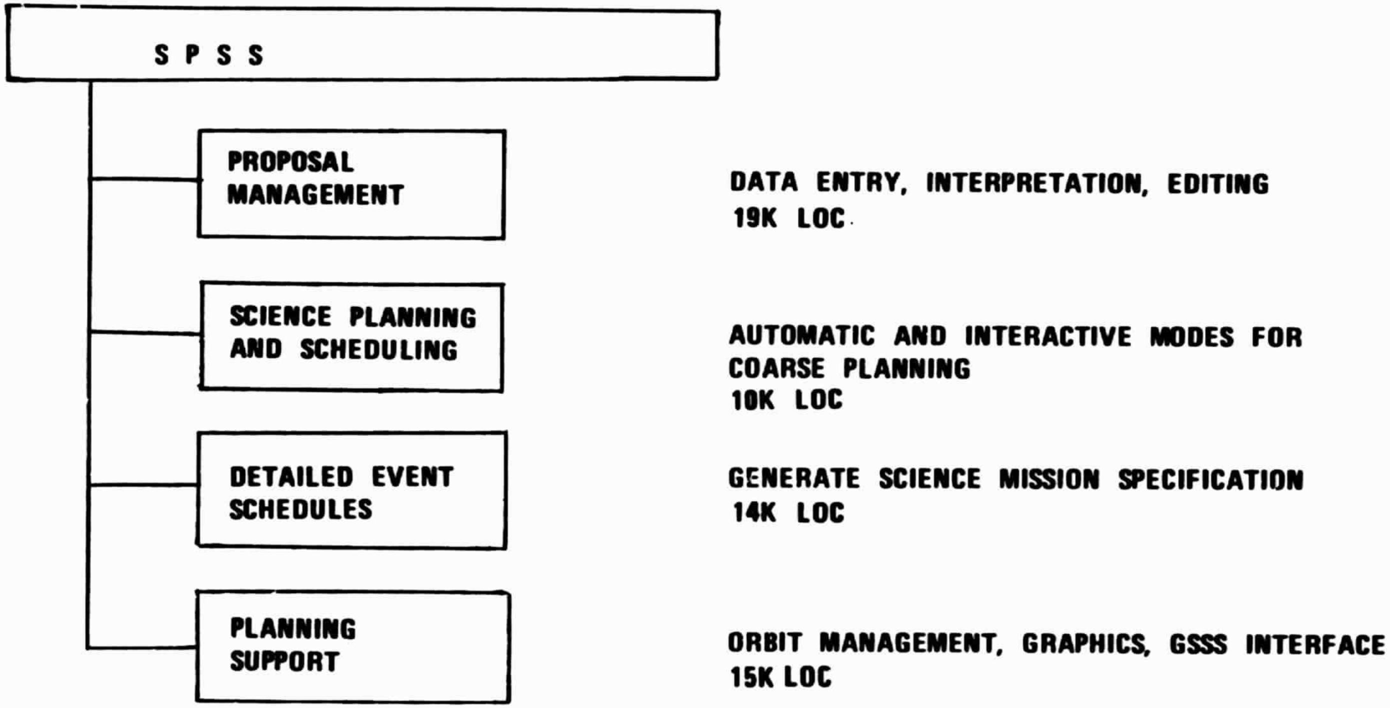


FIGURE 3.3-1 SPSS FUNCTIONS

The basic software structure is centered around the functions listed above. All of these are germane to SIRTF and are consistent with the preliminary SIRTF operational concept. **Basic SPSS software design transportability is high.**

Design of the science proposal entry, modification and verification software is suitable for SIRTF if the proposals are similar in content. An extensive set of detailed templates are used for interactive input. Seventy templates are used to input the 2,000 fields (on average) to completely define a proposal. SPSS software performs range and consistency checks on all input values. In addition, report generation software allows investigative queries and reports on proposals entered into the data base.

The source code elements, however, are specific to the ST instrument specifications and, while similar, would need modification. Although the detailed design is convertable to SIRTF, the source code remains dependent on ST instrumentation. **Proposal management design transportability is medium, source code transportability is low.**

Basic planning and scheduling tools also are easily convertable to SIRTF (design and code). SPSS provides both an automatic and interactive capability for scheduling science activities on an ST calendar time line. In the automatic mode, SPSS uses a "greedy" algorithm technique to schedule candidate experiments. In an iterative process, the algorithm attempts to schedule the highest priority candidates in each potentially acceptable time slot. It evaluates the worth of each by computing a score which reflects a combination of scientific merit (priority, time criticality, etc.) and efficiency (minimizing wasted time, etc.). The algorithm is greedy because it selects the candidates with the highest score for scheduling. Exact scheduling times are not fixed initially, but are assigned to time windows. Candidates are adjusted within these windows as other candidates are added or considered for inclusion in the schedule. In the interactive mode, SPSS software allows for interactive adding and deleting of proposal science activites in the time line.

It should be pointed out that reuse of the SOGS design allows schedule optimization, concurrent operation of multiple science instruments and observation interleaving (as opposed to dedicated block time). **Transportability of the scheduling function design and much of the source code is high.**

Generation of the detailed science mission specification (SMS) involves producing a complete set of commands to define science instrument activity. This function is very specific to the ST instrumentation and command format and its usefulness to SIRTF is dependent on an interface similar to the one defined between SOGS and PASS. Limitation and constraint checking design is modular and transportable to SIRTF. Replacement with source code modules unique to SIRTF would be straightforward. Transportability of the SMS function design is medium and source code is low.

The planning support function includes two major blocks of software transportable to SIRTF. The interface with the Guide Star Selection System software processes pairs of guide stars for given targets to allow accurate positioning. These guide star pairs are used by SPSS for efficient scheduling of different target observations. Vehicle positioning, orbital event computations, orbit management and calendar and candidate list management functions are generic tasks of planning easily convertable to SIRTF applications. Planning support transportability is high.

In summary, most of the overall design and functional level design of SPSS is suitable for and transportable to SIRTF. Approximately 50% of the source code could be reused for SIRTF with minor modification.

Overall transportability of SPSS to SIRTF is medium.

3.3.2 Observation Support System

OSS software supports real-time control of ST science observations. It processes real-time and tape play back science and science engineering data and command requests. Much of the 35,000 lines of executable FORTRAN code is closely tied to the data streams which it processes; however, several functions are transportable to SIRTF. OSS functional organization and lines of code are shown in figure 3.3-2.

The OSS control function processes the receipt of all real-time science and engineering data transmitted through the serial communication lines from PORTS. It performs catalog and storage management functions to handle two-way data transfers; externally with the POCC and internally with the SPSS and PODPS subsystems. Real-time engineering data is saved in short term storage. This processing involves conversion and reformatting of the raw data to a SOGS generic data file format which is used by other OSS software. Design and code transportability of the OSS control function is low.

One SIRTF alternative affecting OSS transportability might be to allow raw telemetry data to be transmitted to the science ground station where it could be decommutated in a manner similar to PORTS. This would make the OSS data handling and receipt functions transportable. This also would allow the other OSS software, which uses the generic data files, to be transportable. In particular, the science imaging and manipulation and the engineering displays of real-time data could be converted to SIRTF applications.

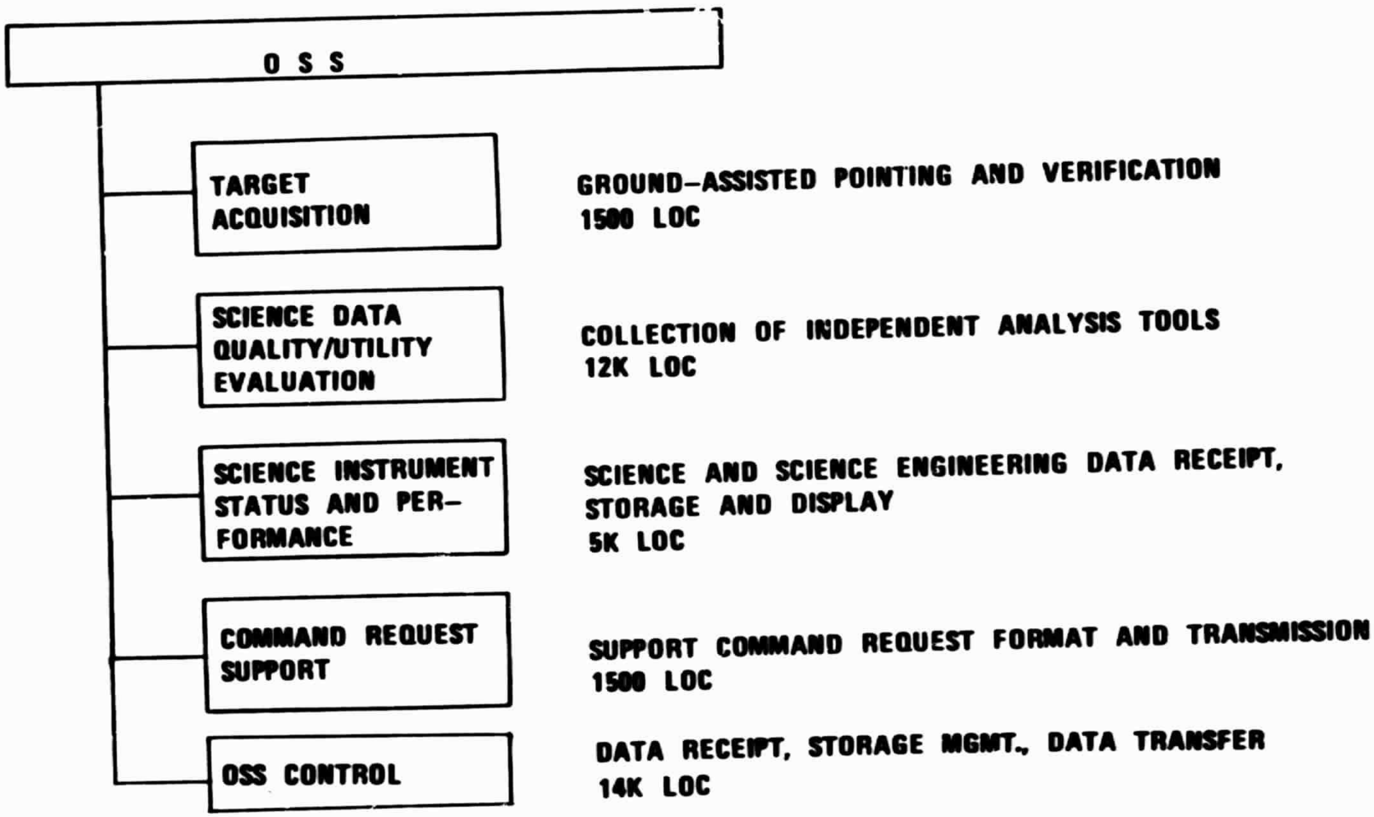


FIGURE 3.3-2 OSS FUNCTIONS

Target acquisition and verification operates in two modes: fixed, pre-planned pointing and ground-assisted. These functions involve sophisticated algorithms designed around the ST camera/imaging devices. Ground-assisted acquisition involves software which translates instrument fixed offsets to ST pointing offsets and roll requests. Target verification is supported by displays and tools (e.g. centroid calculations) to aid manual analysis and compute offsets. Many of these translations and algorithms are table driven, enhancing their portability to SIRTF. In general, transfer curve type data in the infra-red range requires different processing than does mid-ultra-violet to short IR ranges, but many associated functions are similar or identical. The target acquisition function transportability is medium.

The science data quality/utility evaluation function is performed by an assortment of independent analysis tools. These consist of software packages including Fast Fourier Transform, curve fitting and other evaluation routines which are unique to the ST instrumentation. About one third of this function is performed by standard mathematical analysis software packages which would also be useful for SIRTF. TRW proprietary software was developed for the LSI-11 processor interface which drives the DeAnza science imaging and manipulation equipment. Transportability of the science data quality/utility evaluation function is low.

OSS also monitors science instrument status and performance. Real-time science engineering data, saved in short-term storage by OSS control software, is used to produce engineering displays. This software produces displays viewable on any VT100 compatible terminal. However, transportability of the SI status and performance monitoring function is low.

OSS provides the means for real-time control of science activity. This function encompasses support for command requests to be transmitted to PORTS for command processing. As such, it uses the same command format and structure (PSTOL) as PORTS. If a language with the PSTOL syntax is used, transportability of the command request software is high.

Simplified, the OSS design follows the form of discrete functional modules (described above) invoked individually by users. Hence, portability of the OSS conceptual design is applicable only at the functional level as described above.

In summary, OSS software and design are intimately linked to the data format and instruments unique to ST. However, routines performing target acquisition and verification, data logging and, possibly, science and engineering displays (representing approximately 12,000 lines of source code) stand out as functions necessary and convertable to SIRTF.

Overall transportability of OSS to SIRTF is low.

3.3.3 Post Observation Data Processing System

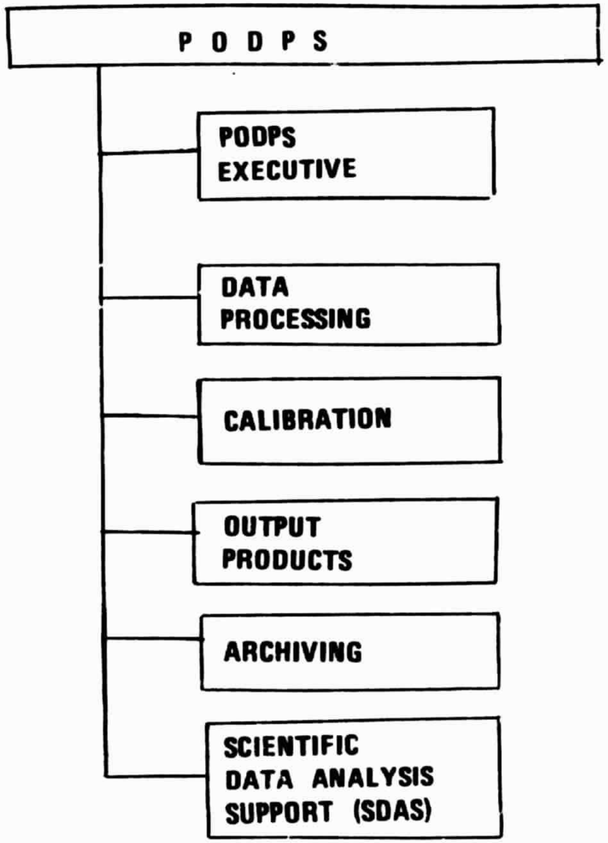
PODPS is responsible for automatic processing of science data and providing the Science Data Analysis System (SDAS) users with interactive analysis capabilities. The 50,000 lines of executable FORTRAN code automatically manage the receipt, editing, calibration and archiving of ST data. In addition, it generates a variety of output products from the raw, edited and calibrated data. The automatic processing is performed by the Routine Science Data Processing (RSDP) portion of PODPS software. PODPS functions and lines of code are shown in figure 3.3-3.

Once the data has been processed by RSDP software and placed in on-line or off-line archives (after 24 hours), it becomes available to the scientist for analysis. Via special SDAS interface software, the user may interactively access the standard RSDP functions to edit, evaluate, convert or calibrate the data. Additionally, graphical images and displays and other output products may be generated. The bulk of the SDAS analysis software is being developed by the ST Science Institute and not addressed in this report.

Conceptual design of the PODPS is very suitable for SIRTF. The automatic processing of RSDP is managed by an event (or stimulus) driven control system called a pipeline. The pipeline method allows users, at scheduling time, to set up limited controls for the automatic processing of RSDP. In addition, the stimulus driven nature of this software supports parallel processing to simultaneously service multiple events. For example, separate RSDP processes can react to the availability of data received from the Data Capture Facility while also processing the arrival of a Real-time Activity file from OSS describing the real-time updates to the SMS for a particular observation. Lastly, many of these RSDP functions are available to off-line users of SDAS. **The transportability of the overall design of PODPS and functional design and code of the automatic processing control software is high.**

The Data Processing portion of PODPS receives, sorts, evaluates and allows editing of science data. Converting the incoming SI data to a generic data file format is an integral part of this process. From these generic data files, most other PODPS functions are performed. **Transportability of the data processing function is low.**

While new algorithms to process SIRTF data will be needed, a significant amount of PODPS code is dedicated to the infra-structure (file handling, data format, message routing, etc.). Acceptance of this generic, internal data file format will make much more of PODPS code transportable.



AUTOMATIC PROCESSING CONTROL AND COMMUNICATION, TERMINAL I/F, SDAS I/F
8K LOC

PARTITIONING, EVALUATION, EDITING
6K LOC

SI DATA CALIBRATION
23K LOC

TAPE, FILM, H/C PLOTTER, GRAPHICS DISPLAY,
5K LOC

OFF-LINE GENERATION AND RETRIEVAL
2K LOC

UTILITY I/F TO SDAS USERS
5K LOC

FIGURE 3.3-3 PODPS FUNCTIONS

Science Instrument data calibration software is closely linked with the ST instrumentation and, with the possible exception of use of the generic file format, has little potential for use on SIRTF. **Transportability of the SI calibration function is low.**

All data is stored temporarily in on-line discs and permanently archived on tape. General observer tapes in Flexible Image Transportation System (FITS) format and other user products are generated. Note that production of output tapes is not fully automatic and requires much human handling. PODPS also supports interactive access to RAMTEK graphic workstations, DeAnza image workstations, hardcopy plotters and a film writer. **Transportability of the archiving and product functions is high.**

Summarizing, the design of PODPS conceptually and at the functional level could be re-used for SIRTF. However, much of the source code is unique to ST instrumentation and data stream format.

Overall, transportability of PODPS to SIRTF is medium.

3.3.4 System Support

SOGS system support software consists of 65,000 lines of executable source code and can be grouped into three general areas: operating system supplementation, command language interface and real-time simulation. These functions and their associated lines of code are shown in figure 3.3-4. The bulk of SOGS software is implemented in FORTRAN and makes liberal use of the DEC VMS operating system and OMNIBASE DBMS utilities. This software is a critical component of transportability. It provides functions necessary to the other SOGS software.

Developed software is layered on top of VMS to augment the operating system capabilities. Support functions and increased capabilities were developed for network and system management, communication services (both internal and external to SOGS) and computer initialization, termination and failover. In addition, specialized processing for error message reporting, handling and recovery is used by the SOGS applications software. **Assuming these functions are compatible with the final SIRTF operations concept, transportability of the operating system augmentation software is high.**

System support software also includes providing a command language interface for SOGS users called COMET. COMET is an extension of the DEC Command Language (DCL) and, as such, transportability is high to any compatible operating system environment. However, this interface is a very visible and subjective system component that must be comfortable to the end user and, more often than not, desired to be state-of-the-art (e.g. graphical windows, pop-up menus, icons, etc.). Therefore it is possible that it would be required to be redone for SIRTF

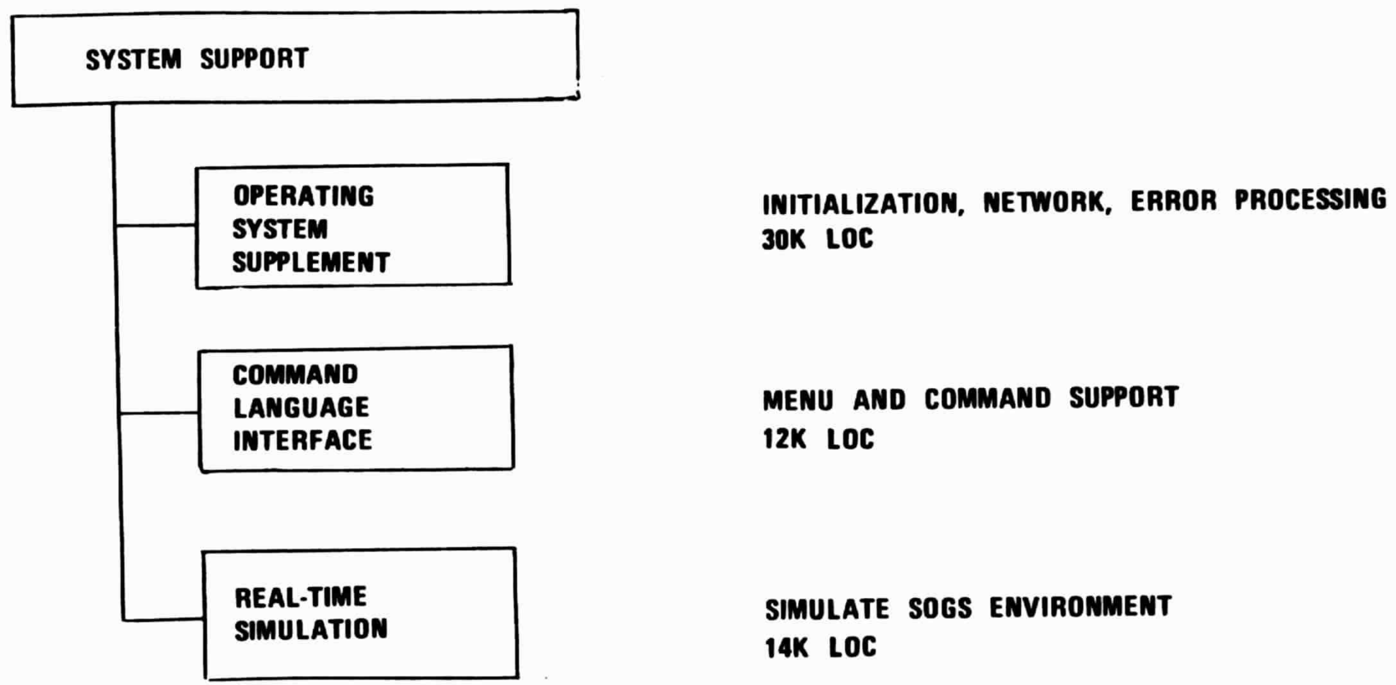


FIGURE 3.3-4 SYSTEM SUPPORT FUNCTIONS

implementation, subject to the fads and technologies of the late 1980's. This does not prevent it from being software calling sequence compatible with the current COMET. Transportability of the command language interface is high.

The data simulation software was developed to produce a realistic environment for the integration and testing of SOGS. It creates data streams emulating the data SOGS receives, thereby providing the capability to exercise the applications software systems with operational or simulated data in a transparent manner. There are two basic parts of the simulator. The data generator creates ST data stream files on disk. Transportability of this part is low. The real-time simulator runs off a control file that defines the emulation of real-time operation in terms of stimulus, resulting file to be sent, communication line, timing, volume, request, etc. Transportability of the simulator is high. Transportability of the real-time simulation function to SIRTF is medium.

Overall, system support software provides capabilities that, in a compatible environment, would be very useful to SIRTF.

Transportability of System Support software is high.

3.4 Hardware

The SOGS architecture is based primarily on 1982 hardware technology. Any transportability of this hardware is based on the continued availability and cost effectiveness of this hardware or compatible upgrades.

SOGS operates on six DEC VAX computers: two at the GSFC and four at the ST Science Institute Facility. This hardware configuration provides functional duplication and separation of software components. Five of the computers are needed for operation. The remaining one is used for simulation, back-up and supplementary off-line support. Much of the peripheral equipment is dual ported to support switching during a manual failover. SOGS software is easily transportable to any similar network of VAXs. This is true for any of the VAX models including the two recently announced upgrades which are 1.5 and 4.0 times more powerful than the 11/780 machines being used for SOGS. Basic hardware architecture transportability to SIRTF is high.

SOGS hardware also includes two Britton-Lee IDM 500 Data Base machines. These were chosen to provide high speed data base access to mass storage and are used extensively by the archiving and science planning and scheduling software. Timing constraints, minimum storage requirements, and query capability were the primary factors in choosing to use a data base machine instead of a software DBMS. A significant amount of SOGS software is closely tied to the DB machine in terms of reduced software and dependence on access speed. Data base hardware transportability to SIRTF is high.

SOGS utilizes sophisticated display work stations consisting of RAMTEK graphics terminals and DeAnza image displays driven by LSI 11 processors. The bulk of image and graphic manipulation is carried out on these work stations through dedicated keypads. The software provides general purpose manipulation capabilities plus overlays keyed to astronomical use. Display work station transportability to SIRTF is high.

Hardware used for external communication is also suitable for SIRTF. Special electronic circuit boards form the NASCOM Interface Unit and the X.25 Interface Unit (for communication with the DCF). In addition, if the SIRTF ground station is split (as in SOGS), the inter-facility multiplexing equipment could be used. Transportability of the communication hardware is high.

3.5 Conclusions

Based on the findings of the task 2 study, the SOGS software is appropriate for application to SIRTF. For the level of detail thus far specified by NASA, SOGS software performs all the required SIRTF functions and in some cases, provides extra capability (scheduling options available in SPSS for example). SIRTF, also will operate in an environment similar to the ST in terms of interfaces with NASCOM, TDRSS, DCF, GSSS and the science community.

However, for the SOGS software to be transportable, certain compatible system components must be present: basic VAX computer design, VMS operating system and system support software. Each of these is easily transportable and, together, could form the basis of a SIRTF system.

Assuming that such an environment is provided and that the SIRTF concept of operations continues to evolve to resemble Space Telescope, the SOGS applications software systems are moderately transportable to SIRTF. The Science Planning and Scheduling System design and source code have medium transportability. Conversion of SPSS will, however, provide capabilities in excess of current plans for SIRTF operation (e.g. ability to schedule simultaneous SI activity). The Observation Support System, while having a number of functions transferable to SIRTF, has low transportability. Post Observation Data Processing is the area most lacking detail in SIRTF. The SOGS PODPS has extensive processing capabilities and medium transportability.

Ultimately, NASA should benefit from reusing SOGS software. The advantages to this approach lie in lowering costs and reducing the risk of a large software development effort. Towards this end, it is recommended that NASA continue to refine the SIRTF operations concept and in particular, deal with issues raised in section 3.2.3. In addition to continuing the top/down evolution of the operations concept, a bottoms/up user engineering approach will help focus attention on system issues needing early resolution.

APPENDIX A

APPLICATION OF THE DCDS METHODOLOGY AND TOOL SET TO THE SSDS STUDY

TRW's Distributed Computing Design System (DCDS) was developed to uniformly control the complete life cycle of system and component development, and as the means to manage that complexity. DCDS procedural techniques define the sequence of steps in systems development. These steps instill a formalism which identifies the data base contents, produces outputs in increments, and provides the criteria for completeness/correctness of outputs at each phase of development. An example showing these steps is presented at the end of this appendix.

The definition of the Space Station Information System (SSIS) functions, their decomposition and allocation to the SSDS, and the definition of the interface specifications are expressed in the DCDS specification language. This covers the traditional specification levels (A, B1) for the data system. When these requirements are decomposed into black box testable requirements, the results are expressed in the DCDS data base. The definition of the distributed network, the definition of the processing tasks, and the definition of the communication requirements are also represented in the DCDS specification language. DCDS is also used to define the allocation of requirements between data processing hardware and software and to describe the process design for individual processors. Operating system requirements are defined by both the hardware-software interfaces and the intra-task interfaces.

Basic to the DCDS methodology is the formalization of decomposition of functions in terms of a graph model of decomposition. Previous methods of decompositions have failed to capture simultaneously the characteristics of preserving input/output concurrency and structure, of preserving the exit criteria, and of preserving performance traceability and computability.

The DCDS Specification language provides the means to express the system characteristics in terms of graph model structures, elements, attributes and relationships, for the Systems Definition Phase. The methodology for the Systems Definition Phase is viewed as a sequence of steps which input and output contents to a DCDS data base, thus resulting in the complete filling of the data base with all associated consistency criteria.

The desire to express a methodology as a sequence of steps leads to an apparent contradiction when issues of resource management and fault tolerance are addressed. Classical topdown systems engineering theory (e.g., MIL-STD-499) suggests that one should decompose functions and then allocate to subsystems. It is only after allocation that subsystem resources needed to accomplish each function can be identified, total utilization estimated, and hence, the need for resource management can be established. Since resource management is a control function which requires allocation, this must be represented back at the system level.

Similarly, classical systems engineering theory proposes that one first define system functions and allocations, then perform a failure mode effects analysis to identify potential problems, and then add functions to detect, isolate, and recover from the failures. These functions are also to be decomposed and allocated to the subsystems.

The DCDS composition graph (shown in the example at the end of this appendix is explicitly designed to be produced in several layers, with the first layer representing the standard flow (i.e., sufficient resources and no faults). Additional layers of functions and paths (not shown in the example) are added to address resource/fault exceptions by adding appropriate exits to the functions on the standard flow, and then identifying how the exception will be handled locally with entry back into the standard flow. If recovery is not possible, an exception exit of the composition occurs which must match the exception layer of the previous level of decomposition.

This approach has distinct advantages of providing a separation of concerns for handling complex problems. First the standard flow is defined, and then layers of requirements are added for each class of exception without disturbing previous layers. This separation of concerns promotes a systematic methodology for doing the system design and presenting the results of the design activity.

The DCDS methodology uses this concept to define the layers of requirements in the following way:

1) The system requirements are decomposed to black box testable requirements (to support integration testing).

2) These requirements are decomposed and allocated to subsystems.

3) Interface designs are defined to accomplish the interface passing of data and control.

4) Subsystem resources are estimated to perform the allocated functions, and a layer of resource management requirements is added to be decomposed, allocated, and interface designed.

5) A failure Mode Effects Analysis is made, and a layer of requirements is added for each class of failure desired. These new functions are decomposed, allocated, interface designed, and resource managed.

The sequence of concerns (i.e., decomposition/allocation/interface design/resource management/fault tolerance) is applicable to all levels of system component design.

The next effort, following the system definition phase, is the data system requirements definition phase. The purpose of this phase is to transform the data processing function and performance requirements, previously defined in system terminology and parameters, into a more detailed definition of data systems requirements expressed in testable stimulus-reponse (end-to-end processing) terms. The starting point of this phase is where the system definition phase has identified functions and operating rules; and the interfaces between the logical subsystems; and the functions have been decomposed and allocated to the data system. The logical subsystems and their associated operating rules, interfaces, messages, and objects are already identified in the DCDS Specification Language together with the functions allocated to the data system. Note that the information recorded in the DCDS specification language resides in unified DCDS data base.

The attainment of the data system processing requirements is accomplished utilizing the next phase of the DCDS shown in Step 6 contained in the example at the end of this appendix. Although there are several requirement techniques in use today, all but DCDS are based on function decompositions. TRW research has shown that the major difficulties from the functional decomposition approach are that processing sequences and related conditions cannot be clearly depicted without "threading" the specification. Since this thread typically runs through several functions, it is difficult to allocate performance requirements across the several functions, given the number of threads to be considered.

DCDS overcomes problems caused by functional decomposition in a very natural way by utilizing a stimulus-response point-of-view which examines requirements using a concept called Requirements Network, or R_NET, one of which is developed for each input interface. For each message received across an interface, the sequence of processing and the conditions under which various paths of processing may occur, including error paths, are defined in R_NETS (see step 6) to describe the data processing actions required to service that message.

The major benefits of this stimulus-response approach are that it facilitates the creation of a black box testable and executable model of the data system (without global timing) from which the deliverable software can be derived. That means it reduces errors by transforming an English description of the system to a machine analyzable model as early as possible. This performs the critical step of ensuring that the model meets the intent of the English Language specification.

The stimulus-response approach of DCDS results in an early detailed understanding of the requirements - an understanding that is essential to risk reduction.

After the requirements have been uniquely identified and both allocated to computer complexes and labeled as hardware and software requirements, a DCDS software requirements data base and a DCDS hardware requirements data base are developed for each computer complex.

The steps performed in preparing the DCDS computer complex requirements data base are:

1) Identify, define, and record in the DCDS Requirements Data Base:

- o Each uniquely identified data tracking requirement with each requirement traced to the System Level Specification

- o Input and output interfaces between the Computer Complex and other Computer Complexes and external equipment (e.g., radars, NASCOM, etc.)

- o The contents of each message received or sent by the computer complex and the corresponding processing performance requirements

- o The data needed by data processing to process the input messages along with the internal data required to complete the processing and to output the required result
- o The Requirements Networks (R_NETS) that describe interfaces, processing steps, data flow and control flow. One R_NET is prepared for each input interface entering the Computer Complex, with separate processing branches for each message. The R_NET data flow is defined by identifying the inputs and outputs to each of the sequenced processing steps. The control flow is defined in terms of data needed to make conditional decisions for branching. In addition, R_NETS which are enabled by events initiated within another R_NET, e.g., an alarm activated by an error condition, are also developed. At the completion of this step, all of the processing paths will have been identified. Iteratively check the DCDS Requirements Data Base to determine the consistency and completeness of the data entered by querying it with a set of static commands. Examples of such queries are:
  - Are there any messages whose data contents are not defined?
  - Are there any data items used in the processing that are not available because they haven't been provided by some other processing step, input message or by initialization?

2) Apply the Data Flow Analyzer from the DCDS tool set to each R_NET. This differs from the static checks in that it performs the thread and data flow analysis for each individual R_NET as opposed to examining the integrity of the DCDS data base using the commands identified above. This step accomplishes the different task of making sure every message is properly processed.

3) Extract from the DCDS computer complex requirements data base the input/output timing requirements for each R_ NET. Determine and record the validation points; i.e., the information that must be recorded during testing to validate each performance requirement. The definition of a validation point includes not only the data needed but the path and points on the R_NET where this data is to be captured during testing.

When the DCDS Requirements Data Base is complete, and all issues resolved, there is high assurance that the data processing requirements are complete, consistent with respect to interfaces, unambiguous in the sense that the high level requirements have been properly interpreted via R_NETS, traceable to higher level requirements and testable.

An example of DCDS applied to the SSIS and to the data management system portions of it (SSDS) follows in figure A-1:

A Space Station Information System Application of DCDS

The examples in the next several pages show how we apply the techniques of the Distributed Computing Design System (DCDS) to the problem of defining and documenting the functional requirements, design/technology options, trade studies, and system design of the SSIS/SSDS. For ease and brevity of presentation we have selected a relatively simple aspect of the SSIS/SSDS problem which will nevertheless still present most of the powerful capabilities of DCDS. The presentation will follow the step-by-step DCDS disciplined approach to the system engineering analysis of the SS system. Included in each step are a set of graphics representing the state of the system at that point in the approach. The steps are also represented by a sample of the DCDS specification language which when completed leads to requirements specifications.

Step 1

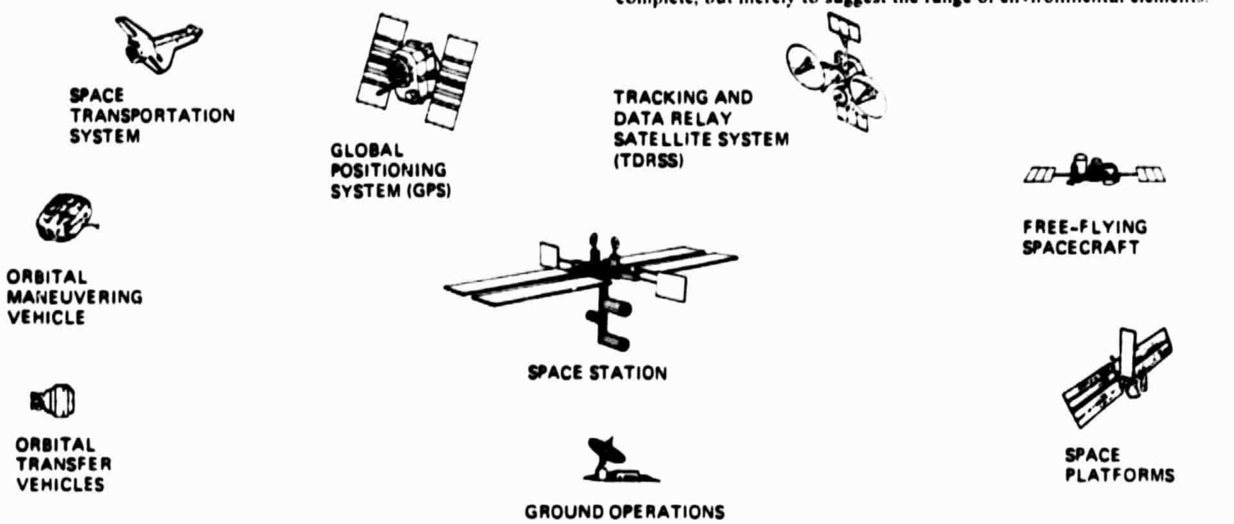
DCDS Viewpoint

Step 1: Define the system and its environment. The view adopted is that the system is a black box within a sphere of perception. The list of elements which move in and out of this perception space, taken together with any physical conditions within that space, are taken as the system's environment.

Space Station Application

Step 1: Remembering that our objective is the systems engineering description and analysis of the SSIS and SSDS we take our black box system to be the physical Space Station and associated Ground Operations (see below). The environment is composed of the elements of the space station infrastructure which move in and out of the perception space of this system. The illustration is not intended to be complete, but merely to suggest the range of environmental elements.

FIGURE 1. DEFINE SYSTEM AND ENVIRONMENT



DCDS Specification Language

```
r..\.......................Step 1 inputs..............................

SYSTEM   SPACE_STATION_PLUS_GROUND
      PERFORMS FUNCTION PERFORM_SPACE_STATION_FUNCTIONS
      INPUTS  ITEM ITEMS_ENTERING_SPACE_STATION_PERCEPTION_SPACE
      OUTPUTS   ITEM ITEMS_EXITING_SPACE_STATION_PERCEPTION_SPACE
      EXHIBITS PERFORMANCE_INDEX LIFE_CYCLE_COST

      DESCRIPTION   "The space-station-plus-ground system is composed of
the space station itself, all ground facilities devoted to the space
station, and the perception space of the space-station-plus-ground.
Objects entering this perception space include the following
1)  GPS satellites, used for estimating position
2)  TDRS satellites, used to communicate information from space station
      to ground, and ground to space station
3)  Orbital Maneuvering Vehicles (OMVs), used to retrieve satellites
      for servicing..."
:
ITEM_COMPOSITION ITEMS_ENTERING_SPACE_STATION_PERCEPTION_SPACE _
DESCRIPTION

      REFINES ITEMS_ENTERING_SPACE_STATION_PERCEPTION_SPACE


DESCRIPTION   "The items entering the perception space of the space
station include satellites, OMVs, Space Transportation Vehicles ....
and messages/controls from the ground."
      SUBGRAPH LAYER STANDARD_FLOW
         PARALLEL WITH_COORDINATION  NULL
           BRANCH ITEM GPSS
           BRANCH ITEM TRRSS
           BRANCH ITEM  OMVS

         END_PARALLEL
      END
```

Figure A-1. Sequence of Steps in System Development

Step 2

Step 2A: Identify the components of the black box system. This identification process should list all internal components of the system as well as the external elements that occupy the system's environment.

Step 2A: We define the necessary list of components as shown below:

External Elements:

- STS/Orbiter
- OMV
- OTV
- Space Platforms
- Serviceable FFs
- TDRSS
- GFS
- Space Environment (low g, high vacuum, orbital conditions, etc.)

Internal Elements:

- Space Station
- Ground Operations

See Figure 2A for detailed breakdown

Step 2B: Identify inputs and outputs of all types which are expected to be associated with the black box system and its environment. Identify the components of the system affected by each type of input or output.

Step 2B: Figure 2B presents the system and its components, and illustrates some of the typical inputs/outputs expected to involve the SS and Ground Operations. The figure also seeks to suggest the distributed nature of the SSIS and hence the fact that the SSIS is not a subsystem in the conventional sense of the SS/ Ground Operations system.

FIGURE 2. A INTERNAL SYSTEM COMPONENTS FOR SPACE STATION AND GROUND OPERATIONS

NOTE: THE COMPONENTS TITLES USED HERE REPRESENT TECHNICAL ACTIVITIES WHICH MUST BE ACCOUNTED FOR IN TERMS OF FUNCTIONAL AND PERFORMANCE REQUIREMENTS. THE FINAL DEFINITION OF SS AND GROUND SUBSYSTEMS WILL RESULT FROM THE ANALYSIS OF FUNCTIONAL REQUIREMENTS AND TOP DOWN SYSTEMS DESIGN RESPONSIBLE TO SS PROGRAM AND MISSION GOALS.

SPACE STATION:

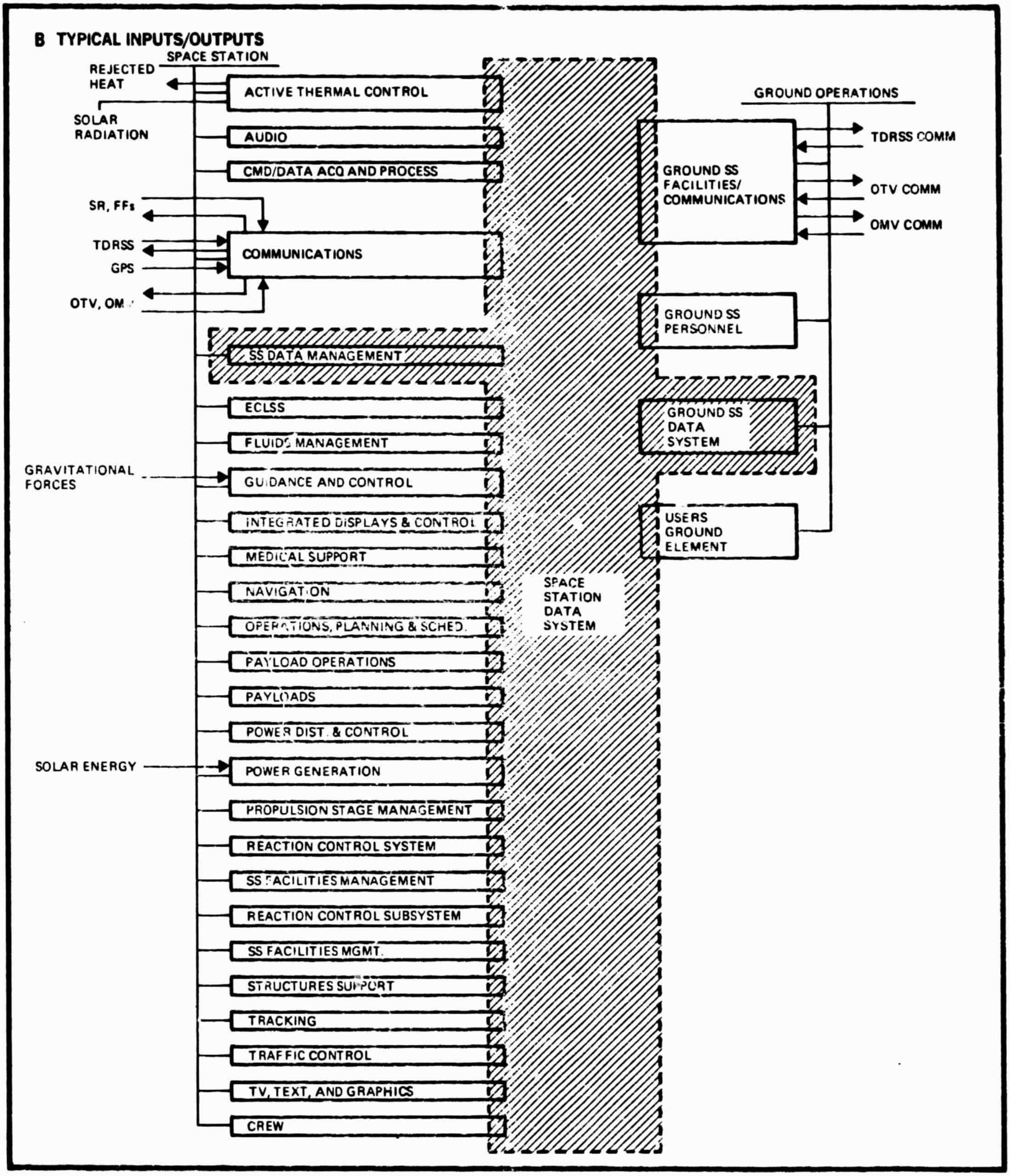
- ACTIVE THERMAL CONTROL
- AUDIO
- COMMAND/DATA ACQUISITION AND PROCESSING
- COMMUNICATIONS
- DATA MANAGEMENT
- ENVIRONMENTAL CONTROL AND LIFE SUPPORT
- FLUIDS MANAGEMENT
- GUIDANCE AND CONTROL
- INTEGRATED DISPLAYS AND CONTROLS
- MEDICAL SUPPORT
- NAVIGATION
- OPERATIONS, PLANNING AND SCHEDULING
- POWER DISTRIBUTION AND CONTROL
- POWER GENERATION
- PROPULSION STAGE MANAGEMENT
- REACTION CONTROL SUBSYSTEM (RCS)
- SPACE STATION FACILITIES MANAGEMENT
- STRUCTURES SUPPORT
- TRACKING
- TRAFFIC CONTROL
- TV, TEXT AND GRAPHICS
- CREW
- PAYLOADS

GROUND OPERATIONS:

- GROUND SS FACILITIES
- GROUND SS PERSONNEL
- GROUND SS DATA SYSTEM
- USERS GROUND ELEMENT

```
.........................Step 2 inputs.............................

CONFIGURATION SPACE_STATION_PLUS_GROUND_COMPONENTS_NO_1

     BUILDS SYSTEM  SPACE_STATION_PLUS_GROUND

     DESCRIPTION  "This configuration of the space station plus ground
components is defined in terms of three sets of components: a set of
external components which reside in the environment, a set of
components which will reside on the space station, a set of ground
components, and a space station information system (SSIS) consisting of
all of the data processors and associated software and communicaations
which perform all space station computing";

     CONTAINS
        SUBSYSTEM EXTERNAL_GPSS
        SUBSYSTEM EXTERNAL_TDRSS
        SUBSYSTEM EXTERNAL_OMVS
        :
        SUBSYSTEM SPACE_STATION_INFORMATION_SYSTEM
        SUBSYSTEM SS_THERMAL_CONTROL
        SUBSYSTEM SS_AUDIO
        SUBSYSTEM SS_COMMUNICATIONS
        SUBSYSTEM SS_CREW
        :
        SUBSYSTEM GROUND_FACILITIES
        SUBSYSTEM GROUND_CREW
        SUBSYSTEM GROUND_COMMUNICATIONS
```

Figure A-1. Sequence of Steps in System Development (Continued)


B TYPICAL INPUTS/OUTPUTS
SPACE STATION
REJECTED HEAT
SOLAR RADIATION
ACTIVE THERMAL CONTROL
AUDIO
CMD/DATA ACQ AND PROCESS
SR, FFs
TDRSS
GPS
OTV, OMV
COMMUNICATIONS
SS DATA MANAGEMENT
ECLSS
FLUIDS MANAGEMENT
GRAVITATIONAL FORCES
GUIDANCE AND CONTROL
INTEGRATED DISPLAYS & CONTROL
MEDICAL SUPPORT
NAVIGATION
OPERATIONS, PLANNING & SCHED.
PAYLOAD OPERATIONS
PAYLOADS
POWER DIST. & CONTROL
SOLAR ENERGY
POWER GENERATION
PROPULSION STAGE MANAGEMENT
REACTION CONTROL SYSTEM
SS FACILITIES MANAGEMENT
REACTION CONTROL SUBSYSTEM
SS FACILITIES MGMT.
STRUCTURES SUPPORT
TRACKING
TRAFFIC CONTROL
TV, TEXT, AND GRAPHICS
CREW
SPACE STATION DATA SYSTEM
GROUND OPERATIONS
GROUND SS FACILITIES/ COMMUNICATIONS
TDRSS COMM
OTV COMM
OMV COMM
GROUND SS PERSONNEL
GROUND SS DATA SYSTEM
USERS GROUND ELEMENT

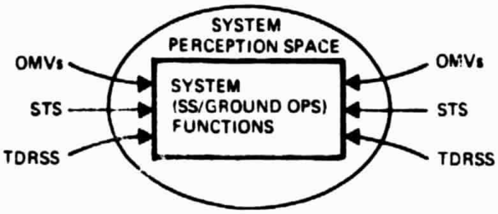
Step 3

DCDS Viewpoint

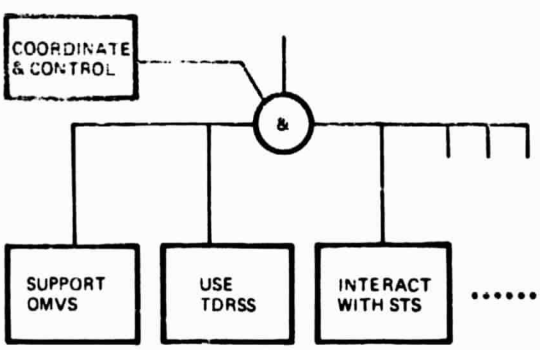
Step 3: Decompose the system to the point where the functions and interfaces of this black box can be identified. The objective of this step is the discovery of the system level functional requirements and interfaces.

FIGURE 3. DECOMPOSING THE SYSTEM

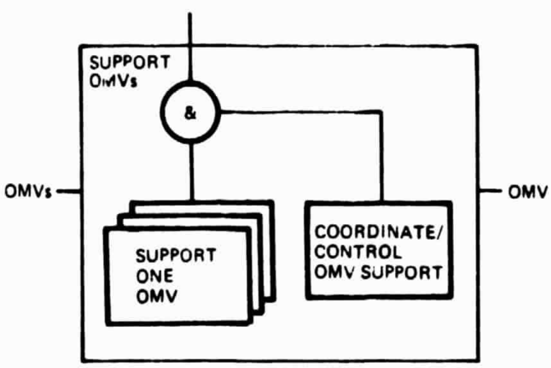
A. OVERALL SCHEMATIC



B. FIRST STEP IN DECOMPOSING SYSTEM



C. FIRST LEVEL DECOMPOSITION FOR SUPPORT OMVs



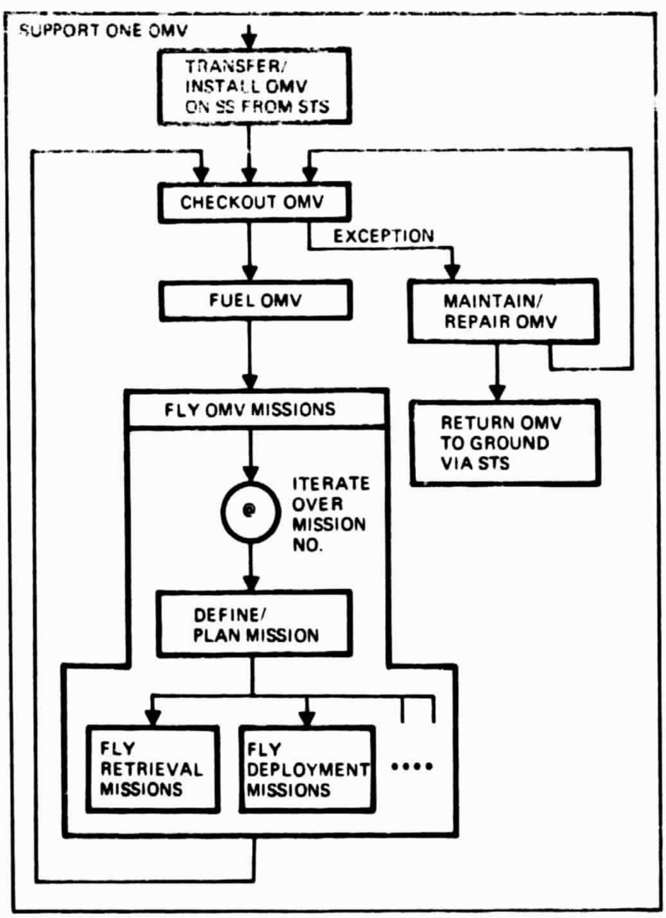
Space Station Application

Step 3: A schematic view of the situation we have been constructing is shown in Figure 3A. This figure suggests that the black box system (the physical SS plus ground operations) encounters a wide variety of external elements entering its perception space. The system performs functions of various sorts on these elements and they pass out of the system's perception space. (A more detailed list of SS functions is in Section 2.2.1.)

The first step of our decomposition of the system is illustrated in B. The top level functions of the system involve such parallel (concurrent) functions as Support OMVs, Use TDRSS, Interact with STS, etc. A coordinate and control function will arise every time there is a situation involving parallel activities. Indeed, the DCDS technology enables us to define and keep track of a hierarchy of control functions. In the course of the application of DCDS to the SSIS/SSDS problem, each of these high level functions would be further decomposed. However, for the purposes of this example we elect to decompose only one of the system functions, i.e., Support OMVs.

Figure C illustrates the first level decomposition of the system level functional Support OMVs. The analysis makes provision for the possibility that more than one OMV will eventually be involved in SS operations. Thus, this decomposition introduces the idea that a simpler function (Support One OMV) exists and that this could be operated concurrently for as many single OMVs as needed. It also introduces the concept of a coordinate/control function to handle multiple OMVs. This is a next level down in the previously mentioned hierarchy of control.

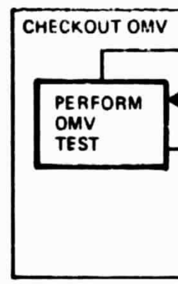
D. TIME-ORDERED FUNCTION SEQUENCE



The next step in our
Support One OMV i
figure is not an exhau
be repeated across th
functional flow starti
STS/Orbiter and sub
was chosen since it r
perception space of t
by a checkout OMV
procedure, we decom
remainder of D is wo
the ability of DCDS t
exceptional sequence
OMV does not pass
Maintain/Repair ON
return the OMV to t
result in the return o

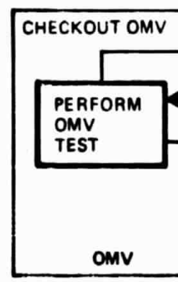
The decomposition c
ultimate objective in
of the simple functio
OMV to the OMV it
allocate the Direct/C
option as OMV Self
SS/Ground Ops syst
as an input and to m
the Direct/Control T
system. This leads t
commands as a syst
The repetition of thi
produces the result
functions, inputs an
by operations withi

E. OPTION 1 – OM



CONCLUSION: C
B
A
A

F. OPTION 2 – SS/



CONCLUSION: OU
BE
ON
(A
DI

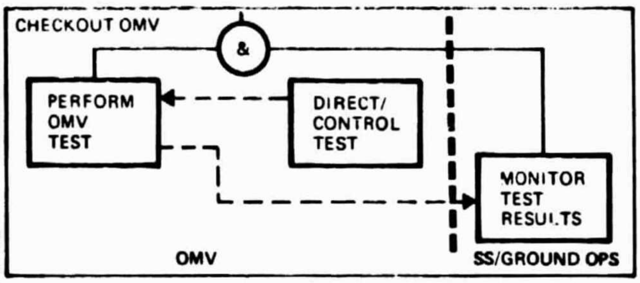
Figure A-1. Sequence of Steps in System Development (Continued)

FOLDOUT FRAME

The next step in our decomposition (see D) is to break the function Support One OMV into a time ordered sequence of functions. The figure is not an exhaustive breakdown, it illustrates a step which must be repeated across the entire system's functional spectrum. It shows a functional flow starting with transfer of an OMV from the STS/Orbiter and subsequent installation on the SS. This initial step was chosen since it represents the first way the OMV can enter the perception space of the system of SS/Ground Ops. This is followed by a checkout OMV function. To complete this step of the DCDS procedure, we decompose the Checkout OMV function. The remainder of D is worth reviewing before we go on, since it illustrates the ability of DCDS to deal with a nominal time sequence as well as an exceptional sequence of functions. To deal with the situation where an OMV does not pass the Checkout function we have provided a Maintain/Repair OMV function. This function would presumably return the OMV to the Checkout function and the nominal path or result in the return of the OMV via Shuttle to the ground.

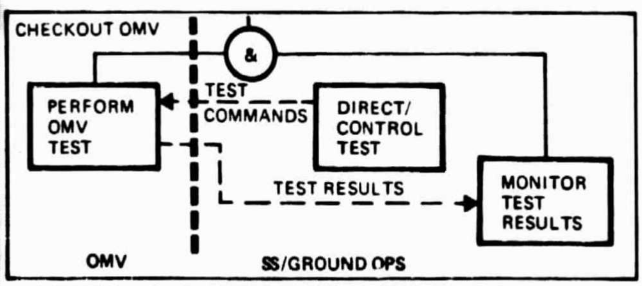
The decomposition of the Checkout OMV function leads to our ultimate objective in Step 3. Figures E and F are alternate allocations of the simple functional blocks we derived for Checkout OMV to the OMV itself and to our system (SS/Ground Ops.). In E we allocate the Direct/Control Text block to the OMV and designate this option as OMV Self Test. We also learn that in this option the SS/Ground Ops system must be prepared to receive OMV test results as an input and to monitor these results as a function. In F we allocate the Direct/Control Test function to the SS/Ground Ops black box system. This leads to the discovery that this option requires test commands as a system output and test results as a system input. The repetition of this type of allocation across all system functions produces the result we were seeking. Namely, the definition of the functions, inputs and outputs imposed on the SS/Ground Ops system by operations within its environment.

E. OPTION 1 – OMV SELF TEST



CONCLUSION: OUR BLACK BOX SYSTEM (SS/GROUND OPS) MUST BE PREPARED TO RECEIVE OMV TEST RESULTS AS AN INPUT AND TO MONITOR THESE TEST RESULTS AS A FUNCTION.

F. OPTION 2 – SS/GROUND OPS DIRECT, MONITOR TEST



CONCLUSION: OUR BLACK BOX SYSTEM (SS/GROUND OPS) MUST BE PREPARED TO ISSUE TEST COMMANDS TO THE OMV, (AN OUTPUT) AND RECEIVE OMV TEST RESULTS (AN INPUT) AND TO PERFORM THE FUNCTIONS OF DIRECT/CONTROL TEST AND MONITOR TEST RESULTS.

Step 3 inputs

```
FUNCTION_COMPOSITION
PERFORM_SPACE_STATION_FUNCTIONS_DECOMPOSITION_NO_1

     PERFORMS FUNCTION PERFORM_SPACE_STATION_FUNCTIONS

     DESCRIPTION   "This decomposition approach divides the functions
of the space station into the functions for dealing with each of its
inputs"

     SUBGRAPH LAYER STANDARD_FLOW
        BRANCH FUNCTION SUPPORT_OMVS
        BRANCH FUNCTION USE_TDPSS



     END_PARALLEL
END


FUNCTION_COMPOSITION_SUPPORT_OMVS_DECOMPOSITION_NO_1

     DECOMPOSES FUNCTION SUPPORT_OMVA

     DESCRIPTION   "This decomposition approach exPresses the overall
function of supporting all OMVs into a function for supporting each OMV
Plus a coordination function."

     INPUTS ITEM ONE_OMV
        ITEM GROUNDCOMMANDS_FOR_SUPPORTING_OMV

     OUTPUTS ITEM ONE_OMV

     SUBGRAPH LAYER STANDARD_FLOW
        REPLICATE DOMAIN SET NUMBER_OF_OMVS
          WHITH_COORDINATION FUNCTION COORDINATE_OMVS
          FUNCTION SUPPORT_ONE_ONV
        END_REPLICATION
END

FUNCTION_COMPOSITION SUPPORT_ONE_OMV_DECOMPOSITION_NO 1

     DECOMPOSES FUNCTION SUPPORT_ONE_OM

     DESCRIPTION   "This decomposition of the SUPPORT_ONE_OMV function
exPresses the overall function in terms of a time [illegible] of actions
to accePt OMV delivery from an orbiter, install the OMV, check it out, .
"

     SUBGRAPH LAYER STANDARD_FLOW
        FUNCTION ORBITER_DELIVERY_OF_OMV
        FUNCTION INSTALL_OMV
        FUNCTION CHECKOUT_OMV
        FUNCTION FLY_OMV_MISSIONS

     ENI_LAYER STANDARD_FLOW



  LAYER EXCEPTIONS
     HANDLER HANDLE_OMV_CHECKOUT_FAILURE
       ENTRY_LIST EXCEPTION OMV_CHECKOUT_FAILURES
         ENTRY FROM FUNCTION CHECKOUT_OMV
                 FUNCTION FLY_OMV_MISSIONS
       END_ENTRY_LIST
     FUNCTION OMV_MAINTENANCE_AND_REPAIR
       SELECTION
         BRANCH (MAINTENANCE_STATUS = REPAIRED
           GO_TO CHECKOUT_OMV
         BRANCH (MAINTENANCE_STATUS = NOT_REPAIRED
           FUNCTION RETURN_TO_GROUND
       END SELECTION
```

DCDS: DEFINE OPTIONS

```
INISION   D_001
EQUATED TO


SYNONYM    CHECKOUT_OMV
PROBLEM    "Current requirements do not specify where DIRECT_CONTROL_OMV_
TEST IS TO BE PERFORMED".

ALTERNATIVES
   1 -  OMV shall have the capability to Perform a 'self test' and the
        SS Ground operations shall monitor the results."
   2 -  The SS Ground Operations shall have the capability to direct,
        control, and monitor the OMV checkout tests.•

DATE_OPENED    "09-26-83"
TRACES FROM    NASA DOCUMENT.....
               PARAGRAPH.... "CHECKOUT OMV"
ASSUMPTION    "The alternatives most consistent with the Philosophy of
               this illustration is alternative 2"
DATE UPDATED   "09-26-83"
TRACED TO      DIRECT_CONTROL_OMV_TEST
CHOICE
DATE_CLOSED
```

Step 4

Space Station Application

DSDS Viewpoint

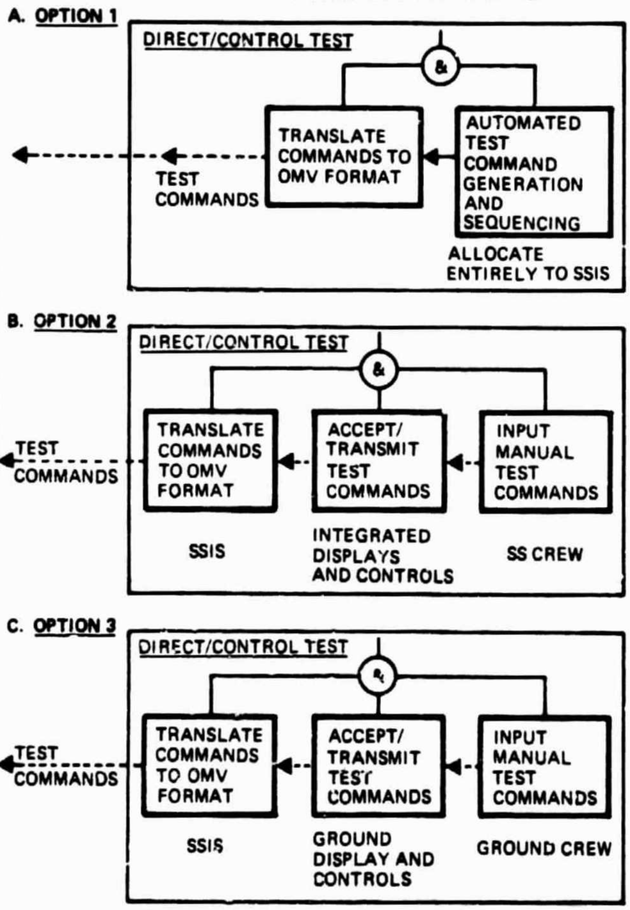
Step 4. From Step 3 we have the functions which are to be carried out by the black box system. We now decompose those functions for the purpose of allocation of functional requirements to the components of the black box system.

Step 4: One objective in this step is decomposition of functions allocated in Step 3 to the SS/Ground Ops system for the expressed purpose of allocating functions to the SSIS/SSDS, and to distinguish between functions to be done on the SS and those to be done in Ground Operations. We continue our example by decomposing the Direct/Control Test function shown in Figures E and F in Step 3.

Figure 4 illustrates three alternate allocations of the function Direct/Control Test. Figure A is a case where the function is carried out entirely with the SSIS as an automated activity. Figure B allocates to the SS crew the task of manually inputting test commands which are accepted by some type of console/display system of the SS's Integrated Displays and Controls component and finally a SSIS function translates these commands for transmission to the OMV. Figure C introduces the possibility that the OMV testing could be directed manually by the ground crew and through the use of a communications line, the same SSIS function used in 4B could again format the commands for transmission to OMV.

While many of the details of the actual SS problem are being left out of these illustrations, they do show that the DCDS approach allows us to systematically allocate functions to the SSIS and other SS/Ground Ops components. It has a natural means of handling options for the allocaton of functions, and through its powerful software it can continuously check the growing tree of functional requirements, and interface for consistency and completeness.

FIGURE 4 DIRECT/CONTROL TEST OPTIONS



DCDS: DEFINE OPTIONS

```
DECISION   D_002
EQUATED TO

SYNONYM    DIRECT_CURRENT_OMV_TEST
PROBLEM    "Current requirements do not specify both the extent of
           automation for OMV checkout and ground personnel
           involvement".

ALTERNATIVES
1 - The function of directing and controlling the OMV checkout shall be
    performed entirely by the SSIS as an automated activity.
2 - The function of directing and controlling the OMV checkout shall be
    directed by manual input from the SS crew.
3 - The function of directing and controlling the OMV checkout shall be
    directed by manual input from the ground operations crew.

DATE_OPENED   "09-26-83"
TRACED FROM   NASA DOCUMENT.....
              PARAGRAPH NUMBER "DIRECT_CONTROL_OMV_TEST"
TRACED FROM   FUNCTION DECOMPOSITION  CHECKOUT_OMV
ASSUMPTION   "The alternative that must be consistent with the
              philosophy of this example is alternative 3".
DATE UPDATED   "09-26-83"
TRACED TO   INTERFACE_FUNCTION_001
CHOICE
DATE CLOSED
```

DCDS: OPTION 3

```
FUNCTION COMPOSITION

DIRECT_CONTROL_TEST
DECOMPOSES DIRECT_CONTROL_TEST
DESCRIPTION   "The control of the OMV checkout requires Ground crew
              manual inputs, Ground display and controls and an SSIS
              function to translate commands to OMV format with SS crew
              involvement".
TAKEN FROM   NASA DOCUMENT.....
             PARAGRAPH NUMBER.....
TRACED FROM   DECISION  D_002
SUBGRAPH LAYER STANDARD_FLOW

BRANCH TRANSLATE_COMMANDS_TO_OMV_FORMAT
BRANCH ACCEPT_TRANSMIT_TEST_COMMANDS
BRANCH INPUT_MANUAL_TEST_COMMAND
END PARALLEL
END LAYER

ITEM TEST COMMANDS
     OUTPUT BY FUNCTION TRANSLATE_COMMANDS_TO_OMV_FORMAT
     INPUT BY FUNCTION INPUT_MANUAL_TEST_COMMANDS

FUNCTION TRANSLATE_COMMANDS_TO_OMV_FORMAT
     ALLOCATED TO SUBSYSTEM_SSIS
FUNCTION ACCEPT_TRANSMIT_TEST_COMMANDS
ALLOCATED TO SUBSYSTEM_GROUND_OPERATIONS
```

Figure A-1. Sequence of Steps in System Development (Continued)

STEP 5a

DCDS Viewpoint

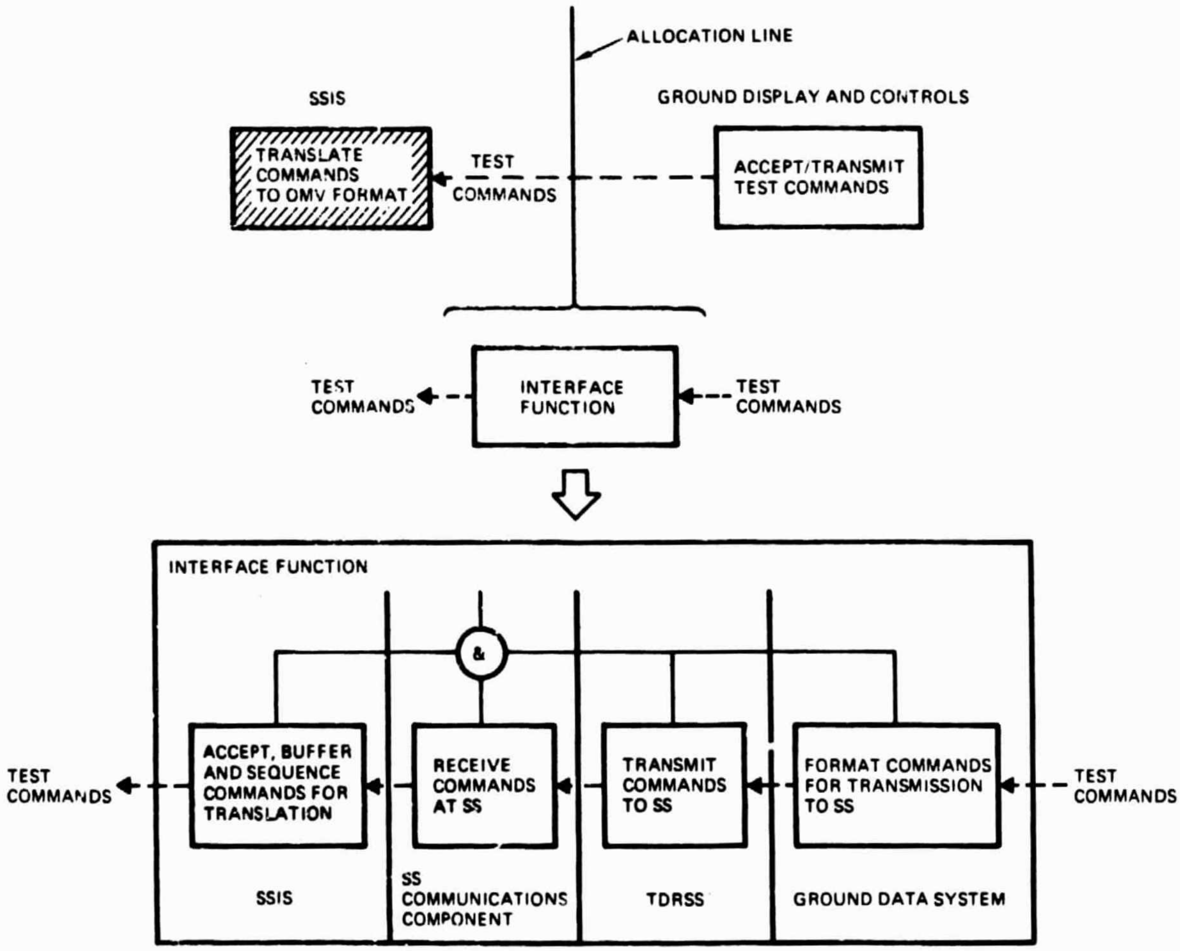
Step 5a: In this step, we define interface requirements. Each allocation line drawn across one of the expanded functional blocks cuts across the one or more of the dotted line input/output paths of the function. A given allocation line is expanded to permit a definition of an interface function which has precisely the inputs and outputs which cross the original allocation line. This function is itself allocated to the components of the system.

Space Station Application

Step 5a: Figure 5A is an illustration of the expansion of an allocation line into an interface function. We start with the situation portrayed in Figure C of Step 4 and expanded the allocation line between Translate Commands and Accept/Transmit Test Commands.

This complex option quickly reveals a need for involvement of a string of SS/Ground Ops components to support the interface called for. In the actual SSIS system design problem, the trade studies needed to determine which of Step 4's options should be pursued would take into account the complexities of this interface requirement.

FIGURE 5A



STEP 5b

Step 5b: The go
allocation proce
processor. Wher
processing will
alternatives will

Figure A-1. Sequence of Steps in System Development (Continued)

STEP 5b

DCDS Viewpoint

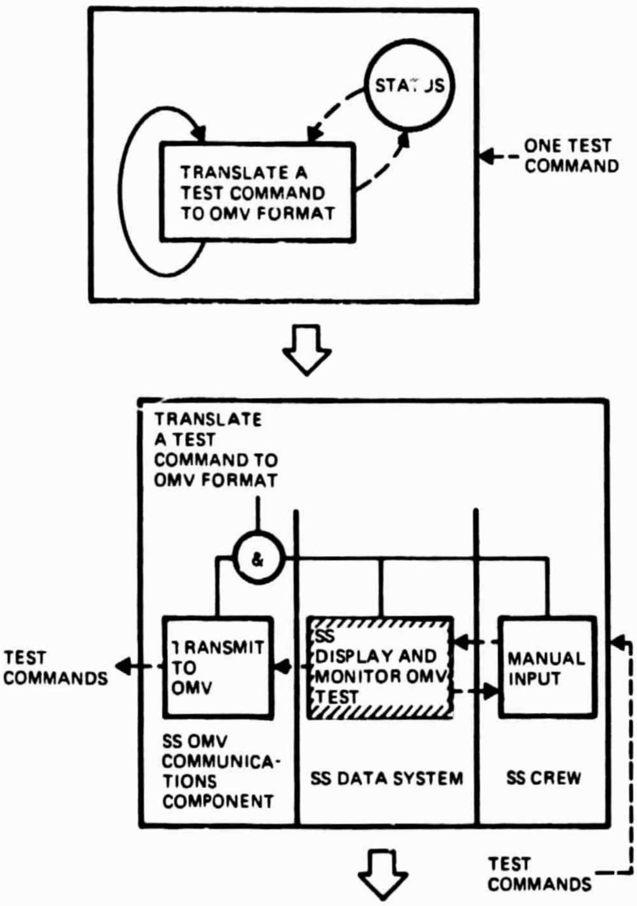
Step 5b: The goal of this step is to continue the decomposition and allocation process until the proper functions are allocated to the data processor. When this is accomplished, the requirements for data processing will be derived. Assumptions will be made along the way; alternatives will be documented and traceability will be managed.

Space Station Application

Step 5b: In step 5a we decomposed the interface function into four functions and allocated them to segments of the space station components. To continue this example we will decompose this function "Translate Test Command to OMV" and allocate the resulting functions to segments of the space station components; one of them being the Space Station Data System.

Figure 5B is an illustration of how the process results in a set of functional requirements for the Space Station data system to process OMV test commands. For this illustration it is assumed that the SS crew will monitor and evaluate the OMV test comm... which originated from the ground. The alternate option o... ...ing the SS crew in this process is recorded in the DCDS data ... as part of the list of trade studies.

FIGURE 5B



Step 5 inputs

```
FUNCTION COMPOSITION

TRANSLATE_COMMANDS_TO_OMV_FORMAT
DESCRIPTION    "This function requires SS crew involvement, data
               processing by the SS data system for display and monitor
               purposes and the capability of transmit transmit OMV test
               commands, one at a time with coordination."
TRACED FROM    DECISION D_003
TRACED FROM    NASA DOCUMENT .....
               PARAGRAPH NUMBER.....
INPUTS ITEM ONE_TEST_COMMAND
OUTPUTS ITEM ONE_TEST_COMMAND
SUBGRAPH LAYER STANDARD_PLAN
REPLICATE DOMAIN_SET NUMBER_OF_TEST_COMMANDS WITH_COORDINATION FUNCTION
               COORDINATE_ONLY_TEST_COMMANDS FUNCTION TRANSLATE_A_TEST_
               COMMAND_TO_OMV_FORMAT
     END REPLICATION
END

FUNCTION COMPOSITION TRANSLATE_A_TEST_COMMAND_TO_OMV_FORMAT


DECOMPOSES FUNCTION TRANSLATE_A_TEST_COMMAND_TO_OMV_FORMAT
DESCRIPTION    "This decomposition approach divides the function into an
               OMV communication component, SS display and monitor OMV
               test function to be performed by the SSDS and manual
               inputs by the SS crew".
TRACED FROM    DECISION D_003
SUBGRAPH LAYER STANDARD_FLOW


PARALLEL WITH COORDINATION FUNCTION
        BRANCH FUNCTION TRANSMIT_TO_OMV
        BRANCH FUNCTION SS_DISPLAY_AND_MONITOR_OMV_TEST
        BRANCH FUNCTION MANUAL_INPUT
     END PARALLEL
END LAYER

FUNCTION TRANSMIT_TO_OMV


ALLOCATED TO SUBSYSTEM SS_OMV_COMMUNICATIONS_COMPONENT

FUNCTION SS_DISPLAY_AND_MONITOR_OMV_TEST


ALLOCATED TO SUBSYSTEM SS_DATA_SYSTEM

FUNCTION MANUAL_INPUT


ALLOCATED TO SUBSYSTEM SS_CREW
```

FOLDOUT FRAME

Step 6

DCDS Viewpoint

Data Processing Requirements

Step 6. After the DCDS data base has been validated, the next phase, data processing requirements definition is initiated. This consists of identifying all of the interfaces to the data processor and deriving Requirements Networks (R-Nets in DCDS terminology) for each interface. In parallel, uniquely defined and testable, data processing requirements are documented in the DCDS data base.

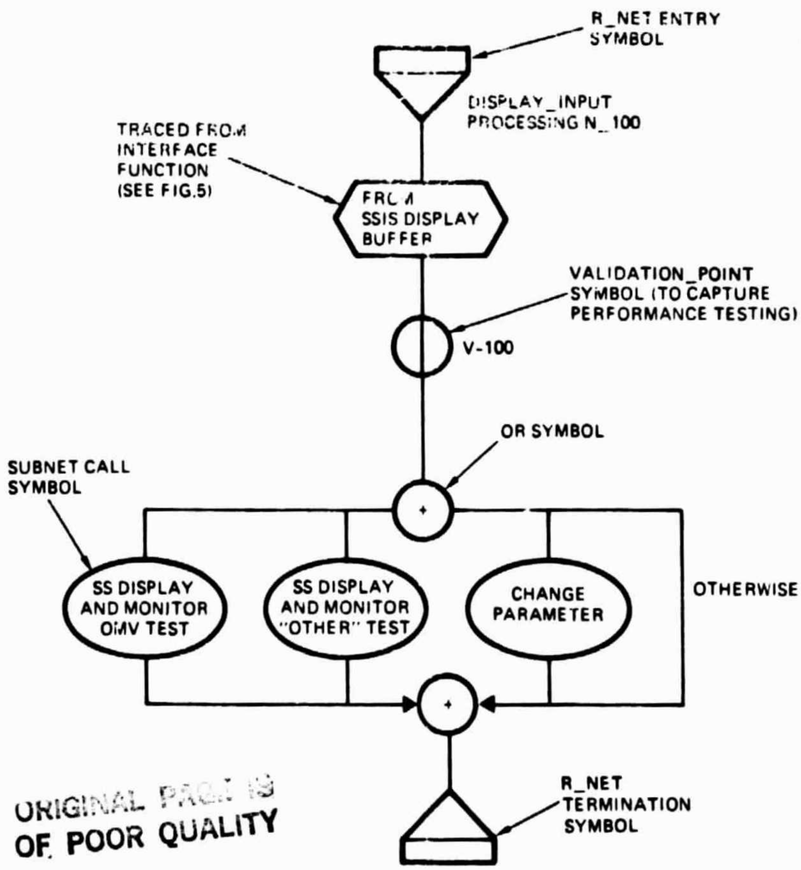
FIGURE 6. DECOMPOSE INTO DATA PROCESSING REQUIREMENTS

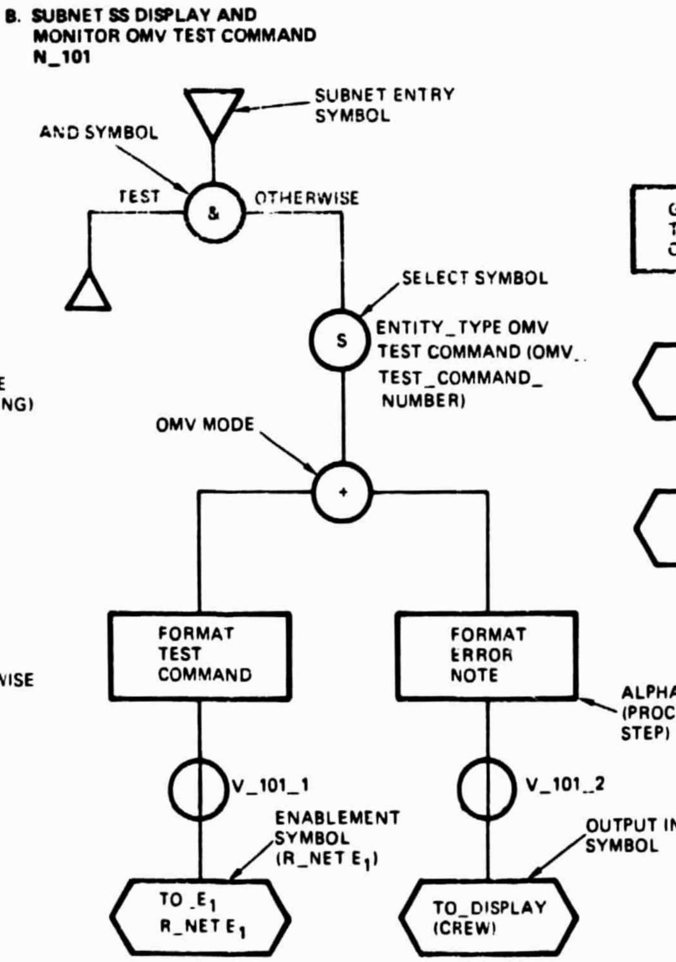
Space Station Application

SS Data System Requirements

Figure 6 represents an example of the data processing required to process one test command on board the space station with crew interaction. The representation is an R-Net which is input to the DCDS data base for further analysis. It represents a "thread" of processing steps to display and monitor each test command entered into the SS data system.

This set of requirements uniquely identified and traced to top level functions, interface designs, top level requirements is to be incorporated both in the requirements traceability management report and the SS data system requirements document. Note that at this stage of development, SS data system is a "black box" testable data processor, later to be segmented into the distributed topology (also part of the DCDS methodology).

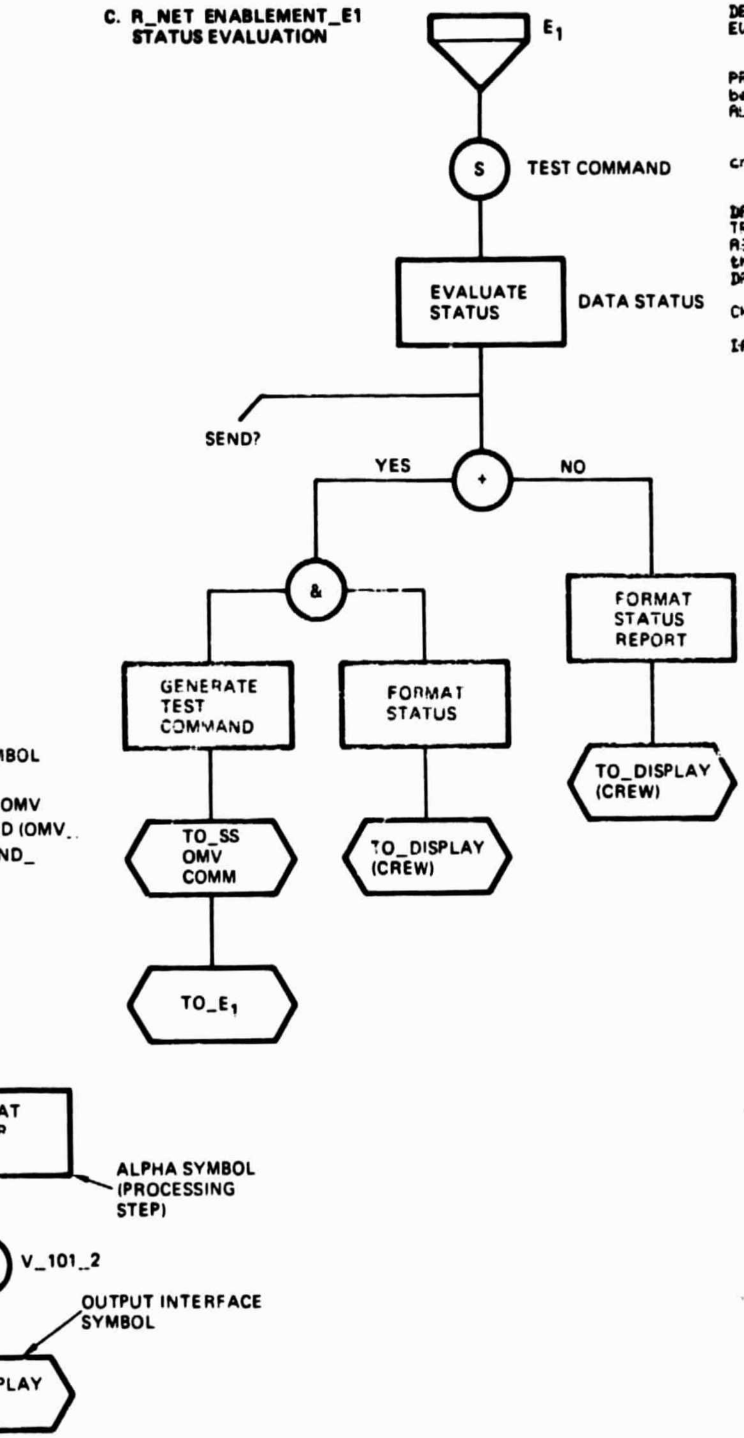




ORIGINAL PAGE IS OF POOR QUALITY

Figure A-1. Sequence of Steps in System Development (Continued)

FOLDOUT FRAME



Step 6 input

DCDS: DEFINE OPTIONS

```
DECISION   D_003
EQUATED TO

     SYNONYM   SS_DISPLAY_AND_MONITOR_OMV_TEST
PROBLEM   "Current requirements do not specify that the SS crew shall
be involved in OMV tests".
ALTERNATIVES
     1 -  OMV tests shall not be displayed to the SS crew
     2 -  OMV test commands shall be passed on automatically to the SS
crew at the console display  when required


DATE_OPENED    "09-26-83".
TRACED_FROM    NASA DOCUMENT---
ASSUMPTION     "The alternative most consistentwith the philosophy of
the example is alternative 2".
DATE_UPDATED   "09-26-83".

CHOICE

DATE_CLOSED
```

```
R_NET:   DISPLAY_INPUT_PROCESSINGS
EQUATED TO SYNONYM   N_100
DESCRIPTION   "This R_NET receives all inputs coming from the SSIS
consolue display buffer and processes them via SUBNETs."
ENTERED_BY   "A. MILLER."
STRUCTURE

     INPUT_INTERFACE  FROM_SSIS_DISPLAY_BUFFER

     VALIDATION_POINT  V_100

     IF (MESSAGE TYPE_IN = 200)
        SUBNET SS_DISPLAY_AND_MONITOR_OMV_TEST

     OR (MESSAGE TYPE_IN = 240)
        SUBNET SS_DISPLAY_AND_MONITOR_OTHER_TEST

     OR (MESSAGE TYPE_IN = 240)
        SUBNET CHANGE_PARAMETER

     OR (MESSAGE TYPE_IN = XXX)
        .
        .
        .

     OTHERWISE

     END

     TERMINATE
END.

SUBNET   SS_DISPLAY_AND_MONITOR_OMV_TEST
EQUATED TO SYNONYM   N_101
DESCRIPTION   "This subnet responds to an OMV TEST COMMAND".
TRACED FROM   NASA DOCUMENT ...
ENTERED_BY   A. MILLER
STRUCTURE

     DO

     SELECT ENTITY_TYPE_OMV_TEST_COMMAND

     SUCH THAT (TEST_COMMAND NUMBER= TEST_COMMAND_NUMBER_IN)

     IF (FOUND)
        ALPHA FORMAT_TEST_COMMAND_MESSAGE
        VALIDATION POINT V_101_1
        OUTPUT_INTERFACE TO_R_NET_E

     OTHERWISE
        ALPHA FORMAT_ERROR_NOTE
        VALIDATION POINT V_101-2
        OUTPUT INTERFACE TO CREW DISPLAY

     END

     AND

     RETURN

     END
END.
```

SUMMARY

When we reach Step 6, all functions including the interface functions are decomposed and allocated to the "black box" data system. The SSIS is defined in terms of functions allocated to SS ground components. The alloctions are traceable to top level functions, mission objectives, assumptions and decisions made during the process and the related trade studies performed during this study. Alternate approaches, open issues and decisions are documented in the DCDS data base. Completion checks are performed to see if all of the functions have the proper "connectivity."

Step 6 is the next phase of DCDS, the Data Processing Requirements Definition; it is explained on the opposite page. The validation that follows Step 6 uses DCDS's automated verification capability to support the "correctness of design" activity in SOW Task 4. A sample of the DCDS output for completeness checking is shown in figure A-2.

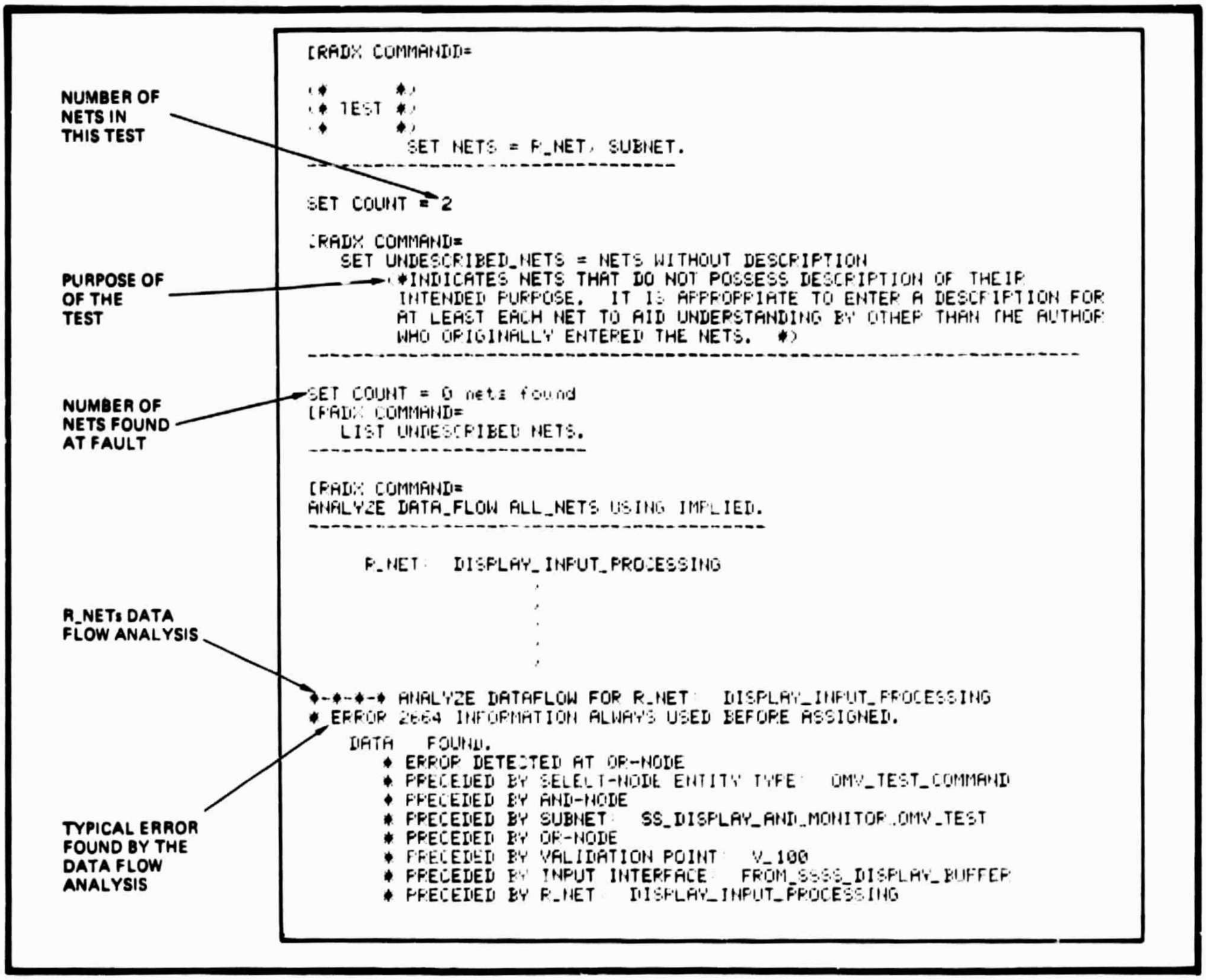


Figure A-2. Automated Checking Enforces Methodology Standards