DECEMBER 1985

NASA CR-177841

MDC H1343A

SSDS

MDAC

Ford IBM RCA

SPACE STATION DATA SYSTEM ANALYSIS/ARCHITECTURE STUDY

Task 2 – Options Development, DR-5
Volume III – Programmatic Options

(NASA-CR-177841) SPACE STATION DATA SYSTEM ANALYSIS/ARCHITECTURE STUDY. TASK 2: OPTIONS DEVELOPMENT, DR-5. VOLUME 3: PROGRAMMATIC OPTIONS (McDonnell-Douglas Astronautics Co.) 240 p HC A11/MF A01

N86-20477

Unclas
G3/18 04594

MCDONNELL DOUGLAS CORPORATION

SPACE STATION DATA SYSTEM ANALYSIS/ARCHITECTURE STUDY

Task 2 – Options Development, DR-5
Volume III – Programmatic Options

DECEMBER 1985

MDC H1343A
REPLACES MDC H1940
DATED MAY 1985

MCDONNELL DOUGLAS ASTRONAUTICS COMPANY-HUNTINGTON BEACH
5301 Bolsa Avenue Huntington Beach, California 92647 (714) 896-3311

PREFACE

The McDonnell Douglas Astronautics Company has been engaged in a Space Station Data System Analysis/Architecture Study for the National Aeronautics and Space Administration, Goddard Space Flight Center. This study, which emphasizes a system engineering design for a complete, end-to-end data system, is divided into six tasks:

Task 1. Functional Requirements Definition
Task 2. Options Development
Task 3. Trade Studies
Task 4. System Definition
Task 5. Program Plan
Task 6. Study Maintenance

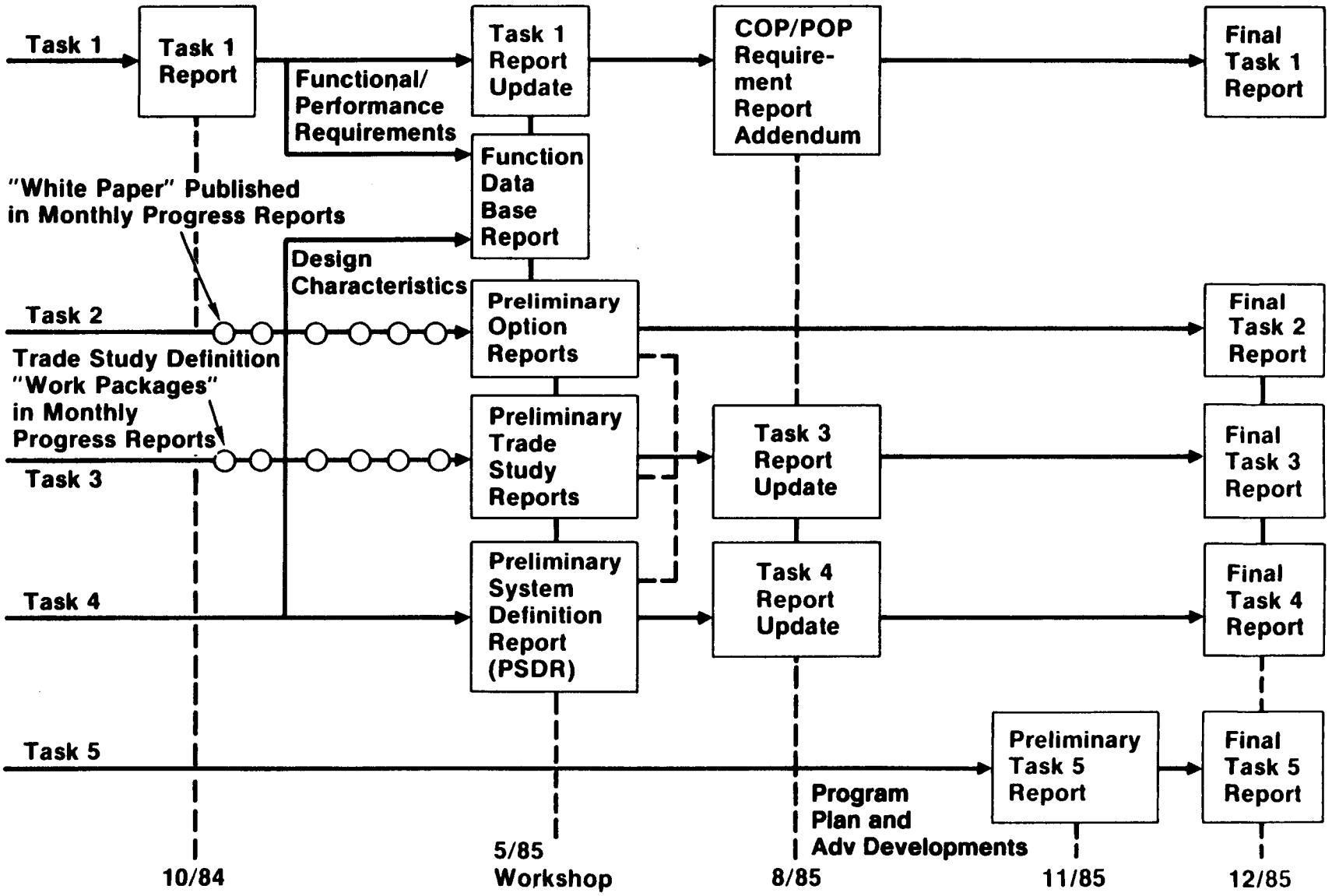
McDonnell Douglas was assisted by the Ford Aerospace and Communications Corporation and IBM Federal Systems Division.

This report was prepared for the National Aeronautics and Space Administration Goddard Space Flight Center under Control No. NAS5-28082.

Questions regarding this report should be directed to:

Glen P. Love
Study Manager
McDonnell Douglas Astronautics Company
Huntington Beach, CA 92647
(714) 896-2292

SSDS A/A DOCUMENTATION SCHEDULE



Task 2, Volume III
TABLE OF CONTENTS

GLOSSARY

A	Automatic
A&R	Automation and Robotics
A/A	Analysis/Architecture
A/D	Advanced Development
A/L	Airlock
A/N	Alphanumeric
AC&S	Attitude Control System
ACA	Attitude Control Assembly
ACO	Administrative Contracting Officer
ACS	Attitude Control and Stabilization
ACS/COM	Attitude Control System/Communications
ACTS	Advanced Communications Technology Satellite
AD	Ancillary Data
AD	Advanced Development
ADOP	Advanced Distributed Onboard Processor
ADP	Advanced Development Plan
AFOSR	Air Force Office of Scientific Research
AFP	Advanced Flexible Processor
AFRPL	Air Force Rocket Propulsion Laboratory
AGC	Automatic Gain Control
AGE	Attempt to Generalize
AI	Artificial Intelligence
AIE	Ada Integrated Environment
AIPS	Advanced Information Processing System
AL1	Air Lock One
ALS	Alternate Landing Site
ALS/N	Ada Language System/Navy
AMIC	Automated Management Information Center
ANSI	American National Standards Institute
AOS	Acquisition of Signal
AP	Automatic Programming
APD	Avalanche Photo Diode
APSE	Ada Programming Support Environment
ARC	Ames Research Center

ART	Automated Reasoning Tool
ASCII	American Standard Code for Information Exchange
ASE	Airborne Support Equipment
ASTROS	Advanced Star/Target Reference Optical Sensor
ATAC	Advanced Technology Advisory Committee
ATC	Air Traffic Control
ATP	Authority to Proceed
ATPS	Advanced Telemetry Processing System
ATS	Assembly Truss and Structure
AVMI	Automated Visual Maintenance Information
AWSI	Adoptive Wafer Scale Integration
B	Bridge
BARC	Block Adaptive Rate Controlled
BB	Breadboard
BER	Bit Error Rate
BIT	Built-in Test
BITE	Built-in Test Equipment
BIU	Buffer Interface Unit
BIU	Bus Interface Unit
BIU	Built-in Unit
BMD	Ballistic Missile Defense
BTU	British Thermal Unit
BW	Bandwidth
C	Constrained
C^2	Command and Control
C^3	Command, Control, and Communication
C^3I	Command, Control, Communication, and Intelligence
C&DH	Communications and Data Handling
C&T	Communication and Tracking Subsystem
C&T	Communications and Tracking
C&W	Control and Warning
C/L	Checklist
CA	Customer Accommodation
CAD	Computer-Aided Design
CAE	Computer-Aided Engineering
CAIS	Common APSE Interface Set
CAM	Computer-Aided Manufacturing

CAMAC	Computer Automatic Measurement and Control
CAP	Crew Activities Plan
CASB	Cost Accounting Standard Board
CASE	Common Application Service Elements
CATL	Controlled Aceptance Test Library
CBD	Commerce Business Daily
CBEMA	Computer and Business Equipment Manufacturing Association
CCA	Cluster Coding Algorithm
CCB	Contractor Control Board
CCB	Configuration Control Board
CCC	Change and Configuration Control
CCD	Charge-Coupled Device
CCITT	Consultive Committee for International Telegraph and Telephone
CCITT	Coordinating Committee for International Telephony and Telegraphy
CCMS	Checkout Control and Monitor System
CCR	Configuration Change Request
CCSDS	Consultative Committee for Space Data System
CCTV	Closed-Circuit Television
cd/M^2	Candelas per square Meter
CDG	Concept Development Group
CDMA	Code Division Multiple Access
CDOS	Customer Data Operations System
CDR	Critical Design Review
CDS	Control Data Subsystem
CE	Conducted Emission
CEI	Contract End-Item
CER	Cost Estimating Relationship
CFR	Code of Federal Regulations
CFS	Cambridge File Server
CG	Center of Gravity
CIE	Customer Interface Element
CIL	Critical Item List
CIU	Customer Interface Unit
CLAN	Core Local Area Network
CM	Configuration Management
CM	Center of Mass
CMDB	Configuration Management Data Base

CMG	Control Moment Gyro
CMOS	Complementary Metal-Oxide Semiconductor
CMS	Customer Mission Specialist
CMU	Carnegie-Mellon University
CO	Contracting Officer
COF	Component Origination Form
COL	Controlled Operations Library
COMM	Commercial Missions
COP	Co-orbital Platform
COPCC	Coorbit Platform Control Center
COPOCC	COP Operations Control Center
COTS	Commercial Off-the-Shelf Software
CPCI	Computer Program Configuration Item
CPU	Central Processing Unit
CQL	Channel Queue Limit
CR	Compression Ratio
CR	Change Request
CR&D	Contract Research and Development
CRC	Cyclic Redundancy Checks
CRF	Change Request Form
CRSS	Customer Requirements for Standard Services
CRT	Cathode Ray Tube
CS	Conducted Susceptibility
CSD	Contract Start Date
CSDL	Charles Stark Draper Laboratory
CSMA/CD/TS	Carrier-Sense Multiple with Access/Collision Detection and Time Slots
CSTL	Controlled System Test Library
CTA	Computer Technology Associates
CTE	Coefficient of Thermal Expansion
CUI	Common Usage Item
CVSD	Code Variable Slope Delta (Modulation)
CWG	Commonality Working Group
D&B	Docking and Berthing
DADS	Digital Audio Distribution System
DAIS	Digital Avionics Integration System
DAR	Defense Acquisition Regulation

DARPA	Defense Advanced Research Projects Agency
DB	Data Base
DBA	Data Base Administrator
DBML	Data Base Manipulation Language
DBMS	Data Base Management System
DCAS	Defense Contract Administrative Services
DCDS	Distributed Computer Design System
DCR	Data Change Request
DDBM	Distributed Data Base Management
DDC	Discipline Data Center
DDT&E	Design, Development, Testing, and Engineering
DEC	Digital Equipment Corp.
DES	Data Encryption Standard
DFD	Data Flow Diagram
DGE	Display Generation Equipment
DHC	Data Handling Center
DID	Data Item Description
DIF	Data Interchange Format
DMA	Direct Memory Access
DMS	Data Management System
DoD	Department of Defense
DOMSAT	Domestic Communications Satellite System
DOS	Distributed Operating System
DOT	Department of Transportation
DPCM	Differential Pulse Code Modulation
DPS	Data Processing System
DR	Discrepancy Report
DR	Data Requirement
DRAM	Dynamic Random-Access Memory
DRD	Design Requirement Document
DS&T	Development Simulation and Training
DSDB	Distributed System Data Base
DSDL	Data Storage Description Language
DSDS	Data System Dynamic Simulation
DSIT	Development, Simulation, Integration and Training
DSN	Deep-Space Network
DTC	Design to Cost

DTC/LCC	Design to Cost/Life Cycle Cost
DTG	Design To Grow
E/R	Entity/Relationship
EADI	Electronic Attitude Direction Indicator
ECC	Error Correction Codes
ECLSS	Environmental Control and Life-Support System
ECMA	European Computers Manufacturing Assoc.
ECP	Engineering Change Proposals
ECS	Environmental Control System
EDF	Engineering Data Function
EEE	Electrical, Electronic, and Electromechanical
EHF	Extremely High Frequency
EHSI	Electronic Horizontal Situation Indicator
EIA	Electronic Industry Association
EL	Electroluminescent
EM	Electromagnetic
EMC	Electromagnetic Compatibility
EMCFA	Electromagnetic Compatibility Frequency Analysis
EME	Earth Mean Equator
EMI	Electromagnetic Interference
EMR	Executive Management Review
EMS	Engineering Master Schedule
EMU	Extravehicular Mobility Unit
EMUDS	Extravehicular Maneuvering Unit Decontamination System
EO	Electro-optic
EOL	End of Life
EOS	Earth Observing System
EPA	Environmental Protection Agency
EPS	Electrical Power System
ERBE	Earth Radiation Budget Experiment
ERRP	Equipment Replacement and Refurbishing Plan
ESR	Engineering Support Request
ESTL	Electronic Systems Test Laboratory
EVA	Extravehicular Activity
F/T	Fault Tolerant
FACC	Ford Aerospace and Communications Corporation
FADS	Functionally Automated Database System

FAR	Federal Acquisition Regulation
FCA	Functional Configuration Audit
FCOS	Flight Computer Operating System
FCR	Flight Control Rooms
FDDI	Fiber Distributed Data Interface
FDF	Flight Dynamics Facility
FDMA	Frequency-Division Multiple Access
FEID	Flight Equipment Interface Device
FETMOS	Floating Gate Election Tunneling Metal Oxide Semiconductor
FF	Free Flier
FFT	Fast Fourier Transform
FIFO	First in First Out
FIPS	Federal Information Processing Standards
fl	foot lambert - Unit of Illumination
FM	Facility Management
FMEA	Failure Modes and Effects Analysis
FMECA	Failure Mode Effects and Criticality Analysis
FO	Fiber-Optics
FO/FS/R	Fail-Operational/Fail Safe/Restorable
FOC	Fiber-Optic Cable
FODB	Fiber-Optic Data Bus
FODS	Fiber Optic Demonstration System
FPR	Federal Procurement Regulation
FQR	Formal Qualification Review
FSD	Full-Scale Development
FSE	Flight Support Equipment
FSED	Full Scale Engineering Development
FSIM	Functional Simulator
FSW	Flight Software
FTA	Fault Tree Analysis
FTMP	Fault Tolerant Multi-Processor
FTSC	Fault Tolerant Space Computer
GaAs	Gallium Arsenide
GaAsP	Gallium Arsenic Phosphorus
GaInP	Gallium Indium Phosphorus
GaP	Gallium Phosphorous
GAPP	Geometric Arithmetic Parallel Processor

Gbps	Gigabits Per Second
GBSS	Ground Based Support System
GEO	Geosynchronous Earth Orbit
GEP	Gas Election Phosphor
GFC	Ground Forward Commands
GFE	Government-Furnished Equipment
GFP	Government-Furnished Property
GFY	Government Fiscal Year
GIDEP	Government/Industry Data Exchange Program
GMM	Geometric Math Model
GMS	Geostationary Meteorological Satellite
GMT	Greenwich Mean Time
GMW	Generic Maintenance Work Station
GN&C	Guidance, Navigation, and Control
GPC	General-Purpose Computer
GPP	General-Purpose Processor
GPS	Global Positioning System
GRO	Gamma Ray Observatory
GSC	Ground Service Center
GSE	ground Support Equipment
GSFC	(Robert H.) Goddard Space Flight Center
GTOSS	Generalized Tethered Object System Simulation
H/W	Hardware
HAL	High-Order Algorithmic Language
HDDR	Help Desk Discrepancy Report
HDDR	High Density Digital Recording
HEP	Heterogeneous Element Processor
HFE	Human Factors Engineering
HIPO	Hierarchical Input Process Output
HIRIS	High Resolution Imaging Spectrometer
HM1	Habitation Module One
HM	Habitation Module
HOL	High Order Language
HOS	High Order Systems
HPP	High Performance Processors
HRIS	High Resolution Imaging Spectrometer
I	Interactive

I/F	Interface
I/O	Input/Output
IBM	IBM Corporation
IC	Intercomputer
ICAM	Integrated Computer-Aided Manufacturing
ICB	Internal Contractor Board
ICD	Interface Control Document
ICOT	Institute (for new generation) Computer Technology
ICS	Interpretive Computer Simulation
ID	Interface Diagram
ID	Identification
IDM	Intelligent Database Machine
IDMS	Information and Data Management System
IEEE	Institute of Electrical and Electronic Engineers
IEMU	Integrated Extravehicular Mobility Unit
IF	Intermediate Frequency
IFIPS	International Federation of Industrial Processes Society
ILD	Injector Laser Diode
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
IOC	Initial Operating Capability
IOP	Input/Output Processor
IPCF	Interprocess Communications Facility
IPC	Interprocesses Communication
IPL	Initial Program Load
IPR	Internal Problem Report
IPS	Instrument Pointing System
IR	Infrared
IR&D	Independent Research and Development
IRN	Interface Revision Notices
ISA	Inertial Sensor Assembly
ISA	Instruction Set Architecture
ISDN	Integration Services Digital Network
ISO	International Standards Organization
ITAC-O	Integration Trades and Analysis-Cycle O
ITT	International Telegraph and Telephone
IV&V	Independent Validation and Verification

IVA	Intravehicular Activity
IWS	Intelligent Work Station
JPL	Jet Propulsion Laboratory
JSC	(Lyndon B.) Johnson Space Center
KAPSE	Kernal APSE
KEE	Knowledge Engineering Environment
KIPS	Knowledge Information Processing System
KOPS	Thousands of Operations Per Second
KSA	Ku-band, Single Access
KSC	(John F.) Kennedy Space Center
Kbps	Kilobits per second
Kipc	Thousand instructions per cycle
LAN	Local-Area Network
LaRC	Langley Research Center
LCC	Life-Cycle Cost
LCD	Liquid Crystal Display
LDEF	Long-Duration Exposure Facility
LDR	Large Deployable Reflector
LED	Light-Emitting Diode
LEO	Low Earth Orbit
LeRC	Lewis Research Center
LIDAR	Laser-Instrument Distance and Range
LIFO	Last In First Out
LIPS	Logical Inferences Per Second
LISP	List Processor
LisP	List Processor
LLC	Logical Link Control
LMI	LISP Machine Inc.
LN_2	Liquid Nitrogen
LNA	Low-noise Amplifier
LOE	Level of Effort
LOE	Low-earth Orbit Environments
LOS	Loss of Signal
LPC	Linear Predictive Coding
LPS	Launch Processing System
LRU	Line-Replaceable unit
LSA	Logistic Support Analysis

LSAR	Logistic Support Analysis Report
LSE	Language Sensity Editors
LSI	Large-scale Integration
LTV	LTV Aerospace and Defense Company, Vought Missiles Advanced Programs Division
LZPF	Level 0 Processing Facility
M	Manual
µP	Microprocessor
MA	Multiple Access
MA	Managing Activity
MAPSE	Minimum APSE
Mbps	Million Bits Per Second
MBPS	Million Bits Per Second
MCAIR	McDonnell Aircraft Company
MCC	Mission Control Center
MCC	Microelectronics and Computer Technology Corp.
MCDS	Management Communications and Data System
MCM	Military Computer Modules
MCNIU	Multi-compatible Network Interface Unit
MDAC-HB	McDonnell Douglas Astronautics Company-Huntington Beach
MDAC-STL	McDonnell Douglas Astronautics Company-St. Louis
MDB	Master Data Base
MDC	McDonnell Douglas Corporation
MDMC	McDonnell Douglas Microelectronics Center
MDRL	McDonnell Douglas Research Laboratory
MFLOP	Million Floating Point Operations
MHz	Million Hertz
MIMO	Multiple-Input Multiple-Output
MIPS	Million (machine) Instructions Per Second
MIT	Massachusetts Institute of Technology
MITT	Ministry of International Trade and Industry
MLA	Multispectral Linear Array
MMI	Man Machine Interface
MMPF	Microgravity and Materials Process Facility
MMS	Module Management System
MMS	Momentum Management System
MMU	Mass Memory Unit

MMU	Manned Maneuvering Unit
MNOS	Metal-Nitride Oxide Semiconductor
MOC	Mission Operations Center
MOI	Moment of Inertia
MOL	Manned Orbiting Laboratory
MOS	Metal Oxide Semiconductor
MPAC	Multipurpose Application Console
MPS	Materials, Processing in Space
MPSR	Multi-purpose Support Rooms
MRMS	Mobile Remote Manipulator System
MRWG	Mission Requirements Working Group
MSFC	(George C.) Marshall Space Flight Center
MSI	Medium-Scale Integration
MSS	Multispectral Scanner
MTA	Man-Tended Approach
MTBF	Mean Time Between Failures
MTTR	Mean Time to Repair
MTU	Master Timing Unit
NASA	National Aeronautics and Space Administration
NASCOM	NASA Communications Network
NASPR	NASA Procurement Regulation
NBO	NASA Baseline
NBS	National Bureau of Standards
NCC	Network Control Center
NFSD	NASA FAR Supplement Directive
NGT	NASA Ground Terminals
NHB	NASA Handbook
NISDN	NASA Integrated System Data Network
NIU	Network Interface Unit
NL	National Language
NLPQ	National Language for Queuing Simulation
NMI	NASA Management Instruction
NMOS	N-Channel Metal-Oxide Semiconductor
NMR	N-Modular Redundant
NOS	Network Operating System
NS	Nassi-Schneidermann
NSA	National Security Administration

NSF	National Science Foundation
NSTS	National Space Transportation System
NTDS	Navy Tactical Data System
NTE	Not To Exceed
NTRL	NASA Technology Readiness Level
NTSC	National Television Standards Committee
Nd:YAG	Neodynium Yttrium Aluminum Garnet (laser type)
O&M	Operations and Maintenance
O/B	Onboard
OASCB	Orbiter Avionics Software Control Board
OCN	Operations and Control Network, Operational Control Networks
ODB	Operational Data Base
ODBMS	Onboard Data Base Management System
OEL	Operating Events List
OES	Operating Events Schedule
OID	Operations Instrumentation Data
OLTP	On Line Transaction Processing
OMCC	Operations Management and Control Center
OMV	Orbital Maneuvering Vehicle
ONR	Office of Naval Research
ORU	Orbital Replacement Unit
OS	Operating System
OSE	Orbit Support Equipment
OSI	Open Systems Interconnect
OSM	Orbital Service Module
OSSA	Office of Space Science and Applications
OSTA	Office of Space and Terrestrial Application
OSTDS	Office of Space Tracking and Data Systems
OTV	Orbital Transfer Vehicle
P&SA	Payload and Servicing Accommodations
P/L	Payload
PA	Product Assurance
PAM	Payload Assist Module
PASS	Primary Avionics Shuttle Software
PBX	Private Branch Exchange
PC	Personal Computer
PCA	Physical Configuration Audit

PCA	Program Change Authorization
PCM	Pulse Code Modulation
PCR	Program Change Request
PDP	Plazma Display Panel
PDR	Preliminary Design Review
PDRD	Program Definition and Requirements Document
PDRSS	Payload Deployment and Retrieval System Simulation
PILS	Payload Integration Library System
PIN	Personal Identification Number
PLA	Programmable Logic Array
PLAN	Payload Local Area Network
PLSS	Payload Support Structure
PMAD	Power Management and Distribution
PMC	Permanently Manned Configuration
PN	Pseudonoise
POCC	Payload Operations Control Center
POP	Polar Orbiter Platform
POPCC	Polar Orbit Platform Control Center
POPOCC	POP Operations Control Center
PRISM	Prototype Inference System
PSA	Problem Statement Analyzer
PSA	Preliminary Safety Analysis
PSCN	Program Support Communications Network
PSL	Problem Statement Language
PTR	Problem Trouble Report
QA	Quality Assurance
R	Restricted
R&D	Research and Development
R&QA	Reliability and Quality Assurance
R/M/A	Reliability/Maintainability/Availability
R/T	Real Time
RAD	Unit of Radiation
RAM	Random Access Memory
RAP	Relational Associative Processor
RC	Ring Concentrator
RCA	RCA Corporation
RCS	Reaction Control System

RDB	relational Data Base
RDC	Regional Data Center
REM	Roentgen Equivalent (man)
RF	Radio Frequency
RFC	Regenerative Fuel Cell
RFI	Radio Frequency Interference
RFP	Request for Proposal
RGB	Red-Green-Blue
RID	Review Item Disposition
RID	Revision Item Description
RISC	Reduced Instruction Set Computer
RMS	Remote Manipulator System
RMSE	Root Mean Square Error
RNET	Reconfiguration Network
ROM	Read Only Memory
ROTV	Reuseable Orbit Transfer Vehicle
RPMS	Resource Planning and Management System
RS	Reed-Solomon
RSA	Rivest, Skamir and Adleman (encryption method)
RTX	Real Time Execution
S&E	Sensor and Effector
S/C	Spacecraft
S/W	Software
SA	Single Access
SA	Structured Analysis
SAAX	Science and Technology Mission
SAE	Society of Automotive Engineers
SAIL	Shuttle Avionics Integration Laboratory
SAIS	Science and Applications Information System
SAR	Synthetic Aperture Radar
SAS	Software Approval Sheet
SASE	Specific Application Service Elements
SATS	Station Accommodations Test Set
SBC	Single Board Computer
SC	Simulation Center
SCR	Software Change Request
SCR	Solar Cosmic Ray

SCS	Standard Customer Services
SDC	Systems Development Corporation
SDP	Subsystem Data Processor
SDR	System Design Review
SDTN	Space and Data Tracking Network
SE&I	Systems Engineering and Integration
SEI	Software Engineering Institute
SESAC	Space and Earth Scientific Advisory Committee
SESR	Sustaining Engineering System Improvement Request
SESS	Software Engineering Standard Subcommittee
SEU	Single Event Upset
SFDU	Standard Format Data Unit
SI	International System of Units
SIB	Simulation Interface Buffer
SIFT	Software Implemented Fault Tolerance
SIMP	Single Instruction Multi-Processor
SIRTF	Shuttle Infrared Telescope Facility
SLOC	Source Lines of Code
SMC	Standards Management Committee
SMT	Station Management
SNA	System Network Architecture
SNOS	Silicon Nitride Oxide Semiconductor
SNR	Signal to Noise Ratio
SOA	State Of Art
SOPC	Shuttle Operations and Planning Complex
SOS	Silicon On Saphire
SOW	Statement of Work
SPC	Stored Payload Commands
SPF	Software Production Facility
SPF	Single-Point Failure
SPR	Spacelab Problem Reports
SPR	Software Problem Report
SQA	Software Quality Assurance
SQAM	Software Quality Assessment and Measurement
SQL/DS	SEQUEL Data System
SRA	Support Requirements Analysis
SRAM	Static Random Access Memory

SRB	Software Review Board
SRC	Specimen Research Centrifuge
SREM	Software Requirements Engineering Methodology
SRI	Stanford Research Institute
SRM&QA	Safety, Reliability, Maintainability, and Quality Assurance
SRMS	Shuttle Remote Manipulator System
SRR	System Requirements Review
SS	Space Station
SSA	Structural Systems Analysis
SSA	S-band Single Access
SSCB	Space Station Control Board
SSCC	Station Station Communication Center
SSCR	Support Software Change Request
SSCS	Space Station communication system
SSCTS	Space Station communications and tracking system
SSDMS	Space Station data management system
SSDR	Support Software Discrepancy Report
SSDS	Space Station data system
SSE	Software Support Environment
SSEF	Software Support Environment Facility
SSIS	Space Station Information System
SSME	Space Shuttle Main Engine
SSO	Source Selection Official
SSOCC	Space Station Operations Control System
SSOCC	Space Station Operations Control Center
SSOL	Space Station Operation Language
SSON	Spacelab Software Operational Notes
SSOS	Space Station Operating System
SSP	Space Station Program
SSPE	Space Station Program Element
SSPO	Space Station Program Office
SSSC	Space Station Standard Computer
SSST	Space Station System Trainer
STAR	Self Test and Recovery (repair)
STARS	Software Technology for Adaptable and Reliable Software
STDN	Standard Number
STI	Standard Technical Institute

STO	Solar Terrestrial Observatory
STS	Space Transportation System
SUSS	Shuttle Upper Stage Systems
SYSREM	System Requirements Engineering Methodology
Si	Silicon
SubACS	Submarine Advanced Combat System
TAI	International Atomic Time
TBD	To Be Determined
TBU	Telemetry Buffer Unit
TC	Telecommand
TCP	Transmissions Control Protocols
TCS	Thermal Control System
TDASS	Tracking and Data Acquisition Satellite System
TDM	Technology Development Mission
TDMA	Time-Division Multiple Access
TDRS	Tracking and Data Relay Satellite
TDRSS	Tracking and Data Relay Satellite System
TFEL	Thin Film Electrolumenescent
THURIS	The Human Role in Space (study)
TI	Texas Instruments
TM	Technical Manual
TM	Thematic Mapper
TMDE	Test, Measurement, and Diagnostic Equipment
TMIS	Technical and Management Information System
TMP	Triple Multi-Processor
TMR	Triple Modular Redundancy
TMS	Thermal Management System
TPWG	Test Planning Working Group
TR	Technical Requirement
TRAC	Texas Reconfigurable Array Computer
TRIC	Transition Radiation and Ionization Calorimeter
TSC	Trade Study Control
TSIP	Technical Study Implementation Plan
TSP	Twisted Shielded Pair
TSS	Tethered Satellite System
TT&C	Telemetry, Tracking, and Communications
TTC	Telemetry Traffic Control

TTR	Timed Token Ring
TWT	Traveling-Wave Tube
U	Non-restrictive
UCC	Uniform Commercial Code
UDRE	User Design Review and Exercise
UIL	User Interface Language
UON	Unique Object Names
UPS	Uninterrupted Power Source
URN	Unique Record Name
UTBUN	Unique Telemetry Buffer Unit Name
UTC	Universal Coordinated Time
V&V	Validation and Verification
VAFB	Vandenberg Air Force Base
VAX	Virtual Address Exchange
VHSIC	Very High-Speed Integrated Circuit
VLSI	Very Large-Scale Integration
VLSIC	Very Large-Scale Integrated Circuit
VV&T	Validation, Verification and Testing
WAN	Wide Area Network
WBS	Work Breakdown Structure
WBSP	Wideband Signal Processor
WDM	Wavelength Division Multiplexing
WP	Work Package
WRO	Work Release Order
WS	Workstation
WSGT	White Sands Ground Terminal
WTR	Western Test Range
XDFS	XEROX Distributed File System
YAPS	Yet Another Production System
ZOE	Zone Of Exclusion
ZONC	Zone Of Non-Contact
ZnS	Zinc Sulfide

Volume III

TASK 2 - OPTIONS DEVELOPMENT

3.0 PROGRAMMATIC OPTIONS

SUMMARY

This volume contains the options development for the Programmatic Options Category. The specific programmatic options and their respective volume III section numbers are as follows:

- 3.0 PROGRAMMATIC OPTIONS
  - 3.1 Standardization/Commonality
  - 3.2 System Management
  - * 3.3 Deleted
  - * 3.4 Deleted
  - 3.5 System Development
    - 3.5.1 Hardware Procurement
    - 3.5.2 Software Development
    - 3.5.3 System Integration Test & Verification

For the Options Development general approach and methodology the reader is referred to the introductory sections of Task 2, Options Development, Volume I.

* These items have been deleted or incorporated into other sections.

3.0 PROGRAMMATIC OPTIONS

3.1 STANDARDIZATION/COMMONALITY OPTIONS

The purpose of this report is to describe options associated with standardization and commonality. Many aspects of the Space Station Data System are subject to standards. Table 3.1-1 lists those that are discussed in this report.

TABLE 3.1-1

COVERED TOPICS

Communciation
Data Base Management Systems
Coding
Quality Assurance Software
Quality Assurance Hardware
Hardware
Safety
Reliability
Maintainability
Logistics
Procurement
Video
Audio

The Space Station Definition and Preliminary Design Request for Proposal, August 20, 1984, specifies several requirements among which are:

> "The need for commonality of the system will be a key driver in the Space Station Program design. The long life on orbit, the need for maintainability,...all dictate that the different onboard systems and subsystems need to be minimized."
>
> "The indefinite life of the Space Station Program imposes the requirement for management of changing technology."

In a real sense, these requirements conflict as standardization, implied by the former requirement, will in time be overrun by technology change. A very important concern about standardization is the question of how to balance the need for standard hardware and software elements with the requirement to accommodate new technology. The use of "Standards" can lead to the continued use of obsolete technology if standards are not carefully planned to allow for cost-effective insertion of new technology. Future obsolescence of existing technology and the need for infusion of new technology represents potential conflicts.

The following sections discuss the topics appearing in Table 3.1-1.

3.1.1 Communications ProtocolInterface Standards

This section describes the options for standardization of communications interfaces in the Space Station Program. The options are described for both space and ground, and the International Standards Organization's (ISO) model for Open Systems Interconnection (OSI) is used.

A comprehensive review of communications standards is beyond the scope of this paper. The purpose is to:

- o restrict discussion to options related to the Space Station Program, using the results of Task 1 to focus options
- o describe options for the end-to-end standards architecture
- o overview requirements for standardization within each ISO/OSI level
- o provide references to detailed descriptions of the standards.

The first part of this section (Section 3.1.1.1) provides an overview. The second section (Section 3.1.1.2) discusses each ISO layer in detail and provides SSDS requirements and candidate standards options for the layer.

Much interest has been expressed in how standards resulting from the ISO model, and emerging standards for packet telemetry, might impact the SSDS. The third part of the paper (Section 3.1.1.3) thus concentrates on discussing options for how telemetry and telecommand standards (from the Consultating Committee for Space Data Systems) and network standards (consistent with the ISO/OSI model) might be implemented in an end-to-end architecture for the SSDS.

3.1.1.1 Communications Standards Options Overview

The SSIS will include:

- o on-board Data Management System including local area communications networks,
- o TDRSS links between space and ground,
- o a ground wide area communication network, and
- o local area networks on the ground, at NASA and non-NASA facilities.

Communication standards will be selected for each of the interfaces and links for these networks. This section uses the ISO/OSI model to describe communications functions and standards

options for these networks and links. In addition to providing a basis for describing standards and their functions, various organizations (ISO, CCITT, etc.) are developing standards that are compatible with the ISO/OSI model.

The ISO/OSI model is a natural choice since (a) it is comprehensive (b) it has international acceptance and (c) it is expected that many vendors will be producing equipment compatible with these standards. The fact that a very large base of worldwide support is expected to exist outside of NASA for OSI standards will likely demand lower prices, off-the-shelf hardware and software, and extendable services and networks.

Standards needed by Space Station have also been proposed by other organizations. Specifically, the Consultive Committee for Space Standards has developed standards for:

- Telemetry Packets, Frames, Channel Coding
- Telecommand packets, Frames, Channel Coding
- RF & Modulation
- Time Codes

Specific requirements with respect to standards, as reflected in Section 5.3.8.9 of the Task 1 report, are illustrated on the next page.

Requirements from Task 1 Report

"The SSDS shall provide standardized language, protocol, format, and transmission rates for all SSDS and all SSDS subsystems."

"As a first preference, customer interface standards shall be defined in accordance with the International Standards Organization (ISO) sevem layer model for Open Systems Interconnect (OSI)."

"The SSDS shall use, for each of the seven layers, existing internationally accepted standards as a first priority followed by new standards development (within the OSI model framework)."

"The customer interfaces defined within the first three layers of the OSI model shall conform to standards defined and controlled by such sources as:

NBS, National Bureau of Standards

ANSI, American National Standards Institute

ECMA, European Computer Manufacturing Association

CCITT, Consultative Committee for International Telegraph and Telephone

EIA, Electronic Industry Association

CCSDS, Consultative Committee for Space Data Systems

IEEE, Institute of Electrical & Electronic Engineers

"When practical, appropriate standards from these sources shall be used at higher layers of the OSI model."

"The SSIS/SCS (per Figure 1-2, includes SSDS and non-SSDS elements) shall obtain and/or develop standards for customer interfaces in areas such as software, critical/limited payload health and safety monitoring, man-machine interfaces, command generation, time code, attitude and position data, pointing coordinate systems, data base management systems, graphics displays, data handling/archiving/distribution, documentation, configuration control, cost accounting, data system requirements definition, operations audit trail, etc. When new customer standards are proposed, the SSIS/SCS shall present these standards to a customer panel which will provide an impact statement on behalf of all customers."

The selection of standards approaches gain importance when viewed as broad requirements of the Space Station Program:

- o Migration - functions are expected to migrate from the ground to on-board over time. Thus, initially one would expect an on-board process to be communicating with the ground, and later with the same process implemented on-board. It will be critical to be able to perform this migration while minimizing the impacts on the existing on-board software. Layering, and a flexible set of services as reflected in the choice of standards, are essential to support this.

- o High Data Rates - suggest the need to handle and transport data in as automated a fashion as possible, with a minimum of re-configuration.

Standards can be a significant aid in reducing costs and ensuring that customers are well served. For example, one would not want to require customers to use unusual communications equipment or to receive their data in different formats depending on the source. There are also dangers to standardization. For example, one can get "locked in" to a specific technology. Thus, a careful selection must be made of where and when to standardize, based on the requirements.

3.1.1.1.1 ISO/OSI Model description

The ISO/OSI model identifies seven layers which correspond to levels of abstraction, each layer performing a well-defined set of functions. The layers of the model are described in Table 1.

Application	Provides routines specific to an application. Allows applications on different nodes to communicate.
Presentation	Performs such services as text compression, file formating,
Session	Responsible for establishing and managing connections between two processes and for recovery of failures at the transport layer. Standards at this level also provide for authentification, billing, and agreement on parameters in effect for a session.
Transport	Accepts data in message form from session layer, breaks it down into smaller pieces if required by network, and vice versa. Standards at this level also proide for host-host connection & flow control.

ISO/OSI Layers
Table 1

Network	Provides for routing and congestion control. Standards at this level allow routing, sequenced delivery congestion control, and accounting functions for data transfers between a computer and a network.
Data Link	Responsible for providing an error free line. Performs framing, error detection, sequencing, time out and acknowledgement, and flow control.
Physical	Responsible for data transmission over a physical interface.

Table 1 (Continued)

Systems which are implemented with standards consistent with OSI are "open" in that different systems, from different manufacturers, can intercommunicate.

Each layer has three interfaces:

- o an interface to the layer above
- o an interface to the layer below
- o the "peer" protocol to another instance of the layer on a different system.

The Application layer has an interface to the applications processors and proccessors above requesting or responding to the OSI services below. The physical layer has an interface to the common communications media below. Application interfaces between systems is accomplished by the sender calling on services of the layer below, and the receiving system sending up data through layers to the application.

Typical protocols apply to the last item only and are the subject of this section. The ISO/OSI model is, indeed, a model and not a specification for an implementation. In implementation, one system might have all seven layers implemented within one program, while another might have seven sets of code (Hollis). The definition of the first two interfaces listed above are thus the responsibility of the Space Station Program. Although they are not discussed in this section, their definition is critical and may be a development driver in the Space Station Program (McKay).

3.1.1.1.2 CCSDS Standards Model Description

The CCSDS standards are oriented to the noisy space-to-space and space-to-ground links. Standards are described for telemetry and for telecommands, as outlined in Table 2.

3 Telemetry Packet	Standard data structure used to transport user data, e.g. between sensors in space and user data processing facilities. Packet is an "observation" to be delivered intact to user. Packets may be broken into segments for transport.
2 Telemetry Transfer Frames	Standard data structure for transmission of packets when they are transmitted from spacecraft-to-data capture element. Frames encapsulate packets.
Telemetry Channel Coding	Mechanisms applied at communications gateways to protect the transfer frames against error.
1 RF & Modulation	RF Standards

CCSDS Layers
Table 2

3 Telecommand Packet	Standard data structure which is used to transport commands between user control capability and payload. Virtually identical to telemetry packet
2 Telecommand Transfer Frame	Standard data structure, which encapsulates telecommand packets, used to transport telecommand packet when they are transmitted from a commanding point (space or ground) to the payload Standard command operating procedures are executed to control acceptance of frames, retransmission Different from telemetry frames
Telecommand Channel Coding	Mechanisms to protect telecommand frames against error as they pass through space data channels
1 RF & Modulation	RF Standards

Table 2 (Continued)

A discussion is provided on how each CCSDS standard might map into the ISO/OSI model in section 3.1.1.2 in response to comments on the draft of this section. There are differing opinions on how the CCSDS standards map to the ISO. These differences of opinion likely result from the fact that the CCSDS standards were developed for a different purpose than standards developed directly from the ISO model. Specifically, an early CCSDS presentation stated that (Hooke):

- for the ground, ISO/OSI compatibility was assumed.

- for space, commercial standards were viewed as applicable. However, due to the cost of space data links, some general purpose ground protocols were viewed as not applicable, on the grounds of transmission efficiency.

- accordingly, the CCSDS standards would "wrap around" the OSI standards.

Specifically, it was expected that standards consistent with the ISO model would be used to augment the CCSDS standards by providing (Review Comments):

o services for on-board local area data distribution

o services for the distribution of data

There are several options for how this "wrap around" might be implemented. These options, which are described in Section 3.1.1.3, describe how one would select and apply the CCSDS and ISO standards to implement an end-to-end standards architecture.

3.1.1.1.3 Commercial Models Option Description

In addition to ISO and CCSDS, commercial models also exist for standards. An example is the IBM Systems Network Architecture. A disadvantage of such frameworks is that, once used, hardware and software from other vendors is difficult to interface. The system is, in this sense, not "open." For this reason, and since the NASA requirements described above state that the SSDS standards will be described following the ISO model, these models will not be discussed further in this section.

3.1.1.2 Standards Options Within The ISO Layers

The sections which follow identify, for each ISO layer, standards options which exist, and possible areas where new standards development may occur.

Only those aspects of standards involving inter-process communication are covered within the OSI environment, and within this section. Thus, standards for application-to-application communication are covered, while the interface between the

application layer and the layers below is not covered.

The trends for within the layer have been:

- o a significant amount of agreement on standards for data communication at the lower levels (one-to-three).
- o more standardization over time as one "moves up" levels 4-7.

Within ISO, a standard has three levels of status:

- o Draft Proposal
- o Draft International Standard
- o International Standard

3.1.1.2.1 Application Layer Standards

3.1.1.2.1.1 Option Description

The application layer provides standards specific to a particular application, allowing for applications on different processors to communicate. The application layer has been divided into two sublayers:

- o specific application service elements
- o common application service elements

Common application layer service elements include (Rauch-Hindin):

- login
- password checks
- set up associations to named peers and agree on the semantics of the information to be exchanged.

Specific application service elements include (Rauch-Hindin):

- file transfer & access
- virtual terminals
- message handling
- document transfer
- job transfer & manipulation

- video text
- graphics
- transaction control, real time control
- commitment, concurrency & recovery
- industry protocols (e.g., purchase orders, credit checking, invoice, inventory)

Specific protocols at the application layer are beginning to be defined. They support communication of the "semantics" or meaning between two applications. Application protocols under definition for OSI are:

- o Virtual Terminal Service (Lowe) -- several protocols for distributed terminal applications (Rauch-Hindin):
  - Basic Class Virtual Terminal (Draft - 2/85)
  - Forms Class Virtual Terminal
- o File Service (Lewan & Long) -- defines a standard for transferring, accessing, and managing information stored in or moved between systems as files. (Draft 11/83, Draft International Standard, 2/85) (Rauch-Hindin)
- o Management & Job Transfer Services (Langsford, Naemura, Speth) -- defines standards for control of distributed processing applications (Draft 7/84 - Rauch-Hindin)
- o Message Handling Protocol -- defines standards to allow interconnection of electronic mail systems (Draft, fall, 1984 -- Rauch-Hindin)
- o Directory (Draft, 2/85)
- o Office Document Interchange Facility
- o Commitment, Concurrency, & Recovery (Draft 7/84, Rauch-Hindin)
- o Formal Description Language & Techniques (Draft 2/85, Rauch-Hindin)
- o Common Application Layer Service Elements (Draft, 2/85, Draft International, 12/85, Rauch-Hindin)
- o Purchase Order Creation & Update (Draft agreed to in ANSI, Rauch-Hindin)

The CCSDS Time Code Standard maps to the Application Layer.

The CCSDS Telemetry Packet Standard provides the following services:

- o formatting of user data
- o support for the services of accounting, correlation, delimitating and interpretation of customer data
- o formatting of ancilliary data

The CCSDS Packet & Telemetry & Telecommand Standards has been proposed as mapping to:

- o the application data, outside of the ISO data system protocols,
- o an application layer protocol, providing an industry specific standard,
- o the presentation layer, providing special purpose formatting, as described in the previous section (McKay),
- o the transport layer, providing a special purpose space-to-ground transport standard (Carper),
- o a special protocol applied for the downlink, outside of the ISO data system protocols,
- o a standard providing services of the network, presentation, and application layer combined (CTA).

The standard is viewed as a data formatting convention that provides the basic, core set of labelling message structures to support space missions within an adaptive distributed system (review comments). In this regard, the SSDS is required to support the need to support multiple payloads, owned by different customers, in a way that minimizes interactions between customers and a minimum of software re-configuration as this mix of customers changes.

Thus, most or all data is required to be multiplexed into a single TDRS downlink/uplink in the form of complete, autonomous, self-identifying data units (telemetry/telecommand packets). A related application requirement relates to the provision of ancilliary data (Task 1, Section 5.3.2.4):

- o the SSDS shall provide ancilliary avionics and housekeeping data (timing, state vector, RF communication, System Status, Acquisition of Signal/Loss of signal, moding, pointing, etc.) to the attached payloads and customers.

3.1.1.2.1.2 Applications Options Characterization

Protocols in the application layer are in a less advanced state than the lower levels. Many de-facto standards exist, generally tied to vendor architectures (e.g., DECnet), specific operating systems (e.g., the Unix interprocessor file transfer services), or specific networks (e.g., the ARANET File Transfer and Remote Login services). An approach is being developed to extend access control requirements of the proposed military standard for Common APSE Interface Set (CAIS) (LeGrand). (The CAIS is a basic software interface lifecycle for DOD mission critical computer systems). In this approach, access control is provided within the ISO framework and building on the ISO Virtual Filestore.

Single vendor protocols (e.g., DECnet or SNA) generally extend through all seven layers of the ISO model and vendors have attempted to map their architectures to ISO. The use of such a protocol is, of course, an option for the entire SSDS or for subnetworks, but as discussed previously, we are emphasizing standards of a more general nature. Of major significance, as the NBS's cooperative work with more than 30 computer and semiconductor manufacturers and developing applications layer standards supporting file transfer protocols for local area networks based on the IEEE 802 physical and data link standards. These efforts are the broadest practical attempts to standardize protocols throughout the ISO hierarchy, with working intervendor network already demonstrated.

Candidate protocols within the application layer for SSDS are:

ISO Common Application Services Elements (CASE's) and Specific Application Service Elements (SASE's); (ISO/TC97/SC16) which include File Transfer Service, Virtual Terminal Service, and the Job Transfer and Manipulation Service.

ISO Message Handling Services (ISO/TC97/SC18)

NBS Message Handling Services (ISO/TC97/SC18)

NAPLPS/ANSI Standard X3.110 Graphics Standards

CCSDS Packet Standard (present or modified)

3.1.1.2.2 Presentation Layer

3.1.1.2.2.1 Presentation Layer Options Description

The presentation layer provides independence to applications from differences in the representation of data -- in its "syntax." The presentation level protocols negotiate the transfer syntax for character sets, text strings, data display

formats, graphics transfer, file organization, data types, and financial information (Rauch-Hindin).

Candidate services for the presentation layer include:

- o Code Conversion (e.g., ASCII-to-EBSDIC)
- o Privacy Services -- encryption
- o Data Compression
- o Protocol for negotiation of syntax

Draft OSI standards for the presentation layer are expected in 1984 (Day & Zimmerman), and a Draft International Standard is expected 2/85 (Rauch-Hindin).

The SSDS requirements on the Presentation Layer are less than what may be needed for a public network. For example, while the SSDS provides privacy, GSFC customer requirements state that encyrption is a customer responsibility. Similarly, it seems unlikely that code conversion will be required since customers are expected to use standard terminal equipment. The CCSDS Packet Standard might be viewed as within the Presentation Layer. The information stored or available from the applications processors above the application layer could be framed in the CCSDS format by the Presentation Layer - for example, experiment data might be formatted using CCSDS while queries and virtual terminal interactions might not be so framed (McKay).

3.1.1.2.2.2 Presentation Layer Options Characterization

Candidate protocols include:

ISO Connection, Cointext, Information Transfer, Dialogue Management, Synchronize, Interrupt, and Terminate services described in the draft ISO presentation layer standards.

FIPS 46, the NBS Data Encryption Standard

ANSI Standards X.3.4 (Standard ASCII characters), X3.15 (Bit Sequencing) and X.3.16 (Character Structure and Parity).

AT&T BX.25, which extends into the Presentation and Session layers.

CCSDS Packet Standard (present or modified).

3.1.1.2.3 Session Layer

3.1.1.2.3.1 Session Layer Options Description

The session layer is responsible for establishing and managing connections between two processes, and provides for recovery from failure at the transport layer. The session layer

is responsible for the mechanisms for organizing and structuring the interactions between application processes. Specific functions include (Rauch-Hindin):

- o connection establishment and termination
- o data transfer
- o synchronization between end-user tasks
- o graceful and abrupt closure of a session
- o map user address to names
- o dialog control (who, when, how long, half or full duplex)
- o quarantining of data (buffering of data until instructed to deliver it(.

Draft OSI standards are under development (Day & Zimmerman).

3.1.1.2.3.2 Session Layer Options Characterization

Session Layer Protocol candidates include portions of the previously mentioned protocols (NBS, ISO, AT&T) which map into the ISO model. The main candidates are the ISO recommendations for Session Service Definition (X.215) and Session Protocol Specification (X.225) including services for Normal Data Exchange, Expedited Data Exchange, Token Management, Dialogue Control, Synchronization, Resynchronization, Activity Management, Exception Reporting, Typed Data, and Capability Data. The Session Layer has not received the attention given to other layers and standards are only beginning to emerge.

3.1.1.2.4 Transport Layer

3.1.1.2.4.1 Transport Layer Options Description

The transport layer provides the "users gateway" into the data system and the transport level protocol is to provide a reliable, end-to-end communication path. The levels below the transport layer only deal with protocols between "nearest neighbors" on the communications network, whereas the transport layer is at a "user machine-to-user machine" level. For the Space Station, this would mean "payload-to-payload control center processor", or "core-system-to-operations center processor" communication. The Transport Layer accepts data in message form from the session layer, breaks it down into smaller pieces if required by the network layer, and vice-versa. Transport layer standards provide for end-to-end (host-to-host) connection establishment, ordered delivery of data, error control, and flow control on an end-to-end basis. The transport layer thus supports (Stallings, Rauch-Hindin):

- transfer of data between two transport users, addressing end-user machines without concern for the route of the messages or the addresses of machines enroute between end-user machines.

- grade of service (e.g., acceptable error levels, delays, priority) as requested by users

- connection management (establishing, maintaining, and terminating logical connections between end points)

- handling requests for expedited delivery or security services

- monitoring of quality of service, status reporting, end-to-end error detection and recovery

- multiplexing end-user address onto network.

The ISO Transport protocols are linked to three network types defined by ISO (Stallings):

Type A Network - Network connection with acceptable residual error rate and rate of signaled failures

Type B Network - Network connection with acceptable residual error rate but unacceptable rate of signaled failures

Type C Network - Network connection with residual error rate not acceptable to the transport user.

The ISO has defined five classes of transport protocol, which depend on the user service requirements and the available network services (Stallings):

- Class 0 - Simple - no explicit ordering or error control (used with Type A networks)

- Class 1 - Basic error recovery - provides minimal error recovery and expedited data transfer (used with Type B networks)

- Class 2 - Multiplexing - adds the ability to multiplex multiple transport connections into a single network connection, plus explicit flow control since a single network connection does not control flow for all transport connections (used with Type A networks)

- Class 3 - Error recovery and multiplexing - union of Class 1 and Class 2 capabilities and also contains the resynchronization and reassignment capabilities needed to cope with failure prone networks (used with Type B networks)

- Class 4 - Error detection and recovery - uses full range of transport level capabilities to handle an unreliable, error prone network (used with Type C networks).

It has been proposed (Carper) that SSDS transport requirements include provision of the following services:

- quality of transport service: computer quality or normal quality
- delivery services: immediate delivery or store and forward services
- reliability services: verified delivery (with re-transmission) or unverified delivery (datagram).

Meeting the requirements for these services might best be associated with a Transport Layer standard for the SSDS.

Implementation of some of these services would interact with other layers (e.g., with a network layer having both connection and connectionless/datagram modes).

A related proposed requirement (Carper) is that these services should be symmetric for Space Station, that is, they would be available in either direction. Since many future payloads are expected to utilize considerable processing, one needs to view payload-to-ground communication as being computer-to-computer rather than computer-to-human. Thus, one could have "downlink commands" -- such as data base requests issued by a payload processor, and "uplink telemetry" -- such as data from a computer data base. This flow will be increasingly true as functions migrate from ground to space.

One could view the CCSDS Packet Standard as mapping to the Transport Layer. For example, one could think of the current CCSDS Packet Telemetry Standard as providing computer quality, store and forward, unverified delivery, while the CCSDS Telecommand Standard as providing normal quality, immediate, verified delivery. Provision of all of the above required services would appear to require modifications of existing standards.

3.1.1.2.4.2 Transport Layer Options Characterization

The candidate protocols are:

ISO recommendations for Transport Service Definition (X.214) and Transport Protocol Specification (X.224) which defines the five classes of service described above.

The ARPANET Network and Transmission Control Protocols (NCP and TCP).

CCSDS Telemetry Packet Standards (present or modified version)

3.1.1.2.5 Network Layer

3.1.1.2.5.1 Network Layer Options Description

The Network Layer provides the layers above with independence from the actual data transfer technology used -- e.g., whether the network uses optical fiber, local area networks, packet switching, etc., should all be hidden (Day & Zimmerman). The network layer protocols go from "network port-to-network port." For the Space Station, for example, this would go from the on-board NIU interface, to another such interface, or to the interface to the ground based remote area network. The network layer provides routing and congestion control to higher layers. Standards at this level allow routing, sequenced delivery, and accounting functions for data transfers between a computer and the network, and duplicate functions performed at the data link level, at the network level. Specifically, the functions include:

- o establishing a logical connection between endpoints on the network, setting up routes for packets to travel, and addressing network machines on the route through which the packets travel
- o managing the connection and disconnecting the connection after use
- o delivery of messages over the logical link (control of the logical channel), in the order in which they were sent (sequencing), with error control
- o transport flow control, so that the receiving point on the network is not overloaded with messages
- o manage the use of multiple (parallel) links to increase throughput
- o prevent any one user from overload the transmission resources to that other users are blocked
- o alternate routing, to avoid failed or congested links
- o network traffic monitoring, billing
- o internetworking
- o disassembly of transport messages into packets and re-assembly at destination.

A recent development at the network layer has been the development of "connectionless" protocols for ISO (Rauch-Hindin).

Previous network standards were connection oriented," meaning that calls or connections between the two endpoints were made prior to data transfer. Connectionless mode does not make this assumption and is used to send single data packets, each packet having no relationship to other packets. This mode is preferred by the Department of Defense since they feel it allows (Rauch-Hindin):

- greater survivability, with the network less vulnerable in case of attach and more able to recover

- it allows adaptive routing and simpler network operation because most bookkeeping and message integrity controls are performed at the terminal ends rather than in the network.

Connectionless or datagram mode is particularly appropriate for local area networks due to their high reliability. However, it is noted that ARPANET functions in datagram mode.

Data base research involving geographically separated nodes have shown a need for both connection and connectionless protocols. Many types of real time controls will require acknowledgements and rapid interchange of signals and data not supported by a connectionless mode (McKay). Other types of data transport will not require these services, such as transport of non-interactive experiment data.

The CCSDS Telemetry & Telecommand Segmentation provides:

- o control of channel resources, so that no one source exclusively captures use of the channel

- o segmentation may be performed by the application process, or by the spacecraft data system using one of two formats.

One view is that CCSDS segmentation should be mapped to Layer Three (Network) which were suggested in comments on a draft of this paper. Other inputs have suggested that segmentation was added to meet the needs of specific space agencies and might not be used for the Space Station Program.

The requirements for SSDS end-to-end networking are significantly different from the past. Requirements from the Task 1 report that bear on the data distribution network and indirectly on the selection of protocols are:

- o provide real time distribution of real time and near real time data, including level 0 processing, demultiplexing, buffering, routing, and retransmission (Section 5.3.1.3)

- provide real time, raw payload data to the customer (Section 5.3.1.1)

- support real time re-allocation of data distribution resources to help meet customer priorities (Section 5.3.3.3).

Additional NASA requirements affecting the choice of communications protocols for the network include (CRSS):

- The SSIS/SCS network data handling shall be independent of the format or content of the customer data (CRSS, p3-1)

- The data network shall be able to transport and delivery customer data sets intact, without having any knowledge of their internal format or content (CRSS, p 2.2.3.4)

- customer data shall be delivered without alteration of its contents. Any artifacts imposed by the data transport service, e.g., data reversal due to communications buffering, shall be removed before data delivery to the customer.

Finally, the mission data base poses stringent delivery delay requirements -- data must be delivered to customers within hours rather than months.

The above requirements imply:

- for on-board networks, the need to support an evolutionary expansion of the Space Station with a minimum of effort,

- a need to rapidly separate the downlink data by customer ID and distributed it electronically to the customer, and similarly for the uplink,

- the desire to transport the customer data (the telemetry/telecommand packets) by electronic means from the payload to the customer's premises, perhaps in near real time (as well as non-electronic delivery) with a minimum of network re-configuration,

- that the selection of which Wide Area Standards are appropriate is dependent on the data characteristics and several standards might be used (circuit, packet),

- compatibility with existing standards and the use of interfaces compatible with existing customer equipment.

Layer three of ISO is responsible for network services as described above. The protocol must, thus, have a means of specifying the network address point of the source and destination of the data to be delivered - the payload address (in space) and the customer address (on the ground). CCSDS Segmentation does not provide this information, nor do other layers. The CCSDS Packet and Frame Headers identify the on-board application process, and the downlink channel, but not the customer destination address. This assumes that an end-to-end route is configured for the period during which the payload is operating, as opposed to having this information in the packet header itself.

The emerging design for the SSDS is a distributed processing network, with many endpoints in space, and many endpoints on the ground where processing is performed. The ground endpoints include Data Handling Centers (to perform standard Level 0 processing), multiple Regional Data Centers, and many customers who receive data electronically. In addition, there will be control centers for the Space Station, for the COP and POP, POCC's, and customer control centers.

Furthermore, one customer may have many payloads on the Space Station. Some customers may receive data from one or many payloads and some payload data will be routed to different customers and destinations. Some data will be routed directly to a single customer's facility, while other data will be held at Regional Data Centers and distributed from that point.

Network services must thus be provided, in a way that allows data to be handled in as automatic a fashion as possible and a minimum of network re-configuration. For example, in the SSDS operations concept, customers may logon, be connected to a control center, and send commands to their payload. Resulting data may be returned to their location at a later time. Ideally, this process would occur transparently as long as the customer is operating within the resources (such as TDRS bandwidth) allocated to that customer.

For example, it is possible to infer the destination address based on a combination of the source, and a schedule. That is, one could read the source and, knowing the schedule of who was to receive the data from that source at the time, route the data. If the re-scheduling was infrequent - for example, when customers are added to the network - this appears feasible.

3.1.1.2.5.2 Network Layer Options Characterization

Candidate Network Layer protocols include:

X.25 which also effectively encompasses the Data and Physical Layers,

X.213/DP 8473, the connectionless standard,

X.75 Internetworking Protocol

X.21, Circuit Switching

NASA Specific protocols, such as NASCOM blocks.

We note that some of these protocols which use re-transmission strategies are likely to be inappropriate for bulk traffic on some subnetworks or links (e.g., space-to-ground) but are potentially applicable to others (e.g., the Wide Area Network).

3.1.1.2.6 Data Link Layer

3.1.1.2.6.1 Data Link Options Description

The data link layer is responsible for providing an error free line over any link on the network, e.g., on the link between the host and the network, or links between network nodes. It provides the functional and procedural means to transfer data, and it performs error detection, sequencing, time out and acknowledgement, and flow control. The purpose is thus to take the basic transmission line at the physical layer, and make it appear to the network layer that it has an error free line. The data link layer performs functions similar to those of the network layer, but on a link-by-link basis. Specifically, the data link functions include (Martin):

- o establishing, initializing, and disconnecting a logical link between two points connected by a physical link,
- o transmission of frames over a physical path,
- o control of the link during data interchange,
- o detection of the beginning and ending of a frame,
- o detection of transmission errors, error handling (e.g., re-transmission),
- o provision of data transparency, so that any pattern of data may be sent,
- o link flow control, so that the transmitter does not overload the receiver with data frames.

It has been suggested that the CCSDS Frame Standard provides services that could be used for both the uplink and the downlink (review comments):

- o transport of telemetry packets over the space-to-ground link,

- o time sharing of the links between different data types,
- o a virtual channel which allows segregation of different data types,
- o a status insert so that status and audio data may be inserted synchronously, and
- o a bi-directional transfer mechanism.

The CCSDS Telecommand Transfer Frame Standard provides:

- o transport of telecommand packets from ground-to-space,
- o Command Operation procedures provide for re-transmission of missing frames.

The CCSDS Trasnfer Frame Standard could be thought of as mapping to:

- o the data link layer providing transport over the space-ground links,
- o both the data link and network layers (CTA).

The CCSDS Telemetry & Telecommand Channel Coding Standard provides:

- o error protection and correction over the space-to-ground link.

The CCSCS Channel Coding could be thought of as mapping to:

- o the data link layer, providing special error handling for the space-ground links.

Bit Error Rate requirements range from 10**-7 (CRSS) to 10**-9 for computer data (Phase B RFP).

The SSDS requirements for the data link layer implied are:

- o for the space-to-ground link, standard services to limit the error rates,
- o for ground links, the need to move a large volume of data over a variety of links, perhaps at very high rates, to the NASA centers and the customers premises,
- o the use of protocols which can perform error handling at high rates (e.g., re-transmission protocols may be difficult for some data types), and

- o the use of interfaces compatible with existing customer equipment.

3.1.1.2.6.2 Data Link Options Characterization

The main distinction in Data Link Layer protocols is that of byte and bit oriented protocols. The following are candidate protocols are:

ADCCP (ANSI X3.66 and FED-STD 1003)

LAP and LAPB, the X.25 data link protocols

HDLC (ISO 3309 and 4335), basis for X.25 data link protocol

BISYNC, widely used IBM character oriented standard

SDLC, IBM bit oriented standard

ANSI X3.28, control character protocol

IEEE 802.2, logical link protocol for the other 802 standards

ISDN D-Channel Protocol (LAP D)

ARPANET IMP-IMP protocols

ISO protocols being developed for fiber optic LAN's, which includes work by NASA for the NIU (Network Interface Unit) at JSC and GSFC.

3.1.1.2.7 Physical Layer

3.1.1.2.7.1 Physical Layer Options Description

The physical layer provides for a physical interface between the terminal equipment and the data network -- the mechanical, electrical, functional, and procedural standards to access the physical medium. The concern is to send bits over a physical transmission facility. Specifically, the physical layer:

- o provides an electrical and physical interface between the data source and the data communications equipment,
- o establishing and disconnecting a physical transmission path,
- o transmitting bits over the physical path, and
- o alert the link layer to physical path failures.

The CCSDS RF Standard maps to the physical layer.

3.1.1.2.7.2 Physical Layer Options Characterization

The protocols in this layer are being affected by rapidly evolving hardware technology, particularly in the area of fiber optics, and it is likely that a number of existing standards for Physical and Data Link interfaces are likely to be translated into higher bandwidth fiber optic networks in the next few years. (As noted in the previous section, work is underway at JSC and GSFC in these areas.) Physical layer standards based on star topologies are also likely to arise since this structure seems appropriate to fiber optic LAN's.

Candidate protocols include:

IEEE 802 which includes CSMA/CD, broadband CSMA/CD, token bus, and token ring standards under a common Data Link layer.

RS-232-C

RS-449 (including RS-422, RS-423;) and FED STD's 1030, 1020, and 1031)

X.20 and X.21 (physical), synchronous and asynchronous protocols for X.25

X.21 bis, interim X.25 physical layer standard similar to RS-232-C

X.24 DTE/DCE interface

X.26, X.27, and X.29 modem protocols

ISDN physical layer standards

NASA specific flight standards for LAN's such as the MMS bus.

Fiber Optic LAN Interfaces, Under Development

3.1.1.3 Standards Architecture Options Description

There are several options for how one can select the various ISO and CCSDS standards to implement an end-to-end standards architecture. These options differ in their impacts on the various elements of the SSDS. This section takes the view that it is critical to select the standards for each of these elements using the same model, and with an understanding and balancing of the impacts on all of the elements of the SSDS. Thus, one should consider the needs of the ground segment when selecting the space segment standards, and vice versa, so that no one element is unduly impacted.

Customer interfaces and impacts are particularly important - one should not impose new unique standards data formats - or

worse, multiple and conflicting -- standards on Space Station customers unless there is no other option.

The question is thus what the end-to-end standards architecture should be for the SSDS. How do the CCSDS standards and the standards derived from the ISO/OSI model wrap around each other. Four logical options are presented:

- o ISO Compatible Standards for Local & Wide Area Networks (space & ground) combined with:
  - a) CCSDS Packets as an ISO Upper Layer Standard,
  - b) CCSDS Packets & Frames "Below" On-board ISO, or
  - c) CCSDS as Alternate Downlink Standards for ISO Levels 1-3.
- o ISO Standards Only

Before reviewing these options, a final word on terminology. The word "packet" has been used in this section for two slightly different concepts:

- o a "telemetry packet" means the package of data from the payload, and the relevant ancilliary data, which is to be delivered to the payload customer. The "telemetry packet" may be of variable length but may be very large - for example, a scan line from the instrument. (CCSDS packets have a maximum length of 2**16 - 1.) This might better be called a "message", since the packets may or may not be packet switched (see below).
- o a "packet" in the communications sense is defined by CCITT as "a group of binary digits including data and call control signals which is switched as a composite whole." The data, called control signals, and possible error control information, are arranged in a specified format. "Packet switching" is defined as "the transmission of data by means of addressed packets whereby a transmission channel is occupied for the duration of transmission of the packet only. The channel is then available for use by packets being transferred between different data terminal equipment. Note: The data may be formatted into a packet or divided and then formatted into a number of packets for transmission and multiplexing purposes (Martin). The packets are of variable length up to some maximum (for example, 128 bytes for X.25 packets).

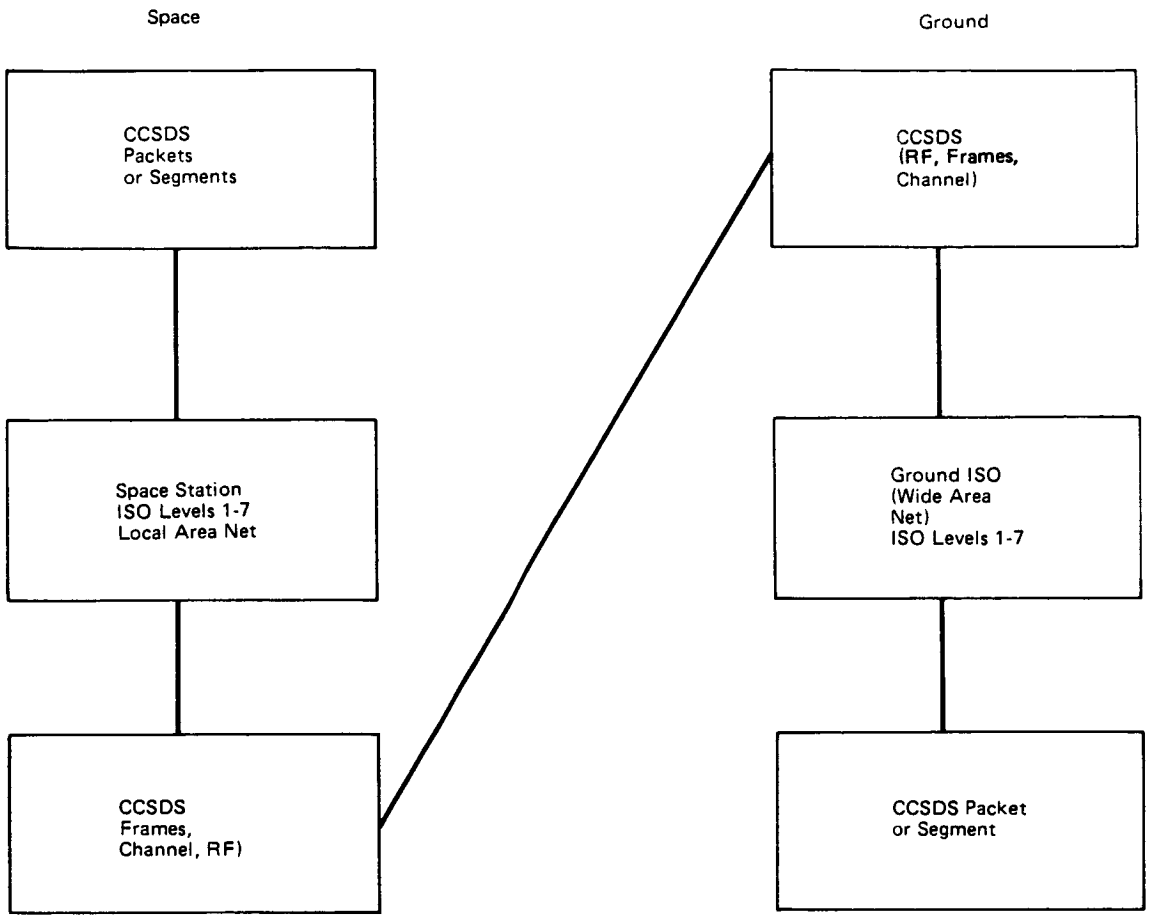
Thus, "telemetry packets" could be either sent on a dedicated physical path, they could be circuit or message

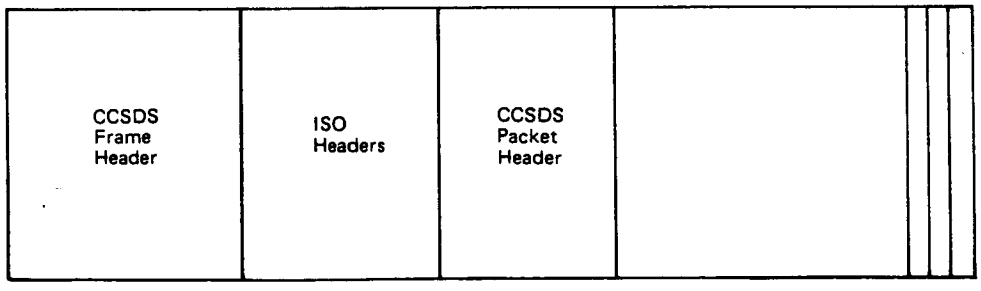
switched, or the telemetry packets could be treated as data, further broken into packets, and sent over a packet switching network.

3.1.1.3.1 CCSDS Packets Implemented as ISO Upper Layer Standard

3.1.1.3.1.1 Option Description

The first option is illustrated on the next page. It shows a "logical view" of how the end-to-end standards architecture might look -- that is, it does show the hardware and software elements.





First Option

In this first option, the CCSDS packet standard (present or modified) is considered to be an upper layer of the ISO model. Possibilities that have been proposed are:

- the CCSDS packet standard is implemented as application data or an application layer standard,
- the CCSDS packet standard is implemented a presentation layer standard, or
- the CCSDS packet standard is implemented as a transport layer standard.

(These possible mappings of CCSDS Standards to the ISO model are further discussed in section 3.1.1.2).

Each "packet" -- the message containing an observation from the payload -- is delivered to the Space Station local area network, which implements some portion protocols in ISO layers 1-7.

If the CCSDS packet is destined to go off, the Space Station (e.g., to the ground) then CCSDS segments and frames are formed, perhaps in a special purpose gateway. A parallel process occurs on the ground, and the protocols used within the ISO layers will likely be different than used on-board (e.g., local area network protocols vs. long-haul communications protocols).

3.1.1.3.1.2 Option Characterization

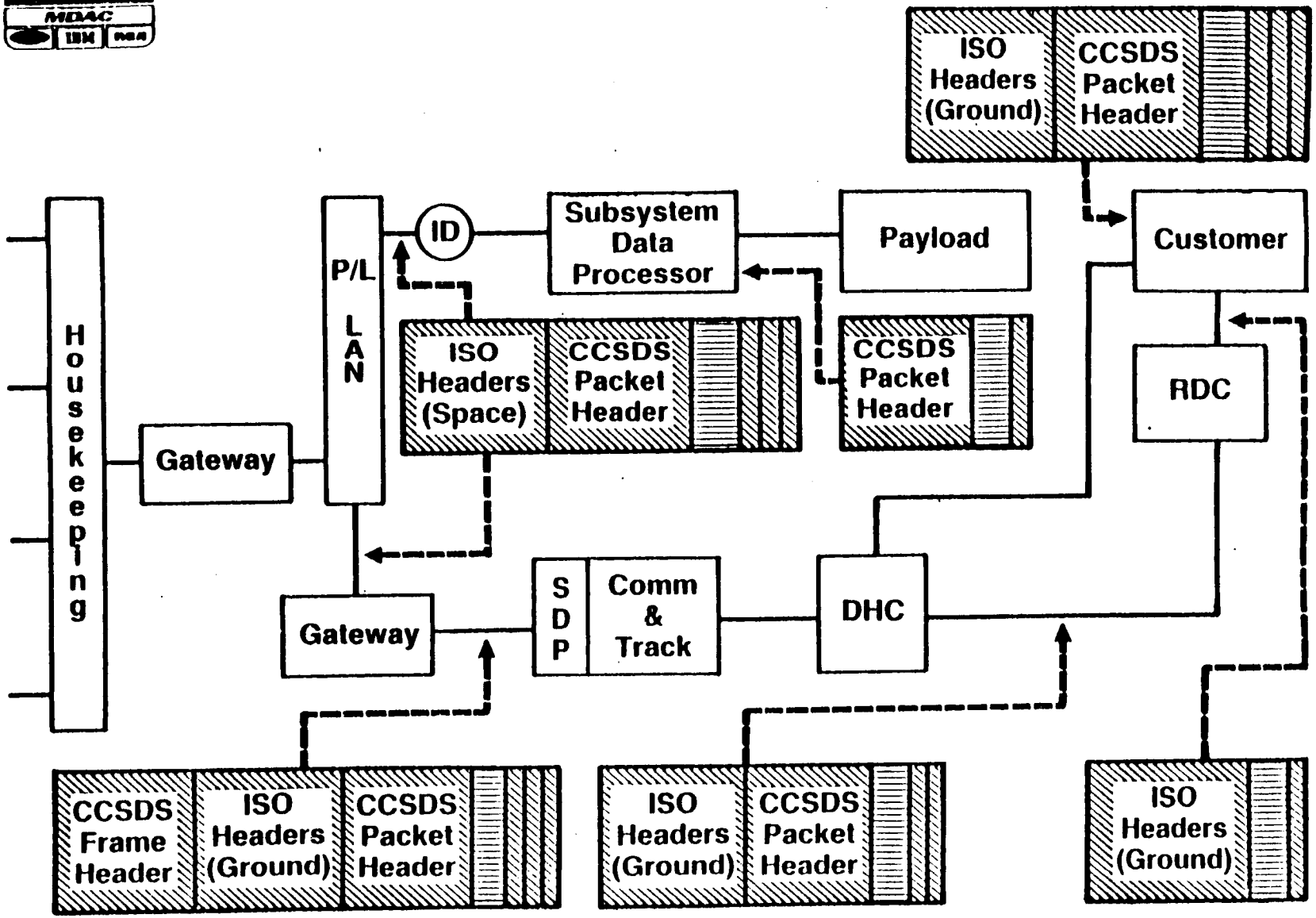
This option becomes more clear if one examines a possible physical architecture. There are several possible ways to implement this option; one is shown on the next page. The way to read the diagram is as follows:

- the reference configuration for the data system is shown, as supplemented by ground elements. Thus, data starts with the payload, connects via interfaces to the on-board payload LAN, thru C&T, to a Data Handling Center, and from there to Regional Data Centers and Customers,
- the formats shown below the figure illustrate how the data might be formatted at that point in the data flow.

The diagram is only intended to apply to the "traditional telemetry" downlink.

CCSDS PACKETS IN ISO UPPER LAYER SAMPLE ARCHITECTURE VIEW

VFZ247



The impacts on the major elements are described below.

The payload always has a packet format, whether it is in the laboratory or in the Space Station or platform. This simplifies testing for the customers.

The on-board LAN must carry the CCSDS information, e.g., headers, trailers, ancillary data. Whether this data is "added overhead" which of the "suboptions" are used for the CCSDS Packet formats and may or may not be significant. If the packet format is considered as part of the application data, then it may duplicate header information used by other layers (such as transport). This may not be the case if the Telemetry/Telecommand standards are implemented, in present or modified form, as an upper ISO layer standard.

Ancillary data is provided over the LAN to the payloads. This is consistent with customer requirements.

The impacts are the TDRSS downlink/uplink, and the Wide Area Network, are similar to those stated above for the on-board LAN.

The Wide Area Network must route data in the downlink to some location, e.g., an RDC, (and vice versa for the uplink). (Similarly, "commands" or data must be routed from a ground point, the NGT to the Space Station element, and finally to the payload.) One approach to performing this routing is to read segment headers and route messages or packets, as illustrated. As an alternative, it may be possible to route a class of data (low to medium rate) more or less automatically. It may be possible to do this by reading the telemetry packet source ID, in combination with a schedule, as discussed in Section 3.1.1.2.

Alternatives are (a) send each customers data on a scheduled channel basis, (b) employ application specific software to process the data, sorting it, and sending it to the destination. Scheduling could be very complex.

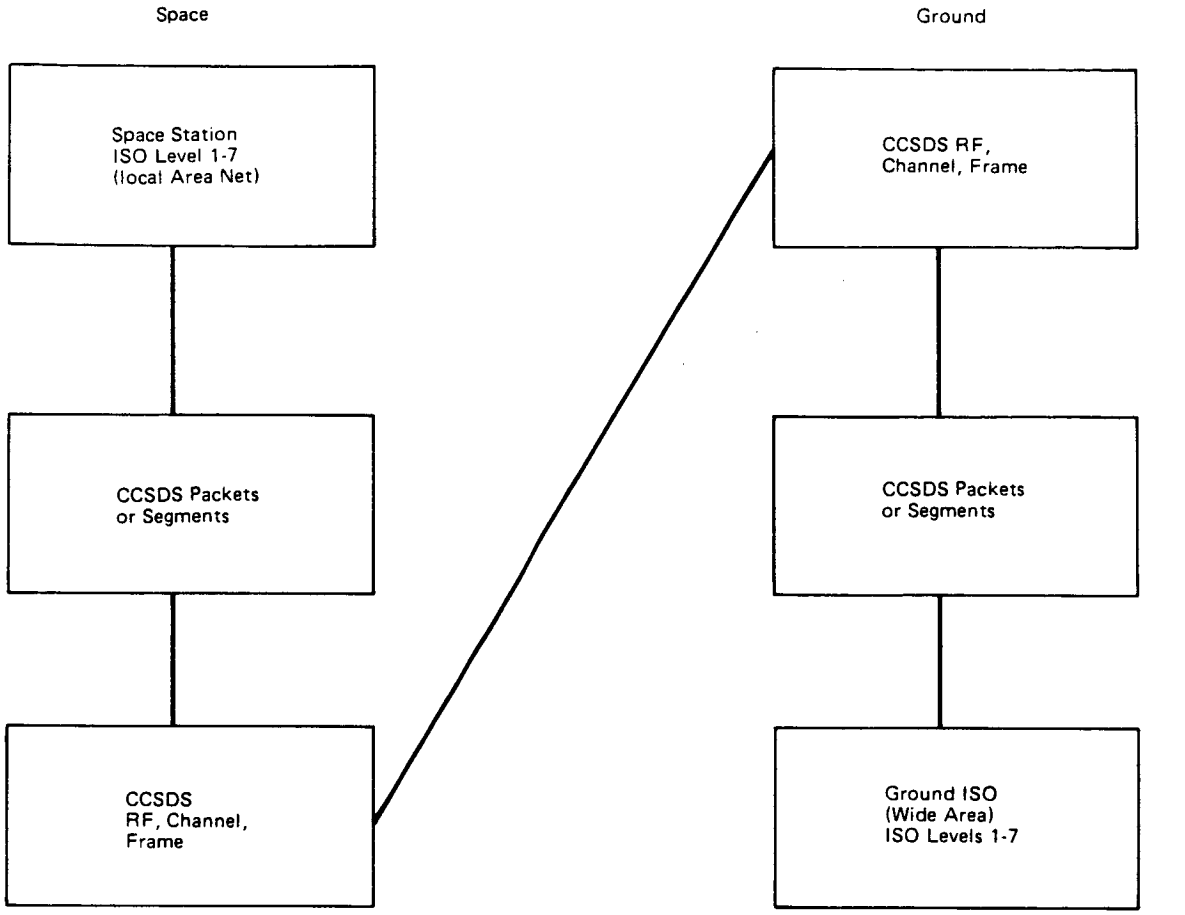
The impacts on the customer of this approach are:

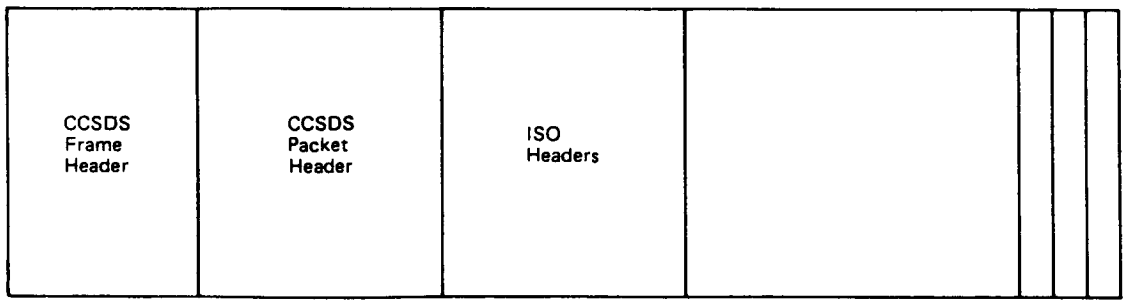
- o the payload interface is constant,
- o timeliness of delivery will depend on whether the data can be routed automatically.

3.1.1.3.2 CCSDS Standards Implemented Below On Board LAN/ISO

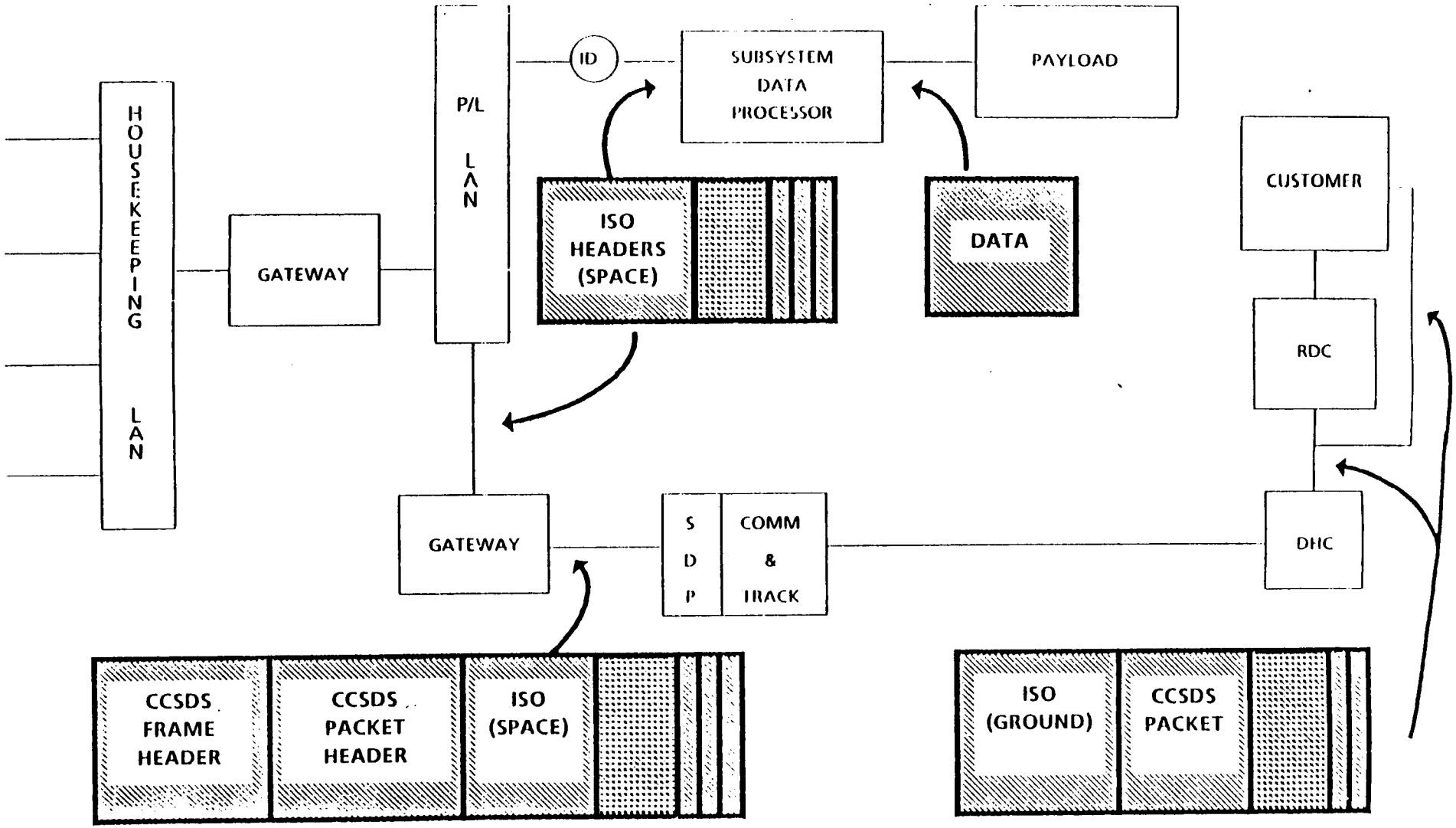
3.1.1.3.2.1 Option Description

In the second option, the entire set of CCSDS standards are "below" the on-board ISO local area network. CCSDS Packets, Segments, and Frames are formed for the downlink. Different





Second Option


HOUSEKEEPING LAN
GATEWAY
P/L LAN
ID
SUBSYSTEM DATA PROCESSOR
PAYLOAD
ISO HEADERS (SPACE)
DATA
CUSTOMER
RDC
DHC
GATEWAY
S D P
COMM & TRACK
CCSDS FRAME HEADER
CCSDS PACKET HEADER
ISO (SPACE)
ISO (GROUND)
CCSDS PACKET

telemetry packets might be created for each payload, since each is intended to be a payload observation. If the same length packets were formed for each payload, some of the value of the packet telemetry approach would be reduced.

3.1.1.3.2.2 Option Characterization

The impacts on the major elements are described below.

The payload format is not constant, since the telemetry packets are formed by the SSDS in the Space Station or Platform. This complicates testing.

The on-board LAN does not carry the CCSDS information, e.g., headers, trailers, ancilliary data.

Ancilliary data is provided over the LAN to the gateway instead of the payloads. This is not consistent with customer requirements. It is likely that the same ancilliary data would be provided to each payload in this implementation. The gateway is now a more complex device over the previous approach.

The impacts are the TDRSS downlink/uplink, and the Wide Area Network could be high. Duplicate sets of header information might be necessary.

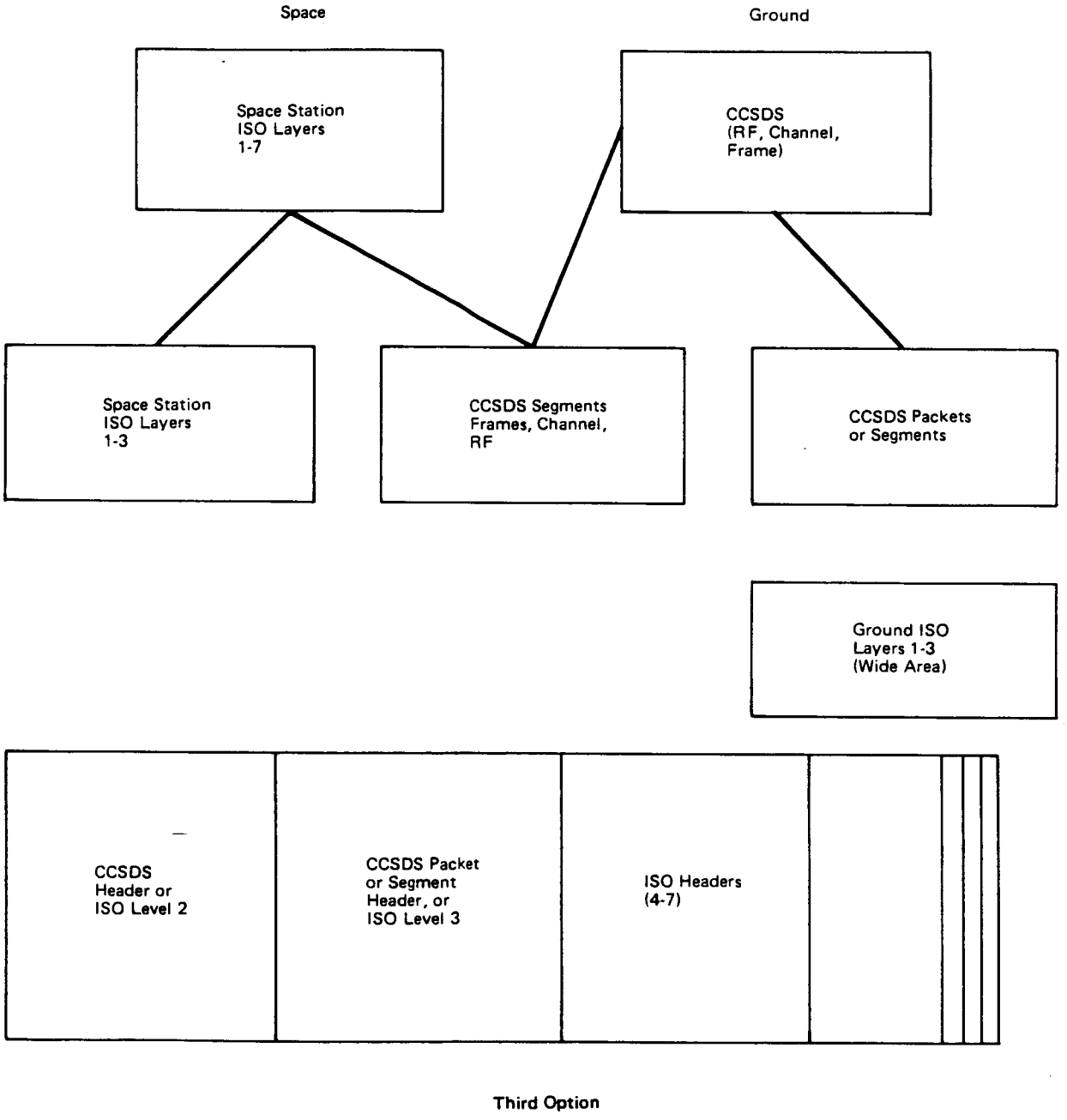
The impacts on the customer of this approach are:

- o implementation and handling of the telemetry packets is an SSDS standard service, but
- o payload interfaces will not be constant, as noted above.

3.1.1.3.3 CCSDS Standards Implemented as Lower Layer ISO

3.1.1.3.3.1 Option Description

The third option is illustrated as follows:

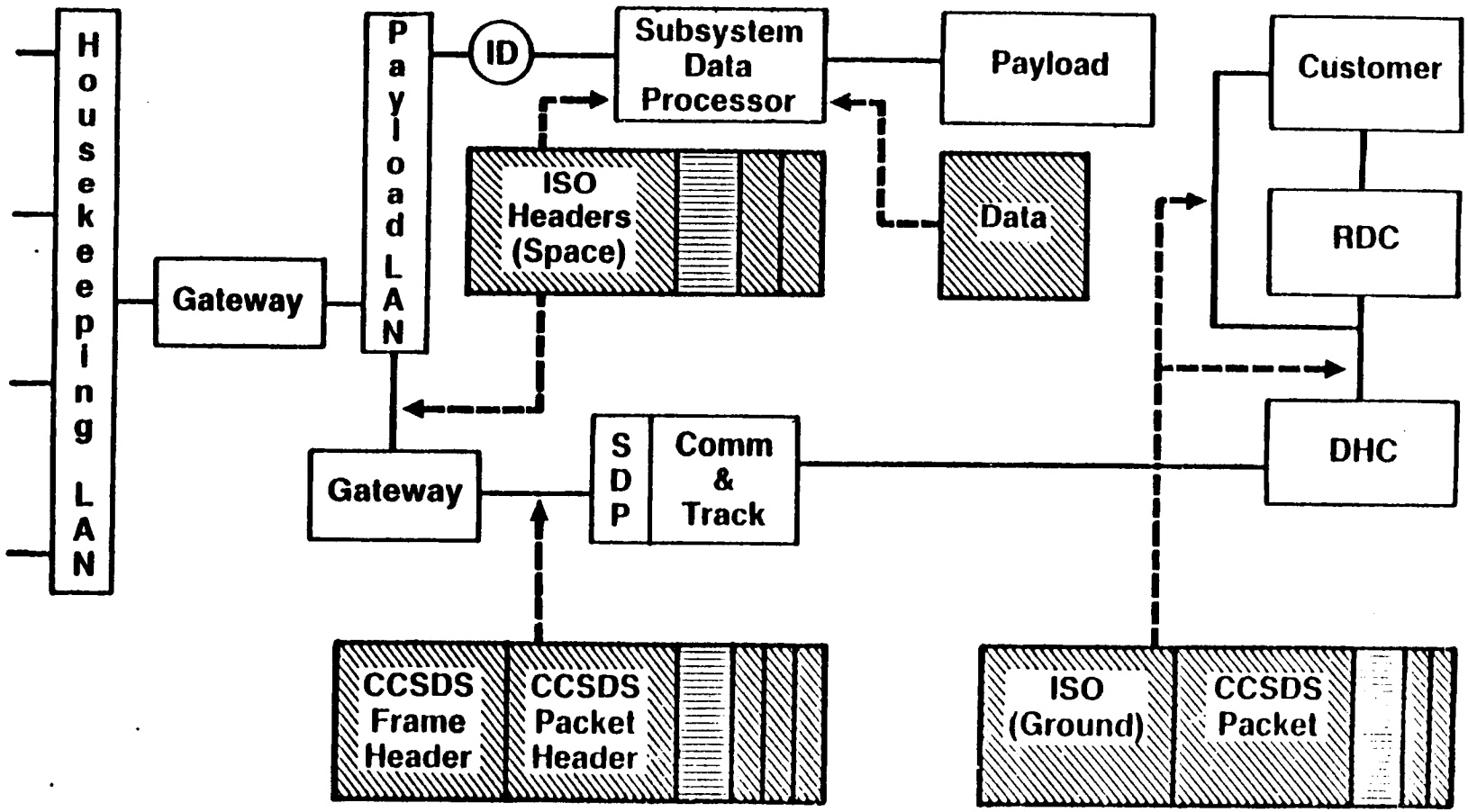


Third Option

CCSDS STANDARDS IN LOWER LAYER ISO/OSI

SAMPLE ARCHITECTURAL VIEW

VGZ249



In this third option (analogous to Swingle & McKay), the on-board ISO layers four-to-seven can either interface to the on-board local area network (using ISO layers one-to-three) for on-board communications, or to the CCSDS Packets, Segments, or a combination of these and relevent ISO information for downlink. It would require that the ISO level three (packets) and CCSDS Packets or Segments be compatible. Information to/from a layer may be different and conversion can be feasible - the formats need not be identical, but would have to be possible to do a transformation.

3.1.1.3.3.2 Option Characterization

The impacts on the major elements are described below.

The payload interface is not constant in a packet format.

The on-board LAN may carry the CCSDS information, e.g., headers, trailers, ancilliary data. Whether this data is "added overhead" which if the "suboptions" are used for the CCSDS Packet formats.

Ancilliary data is not provided over the LAN to the payloads. This is not consistent with customer requirements.

The impacts are the TDRSS downlink/uplink, and the Wide Area Network, are minimized. The impacts on the customer of this approach are:

- o implementation and handling of the packets is split between the SSDS and the customer.

The last "logical" option would be to only adopt standards which are consistent with the ISO model. If one only examines the "official" standards developed under ISO, this does not appear practical. Current network standards developed under the official ISO-umbrella include X.21 (circuit switching) and X.25 (packet switching) which have too much overhead, and which offer insufficient error protection, for the TDRS downlink. For example, X.25 performs error correction by re-transmission. While this may be quite suitable for certain traffic on the Wide Are Network, it is not applicable for much of the downlink traffic. However, using a combination of ISO standards and CCSDS standards appears feasible as illustrated in the other options.

Overhead for applications processors is expected to be offloaded within the next five to ten years to special purpose "black box" processors. Although these will not be flight qualified, some of the software may be transportable to such hardware (McKay).

3.1.1.4 Communications Standards References

Martin, James, Computer Networks And Distributed Processing, Prentice-Hall, Englewood Cliffs, NJ, 1981

Goddard Space Flight Center, NASCOM Development Plan - FY 85-1, September, 1984

Sperry, "FODS Industry Briefing"

Day, J. D., Zimmerman, H. "The OSI Reference Model," Proceeding of the IEEE, Vol. 71, No. 12, December 1983, pp 1334-1340.

Hollis, L. L. "OSI Presentation Layer", Proceeding of the IEEE, Vol. 71, No. 12, December 1983, pp 1401-1403.

Bartoli, P. D., "The Application Layer of the Reference Model of Open Systems Interconnection", Proceeding of the IEEE, Vol. 71, No. 12, December 1983, pp 1404-1407.

Lowe, H., "OSI Virtual Terminal Service", Proceeding of the IEEE, Vol. 71, No. 12, December 1983, pp 1408-1413.

Lewan D., Long, H. Garrett, "The OSI File Service",, Proceeding of the IEEE, Vol. 71, No. 12, December 1983, pp 1414-1419.

Cunningham, I., "Message-Handling Systems and Protocols", Proceeding of the IEEE, Vol. 71, No. 12, December 1983, pp 1425-1430.

desJardins, R., "Afterword: Evolving Towards OSI", Proceeding of the IEEE, Vol. 71, No. 12, December 1983, pp 1446-1448.

Swingle, W. L., McKay, C. W., "Space Station Information".

CCSDS, "Packet Telemetry", Blue Book, May 1984.

CCSDS, "Telecommand Channel Coding", Blue Book, May 1984.

CCSDS, "Telecommand", Green Book, August 1984.

CCSDS, "Telecommand Channel Coding and Procedures", Red Book, May 1984.

CCSDS, "Telecommand Transfer Frame Formats and Procedures", White Book, July 1984.

Tannenbaum, A. S., Computer Networks, Prentice-Hall, Englewood Cliffs, NJ 1981.

Hooke, A. J., Space Station Data Network Concept, Presentation, July 1984.

Conrad, J. W., Standards and Protocols for Communications Networks, Carnegie Press, Madison, NJ.

Stallings, W., "A Primer: Understanding Transport Protocols", Data Communications, November 1984, pp 201-215.

Rauch-Hindin, W., "Communications Standards: ISO Poinsed to Make Its Mark", Systems & Software, March 1984.

Carper, R., Goddard Space Flight Center, Personnel Communications and Comments on Draft Section 3.1.1.

Computer Technology Associates, Computer Networking Study, Final Briefing, January 10, 1985.

Holland, G. L., "NAPLPS Standard Defines Graphics and Text Communications", EDN, January 10, 1985, pp 179-192.

LeGrand, S., "Approach to Extending Access Control Requirements of Proposed Military Standard Common APSE Set (CAIS), Version 1.4, Ford Aerospace, Internal Presentation.

McKay, C. W., Comments on Draft of Section 3.1.1.

Review Committee, Comments on Draft of Section 3.1.1.

3.1.2 Data Base Management Systems Standards

A data base is a collection of data which may be stored in a variety of physical and logical formats. A Data Base Management System (DBMS) is a set of programs which defines and manipulates data in a data base as well as providing query, retrieval and reporting capabilities.

This paper considers a DBMS as a set of special-purpose programming languages. The same language design principles such as orthogonality, simplicity, security, efficiency and formal definitions, apply to DBMS, as to programming languages. DBMS should also include powerful data operators, integrated data definition and data manipulation, integrated catalog and compilation and optimization.

Following is a technical discussion of three DBMS categories: relational, network and hierarchical.

In general, there are few adequate standards existing that should be imposed for Data Base Management Systems.

3.1.2.1 Relational DBMS Standards

Standards can be divided into three subsystems: internal (physical storage standards), conceptual (logical storage standards) and interface standards. Query and host language interfaces represent other DBMS areas that are subject to standards.

Internal or storage level standards do not exist for Relational DBMS. However, NASA may consider writing its own

guidelines for internal data structures. For example,the use or non-use of indices, whether data sets can be "spanned" across volumes and whether data compression or encryption should take place are typical of potential guidelines. There is some truth to this argument, but as relational technology matures, this perceived performance gap should narrow or disappear.

Conceptual or logical data level standards refer to the design and layout of logical tables of data. Table design involves a tradeoff between retrieval or loading speed versus the avoidances of unpleasant anomalies.

NASA might consider using guidelines for table design including a degree of table normalization, number and type of primary keys, and the support of foreign keys.

Interface or external standards refer to the manipulation of logical tables by on-line query languages or host-embedded languages. NASA might consider standards relating to: whether query and host-embedded languages be identical, how external views are to be supported and whether data updating through views should be allowed.

The only relational DBMS standard was developed by the X3H2 committee of the American National Standards Institute (ANSI). The standard uses IBM's SQL language as a base document.

SQL has already been adapted by a number of commercial systems as the Data Base Manipulation Language (DBML) or the query language for their relational data bases.

The SQL data dictionary is written in logical tables which may be retrieved and manipulated by ordinary SQL statements. This feature greatly facilitates the ease of DBMS administration and control.

The primary advantage of relational DBMS is that they free the programmer from the physical layout of the DBMS when manipulating data. Changes in the physical structure or location of data is hidden from the programmer. The programmer can concentrate his or her efforts on logical data relationships.

3.1.2.2 Network DBMS Standards

A standard referred to as CODASYL has been prepared for some network DBMS's. The conceptual and interface CODASYL Standards are well defined in the literature and consist of a Schema and Subschema or Schema Subset. Most commercial network DBMS's follow the terminology and concepts fairly close. Where implementation is too costly, they generally offer a large subset of the CODASYL Standard. Even if commercial CODASYL-like DBMS's do not follow these internal storage levels, NASA might consider issuing its own guidelines to simplify design effort and data base performance.

An internal and storage standard is called a storage schema under CODASYL, it is written in a Data Storage Description Language (DSDL).

The key points of the DSDL are:

1. Storage space is partitioned in disjoint storage areas which consists of an integral number of fixed-sized pages.

2. Record types defined at the Schema level can be represented by one or more storage record types. All occurrences of a given storage record type are stored in the same storage area. Record placement may be sequential, hashed ("CALC" mode) or "clustered".

3. Schema set types are represented by pointer chains or indexes.

4. Indexes can be used to provide additional access paths not exposed in the Schema.

3.1.2.3 Hierarchical DBMS

A hierarchical DBMS contains data physically stored in tree structures. Hierarchical DBMS can model complex network data models by the use of system pointers among trees.

Hierarchical DBMS do not have a standard. NASA might consider using IBM's IMS DBMS for standard terminology and basic concepts.

There is no internal level standard. NASA might consider issuing guidelines relating to:

- o fast-path access
- o the use of secondary indices

There is no conceptual level standard. NASA might consider using IBM's IMS terminology for logical and physical tables. Most hierarchical DBMS will not offer IMS's great variety of data constructs, but a common terminology is important.

3.1.2.4 Data Base Standards References

References

Date, C.J., "Some Principles of Good Language Design, with Special Reference to the Design of Data Base Languages," Sigmod Record, Vol. 14, No. 3, (November, 1984).

Date, C.J., "A Critique of the SQL Data Base Language," SIGMOD RECORD, Vol. 14, No. 3, (November, 1984).

Hobbs, Robert W., "Space Station Data Base Management Study," NASA Contract NAS5-27684, Task 1, (November, 1983).

Bryce, Milt, "Data Base Directions II: The Concersion Problem/Standards," SIGMOD RECORD, Vol. 12, No. 2, (January, 1982).

Bemer, Robert W., "Standards: A Data Base Imperative," Proceeding of the Workshop of National Bureau of Standards and the ACM, ACM, 1975.

Deutseh, R., "Progress Towards Data Base Management Standards," AFIPS Conference Proceedings, 1983 National Computer Conference.

Yao, S.B., et al "Data Base Systems," Selected Reprints in Software, IEEE Computer Society Press.

Champine, G.A., "Current Trends in Data Base Systems," Selected Reprints in Software, IEEE Computer Society Press.

Fry, James P. and Sibley, Edgar H., "Evolution of Data Base Management Systems," ACM Computing Surveys, Vol. 8, No. 1. (March 1976).

Date, C.J., An Introduction to Data Base Systems, Volume I, Addison-Wesley, (February, 1982).

Wiederhold, G.N., "Data Bases," computer, Vol. 17, No. 10, (October 1984).

Hecht, Herbert, "Computer Standard," Computer, Vol. 17, No. 10, (October, 1984).

CODASYL Systems Committee, "Feature Analysis of Generalized Data Base Management Systems," Technical Report, (May, 1971), ACM.

Tsichritzis, D.C. and Klug, A. (Eds), "The ANSI/X3/SPARC DBMS FRAMEWORK: Report of the Study Group on Data Base Management Systems," Information Systems 3, (1978).

Stonebraker, M.R., "A Functional View of Data Independence," Proc. 1974, ACM SIGMOD Workshop on Data Description, Access and Control.

3.1.3 Coding Standards Options

"It is the intent of NASA that this program be accomplished in a more cost-effective manner and with a significantly higher level of productivity than has been typical of prior major programs."[1] It is recognized that a significant effort (from 17% to 28% is typical[2]) of a program can be spent in the process of generating the source code. Therefore, rules that have demonstrated a productivity enhancement are attractive. Another and possibly much larger saving, can be realized in the life-cycle costs where analysis has indicated standardized coding

rules can simplify code integration, code sharing (re-use), and maintenance. As stated in the RFP, some of "The unique characteristics of the SSP...program growth;..." "customer friendly" "perspective; maintainability, commonality, and test and verification concepts; and the need for increased productivity"[3].

Because of the distributed contracting of the software effort on SSDS standard techniques for instilling discipline in the implementation task are critical for project wide management.

Two such standards are currently being developed by the IEEE Software Engineering Standard Subcommittee (SESS) are IEEE-SESS-828 (Configuration management), and IEEE-SESS-730 (Software Quality-Assurance Plans).

The scope of these standards deals with the specific areas of : source code generation (in any programming language), documentation of program source code, data structures (including COMMON's and INCLUDE files), data structures documentation, design approach and coding layout.

3.1.3.1 Bottom Up

This begins with the development of several computational routines whose function is considered important to the application. Once this is finished, the need is seen for a test driver to support testing of those modules and their interfaces. Sometimes the process involves modification of related routines in an attempt to avoid duplication.

The unique aspect of this approach is the immediate attention to the most difficult code segments.

Bottom up does not minimize life-cycle cost as its benefits are realized in the early stages of coding only. The lack of a system perspective tends to generate higher level control code and data structures that are not cohesive. This adversely affects maintainability because the programmers who must do the follow-on work have the additional difficult task of determining how the current code works before they can attempt to modify it.

The lack of early attention to interfaces and overall system requirements tend to make verification and validation very difficult.

This approach handles the critical (time or space) code in the earliest stages, thereby affording the maximum opportunity to correct bad assumptions and it minimizes their ill effects. Also, this approach encourages the creation and use of reusable modules, but integration of these modules can be a problem due to the lack of a system viewpoint.

By not coding from a overall system viewpoint, many assumptions can be made with inadequate attention to consequences

in areas such as commonality of different modules and migration of functions.

3.1.3.2 Top Down Stub

The designer begins by determining what overall functions will be performed. He/she then makes a working top-level module containing all the logic controlling the sequencing between functions, but inserts dummy sub-programs (stubs) for these functions. Finally, he/she tests this executive program thoroughly before proceeding to succeeding steps consisting of fleshing out the stubs. It is important to note that each sub-program can be tested immediately, not only by itself, but as a part of the software system it joins by replacing the stub. HIPO (Hierarchical Input Process Output) is one optional aid. Another is the Walk-Through, an element of "egoless programming," intended to eliminate as many bugs as possible before coding begins.

In this approach, coding proceeds in a hierarchical manner with the most abstract control and human interface issues addressed first.

The system view is critical for the long life required by the Space Station Program. The evolution of software and hardware will only be possible if the original designs and implementations of the code are created with the necessary overall objectives constantly in mind.

Traceability from the original requirements down through all the levels is automatically generated from the top level control structures. This built-in road map helps direct the maintenance crew to the problem areas.

No special test programs need be written (and tested), the top level control root is generated and debugged first. As each stub is completed, it is simply added to the root and its new functionality tested. As long as the software configuration is kept under control, integration and test becomes adding one function at a time into a running system aiding visibility and tracking.

Software progress can be tracked with tangible milestones, the start and completion dates of the root control structure and each stub. Because there is a running system to use as the final test of the interfaces and functionality of each stub, as they are announced, no additional testing within
the system milestones are necessary.

Early integration and definition of interfaces is a major advantage of this method. High-risk modules are not necessarily identified early enough to avoid bad assumptions that may require different approaches and the rewriting of significant amounts of code. Common modules may not be identified and functionally equivalent modules may be duplicated under different stubs.

Unforeseen changes in top level requirements may require massive rewriting of data structures and code.

3.1.3.3 Top Down Problem Statement

In this approach, an attempt is made to define a special problem domain and establish a set of constructs and capabilities for expressing, analyzing, and generating programs. Another variant involves the use of iteratively refined simulations as successive problem statements which drive program design, ISDOS, SODAS, DES, and LOGOS are software tools used with this approach.

In this approach, a problem statement language and its processor attempt to build a set of constructs to analyze the code and data structures required.

No general statement language has been developed that is currently broad enough to support the Space Station Program. A number of tools have been developed[4] to abstract the design effort into general classes of steps that can be automated.

The automated tools ensure consistent documentation and approach but require another learning curve for new users. Formal data structuring enhances maintainability with improved visibility and tracking.

3.1.3.4 Model-Driven

This begins with two models. One is a "process model" of the software design and development, the other is a "properties manual" of the characteristics of a good software product. Note these models initially are independent of the specific capabilities which are required. The process model specifies the overall sequence of activities, which begin with the specifications of another model that is problem specific. This new model is checked via the checklists derived from the properties model. The resulting functional model is then used to generate a top-level system model of four parts; control structures model, data base model, performance model, and a work breakdown structure.

Process and a properties models are developed first to define the projected module. A process of step-wise iteration of first one then the other with walk-throughs and protocols results in a functional model that is used to generate a top level system model.

This option attempts to ensure the generation of the most appropriate top level control program. All the required information needed for the models may not be available in the Space Station Program. The limited problem domain of this approach may exclude this method.

The model driven approach has the following charasteristics. It emphasizes design validation and explicit risk analysis yet requirements/design/simulation languages must be acquired and maintained. This approach avoids premature hardware commitment and emphasizes early integration of functions. On-line analysis and design aids when available may improve productivity after they are learned.

3.1.3.5 Structured Programming

The key ideas are: single entry and exit to all modules; only three basic control structures allowed (sequential, do while, if then else); programs are organized in a hierarchical, modular block structure; formal semantic rules (such as variable declaration, indenting instructions and commentary).

Nassi-Schneidermann (NS) charts are diagrams of the flow of a program that tries not to allow unstructured methods[8].

This option is a more specifically coded methodology rather than the coding design option previously discussed. It attempts to formalize those methods that have proven to generate correct and maintainable code with less manpower then the ad hoc approaches common to coding.

A great deal of code will be generated by a large number of organizations in many remote locations for SSDS. To ensure the maximum production of "good" code, a discipline of proven methods makes sense.

The hallmark of this option is maintainability. The code should be a clear and understandable as possible (i.e., the GOTO is avoided because it tends to confuse the flow of control through the code).

Because the code generated by this option is more easily understood, testing becomes more systematic. Also, the single entry single exist constraint on all modules ensures better test coverage.

Studies[5] have indicated that this option can contribute significantly to software productivity. Formal modularity eases the division of tasks and supports integration. Disadvantages include poor control structure for asynchronous events and error exists. In addition, common or reusable components may not be valid and a strict adherence to the standard would require some code duplication. Also, this approach does not successfully address the problem of common data structures.

3.1.3.6 Prototyping vs. Specifying

Traditionally, there has been two approaches to specifying what a code module will be required to do: Build and Fix - Proceed to build the full system with minimal or fuzzy specifications when rework and patch until it satisfies the

customer (or can't be maintained); and Formal Specifying - Develop a design specificaton document to follow for implementation then rework the product until the customer is happy. A new (for software) method is to build a prototype system from the usual minimal or fuzzy (strawman) specifications that the customer can provide. This prototype is allowed to be a partial implementation stressing the interfaces rather then the guts of the system. A formal exercise of the prototype with the customer allows the filling in of the strawman specifications in light of actual experience. An experiment[6] was run comparing these methods as applied to seven teams of graduate students at UCLA. The results indicate that for roughly equivalently performing systems, the prototyping approach needed 40% less code and 45% less effort (man hours), but specifying produced more coherent designs and code that was easier to integrate. The teams that used the specifying approach stated that it was very easy to overpromise in their specifications (talk is cheap), in contrast the prototyping teams fostered a higher threshold for incorporating marginally useful functions as they had a more realistic feel for the amount of effort required. Additionally, the maintainability of the prototyping teams system was rated remarkably higher. Finally, it was observed that none of the prototyping teams started fresh after the formal exercise, rather 67-95% of the prototype's code ended up in the final system and the prototype was 40-60% the size of the final system.

This method addresses the thorny issue of specifications in an old and reasonable manner. In most areas of engineering, a prototype is intentionally produced to test concepts and interfaces in realization that the full scope of the program is not apparent at its start and false assumptions may lead to bad choices. Software engineering has not used this approach. It is long overdue.

The Space Station Program is so large that the effort to fully specify is not possible. Fuzzy requirements can only lead to misunderstandings and ambiguities. This option allows uncertain specifications to be resolved at the format prototype exercises (User Design Review and Exercise UDRE).

The experiment[6] indicated this method produced a more maintainable but less coherent design. More effort was needed proportionally in testing and fixing.

This method generated much better human-machine interfaces and functionality as the users get a hand in the operational analysis at a stage where their input can have an impact.

The experiment[6] indicated a reduction of the deadline effect at the end of the project. The prototype provides a working base that can demonstrate progress clearly.

A more responsive to the customer design is the thrust of this option. Always having something that works to build upon provides a great deal of visibility into the progress of the

program. A penality is more difficult during the integration stages and a potentially less coherent design.

3.1.3.7 References

Chevers, E.S. "Program Management of High Technology S/W and H/W for the Proposed Space Station."

Dijkstra, E.W. "A Discipline of Programming."

Freeman, P. "Software Design Techniques 3rd Edition IEEE."

Gildersleeve, T.R. "Successful Data Processing System Analysis."

Hart, D.A. "The Effects of ADA and the Intel IAPX432."

Horowitz, H. "Practical Strategies for Developing Large Software Systems."

Whichmann, B.A. "A Comparison of Pascal and ADA."

Yourdon, E. "Techniques of Program Structure and Design."

[1]SPACE STATION DEFINITION and PRELIMINARY DESIGN REQUEST FOR PROPOSAL - SECTION B - PROGRAM COST AND PRODUCTIVITY.

[2] B. W. Boehm, "The High Cost of Software" Practical Strategies for Developing Large Software Systems, ed. E. Horowitz, (Addison-Wesley 1975), page 7.

[3]SPACE STATION DEFINITION and PRELIMINARY DESIGN REQUEST FOR PROPOSAL - SECTION B - SIGNIFICANT FEATURES OF THE RFP.

[4]B. W. Boehm, "Some Steps Toward Formal and Automated Aids to Software Requirements Analysis and Design." TRW Systems Group, Redondo Beach, CA Nov. 1972.

[5]E. W. Dijkstra, "Notes on Structured Programming," Structured Programming, O. J. Dahl, E. W. Dijkstra, C.A.R. Hoare, Academic Press, London 1972.

[6]B. W. Boehm, T. E. Gray, T. Seewaldt, "Prototyping vs.. Specifying: A Multi-Project Experiment," Proceedings 7th Software Engineering, Orlando, FLA, Mar. 1984.

[7]Loughhead, J.N., Miles, G.A., and Turner, J.R., Practical Experience of Engineering Software Production to a Structured Programming Standard, "Proceedings of ESA/ESTEC Seminar Noordwijk, Netherlands 11-14 Oct. '83," p.125.

[8]Loughhead, J.N., Miles, G.A., and Turner, J.R., Practical Experience of Engineering Software Production to a Structured Programming Standard, "Proceedings of ESA/ESTEC Seminar Noordwijk, Netherlands 11-14 Oct. '83," p.127

3.1.4 Software Quality Assurance and Testing Standards

Software Quality Assurance is a monitor and control function whose objective is to ensure that design, implementation, integration, testing and maintenance standards and approved practices are established and followed throughout the software development cycle. SQA tasks are intended to provide a discipline for monitoring software development from identification of requirements to software end products that correctly meet these requirements. This paper addresses the role

of SQA during the testing phase of a software project. Testing can be defined as "the process of executing a program with the intent of finding errors" (G.J. Myers). Testing activities include the definition of test requirements, the definition of test-plans and specifications, module or unit testing, integration testing, system testing, acceptance testing and installation testing. SQA participation in software testing encompasses reviewing test documentation, monitoring, witnessing the conduct of the test, identifying problems during the test and verifying and approving test results.

The options for implementing a software quality assurance function as part of the testing phase of software development include: application of the following standards and methodologies:

- o Military Specifications and Standards
- o NASA Guidelines and Federal Information Processing Standards (FIPS)
- o Industry Methodologies

3.1.4.1 MIL-S-52779

MIL-S-52779, Software Quality Assurance Program Requirements prescribes the requirements for the establishment and implementation of a Software Quality Assurance program by contractors performing on a government software development contract. Other standards which could be used/referenced by SQA include:

- o MIL-STD-1521A, Technical Reviews and Audits for Systems, Equipment and Computer programs.
- o MIL-STD-490, Specification Practices
- o MIL-STD-483, Configuration Management Practices for Systems, Equipment, Munitions, and Computer Programs.
- o MIL-STD-1456, Contractor Configuration Management Plans.

The functions of SQA related to software testing, as specified by MIL-S-52779, include the following:

a. Determining the testability of software requirements.
b. Review of test plans for compliance to contract requirements and appropriate standards.
c. Review of test requirements and criteria for adequacy, feasibility, and software requirements traceability.
d. Review of test procedures for compliance to contract requirements and appropriate standards.
e. Monitoring of tests and approval of test results.
f. Review and approval of test reports.
g. Ensuring that test documentation is maintained to allow test repeatability.

h. Assuring that software development tools (i.e.,

support software and hardware) have been identified and are acceptable.

Other functions related to the aforementioned are described in MIL-STD-1521A, including participating in the following:

a. Functional Configuration Audit (FCA) - a formal audit to verify that the computer program configuration item's actual performance complies with its Development Specification.
b. Physical Configuration Audit (PCA) - a formal examination of the "as built" version of a computer program configuration item (CPCI) to verify conformance to the documentation defining the CPCI.
c. Formal Qualification Review (FQR) - a review to ensure that testing has been accomplished to verify that the CPCI's actual performance complies with its Development Specification.

MIL-S-52779 and MIL-STD-1521A are policy documents which have been used effectively on large software contract.

A proposed Military Standard on Software Quality Assessment and Measurement, MIL-STD-SQAM, has been prepared but have not been approved; this standard contains requirements for a SQA program for mission critical software development.

SQA can help ensure the development of correct, error-free software thus reducing lifecycle costs.

MIL-STD-1521A (USAF), Technical Reviews and Audits for Systems, Equipments, and Computer Programs, DOD, 1 June 1976.

McCabe, T. J., Software Quality Assurance: A Survey, Thomas J. McCabe & Associates, Inc., 1980.

Foster, R> A., Introduction To Software Quality Assurance, 3rd Ed., 1978.

MIL-S-52779(AD), Software Quality Assurance Program Requirements, DOD, 5 April 1974.

3.1.4.2 NASA Guidelines and Federal Information Processing Standards (FIPS)

The document, NASA Software Management Guidelines, provides procedure guidance for the management of NASA software projects. The guidelines presented provide a generic model which can be selectively applied to any NASA software development effort. The functions of SQA related to software testing, as presented in the referenced document, include the following:

a. Reviews/Audits to ensure verification standards have been met.
b. Reviews/Audits to ensure approved test procedures have been performed.
c. Participation in software test design to ensure that all software end items satisfy design, operational, and functional requirements.

FIPS Publication 101, Guideline for Lifecycle Validation, Verification, and Testing of Computer Software, presents a methodology for validation, verification, and testing (VV&T) to be used throughout the software lifecycle. This document describes VV&T activities and products which relate to and provide more detail for the SQA functions listed above.

References

References are:

"NASA Software Management Guidelines"

National Bureau of Standards, Guideline For Lifecycle Validation, Verification, and Testing of Computer Software, Federal Information Processing Standards Publication 101, June 1983.

3.1.4.3 Industry Methodologies

Industrial organizations have recognized the need for software quality assurance; however, the implementation of the function has been weak due to a lack of established methodologies. Most companies have developed their internal policies and standards in response to government requirements. Thus, the functions of SQA during software testing remain the same as specified by MIL-S-52779. One difference in methodologies is the extent of SQA participation during specific tests. Most organizations have adopted the following:

- Development testing (Unit, Integration, and System) will be selectively audited with no test witnessing by SQA.

- Qualification testing (Software Validation/Customer Acceptance) will be controlled and witnessed by SQA.

Another difference is the role of SQA during test documentation preparation. In some organizations, SQA writes all test plans and procedures to be used during qualification testing.

Industry methodologies are evolving and have been used effectively on large software contracts.

3.1.4.4 References

Dunn, R. and Ullman, R., Quality Assurance for Computer Software, McGraw-Hill, Inc., 1982.

Foster, R. A., Introduction to Software Quality Assurance, 3rd Ed., 1978.

McCabe, T. J., Software Quality Assurance, A Survey, Thomas J. McCabe & Associates, Inc., 1980.

Myers, G. J., The Art of Software Testing, John Wiley & Sons, Inc., 1979.

Electronic Industries Association, SQA Panel Report, "Implementation Response to Government Software Quality Requirements," October 1980.

Fogel, G. D., "The Proofing of Data Processing Systems," Presentation, National Conference on Testing Computer Software, October 1984.

3.1.5 Hardware Standards Options

A large variety of hardware standards exists for various levels of hardware from cabinet, chassis, board, to harnessing etc.. Standardization of hardware is highly desirable in order to minimize change-over effects to reconfiguration.

In general, the standards would vary for space and ground components. This standard would be applied for cabinets, chasses and card guides. The card guide method has an unlimited variety in commercial equipment as with cabinets and chassis. Typically, cabinet standards refer to dimensions, cooling capabilities, material construction and finish. Chassis level packaging standards usually relate to dimensions, weight, cooling and shock/vibration resistance. Chassis specifications are usually determined by circuit board design. Card specification usually results in specifications of the motherboard/connector and harness/connector interfaces. Weight, power and signal integrity are the principle items specified. In the case of the mother- board, material selection is usually the same as the circuit board and connector selection.

Circuit board standards are usually established by either Milspec or individual company standards. The board dimensions (X, Y, Z) are generally governed by design goals. The density demands will determine the Z dimensions - two-sided/multi-layer up to twenty one layers. The tradeoff between density and maintainability is exercised as density increases with the high multi-layer count which makes repair more difficult. The material is usually specified by standards based on the desired electrical and mechanical characteristics (flexible versus non-flexible, paper base epoxy glass base to ceramic base) NASA, Mil, and federal standards and specifications provide a wealth of

possibilities on selecting materials.

Standards exist for specifying harnessing, wire-type and cabling standards. Such factors as size and weight restrictions, wire and/or signal requirement, material makeup (copper, aluminum, silver finish....), insulation, etc., are generally established by standard selections.

The following references indicate the breadth of options available for hardware standardization.

REFERENCES

Metal Standards

QQ-S-698	Steel, sheet and strip low carbon
QQ-W-321	Wire, Copper alloy
QQ-B-728	Bronze, phospher
QQ-S-763	Steel Bars, Wires, Shapes and Forging CRES.
ASTM-B348	Titanium and titanium alloys
MIL-HDK-5C	Metallic materials and elements for Aerospace vehicle structures
MIL-STD-22	Welded joint design
AWS-A.2.4	Symbols for welding and nondestructive testing.
AWS-A.3.0	Welding terms and definitions
AWS-D.1.1	Structural welding code -steel
ANSl-SR-17	Metric mechanical fasteners
ANSl-B3610-197	Welded and seamless steel pipe.
ASTMA53-75	Welded and seamless steel pipe.
MIL-STD-188	Dissimilar metals
MIL-T-23103	Thermal performance evaluation airborne

SURFACE TREATMENTS AND INORGANIC COATINGS FOR STEEL

QQ-C-320	Chromium Plating (Electrodeposited)
QQ-I-716	Iron and Steel; Sheet, Zinc Coated (Galvanized)
QQ-N-290	Nickel Plating (Electrodeposited)
QQ-P-35	Passivation Treatments for Corrosion-Resisting steel
QQ-P-416	Plating, Cadmium (electrodeposited)
QQ-S-365	Silver Plating, Electrodeposited, General Requirements for
QQ-T-181	Terne Plates (for Manufacturing Purposes)
QQ-T-425	Tinplate (Electrolytic)
TT-C-490	Cleaning Methods and Pretreatment of Ferrous Surfaces for Organic Coatings.
MIL-STD-171	Finishing of Metal and Wood Surfaces
MIL-HDBK-132	Protective Finishes
MIL-A-40147	Aluminum Coating (Hot-Dip) for Ferrous Parts
MIL-C-8837	Coating, Cadmium (Vacuum Deposited)
MIL-C-13924	Coating, Oxide, Black, for Ferrous Metals
MIL-C-14550	Copper Plating (Electrodeposited)
MIL-C-26074	Coatings, Electroless Nickel, Requirements for
MIL-C-81562	Coatings, Cadmium, Tin-Cadmium and Zinc (Mechanically Deposited)
MIL-C-81751	Coating, Metallic-Ceramic
MIL-F-14072	Finishes for ground electronic equipment
MIL-G-45204	Gold plating, electrodeposited
MIL-L-13762	Lead alloy coating, hot dip (for iron and steel parts)
MIL-L-13808	Lead plating, electrodeposited
MIL-M-6874	Metal spraying, process for
MIL-P-14535	Plating, Black Nickel (Electrodeposited)

MIL-P-14538	Chromium Plating, Black (Electrodeposited)
DOD-P-16232	Phosphate Coatings, Manganese or Zinc Base (For ferrous metals)
MIL-P-16961	Porcelain Enamel Coating for Steel Mufflers of Internal Combustion Engines
MIL-P-20218	Chromium Plating, Electro-deposited, Porous.
MIL-P-23408	Plating: Tin-Cadmium (Electrodeposited)
MIL-P-45209	Palladium Plating, Electrodeposited
MIL-P-81728	Plating, Tin-Lead (Electrodeposited)
MIL-R-46085	Rhodium Plating, Electrodeposited
MIL-S-5002	Surfaces Treatments and Inorganic Coatings for metal surfaces of weapons systems.
MIL-T-10727	Tin Plating: Electrodeposited or Hot-Dipped, for Ferrous and nonferrous metals.
ASTM-A-153	Zinc coating (hot-dip) on iron and steel hardware.
ASTM-B-183	Preparation of low carbon steel for electroplating, Practice for
ASTM-B-254	Preparation of and electroplating on stainless steel, Practice for
ASTM-B-633	Electrodeposited coatings of zinc on iron and steel, Specification for

INORGANIC COATINGS AND FINISHES FOR ALUMINUM AND ALUMINUM ALLOYS

QQ-C-320	Chromium Plating (Electrodeposited)
QQ-N-290	Nickel Plating (Electrodeposited)
QQ-P-416	Plating, Cadmium (Electrodeposited)
QQ-S-365	Silver Plating, Electrodeposited, General Requirements for
QQ-T-181	Terne Plates (for Manufacturing Purposes)
QQ-T-425	Tinplate (Electrolytic)
MIL-STD-171	Finishing of Metal and Wood Surfaces
MIL-HDBK-132	Protective Finishes
MIL-A-8625	Anodic Coatings, for Aluminum and Aluminum Alloys
MIL-C-5541	Chemical Conversion Coatings on Aluminum and Aluminum Alloys
MIL-C-8837	Coating, Cadmium (Vacuum Deposited)
MIL-C-14550	Copper Plating (Electrodeposited)
MIL-C-26074	Coatings, Electroless Nickel, Requirements for
MIL-C-81562	Coatings, Cadmium, Tin-Cadmium and Zinc (Mechanically Deposited)
MIL-C-81751	Coating, Metallic-Ceramic
MIL-F-14072	Finishes for Ground Electronic Equipment
MIL-G-45204	Gold Plating, Electrodeposited
MIL-L-13808	Lead Plating, Electrodeposited
MIL-M-6874	Metal Spraying, Process for
MIL-P-14535	Plating, Black Nickel (Electrodeposited)
MIL-P-14538	Chromium Plating, Black (Electrodeposited)
MIL-P-20218	Chromium plating, Electrodeposited, Porous
MIL-P-23408	Plating: Tin-Cadmium (Electrodeposited)
MIL-P-45209	Palladium Plating, Electrodeposited
MIL-P-81728	Plating, Tin-Lead, (Electrodeposited)

MIL-R-46085	Rhodium Plating, Electrodeposited
MIL-S-5002	Surfaces Treatments and Inorganic Coatings for Metal Surfaces of Weapons Systems
MIL-T-10727	Tin Plating: Electrodeposited or Hot-Dipped, for Ferrous and nonferrous Metals
ASTM-B-253	Preparation of and Electroplating on Aluminum Alloys by the Zincate Process, Practice for
ASTM-B-449	Chromate Treatments on Aluminum, Practice for
AMS-2468	Hard Coating Treatment of Aluminum Alloys
AMS-2469	Process and Performance Requirements for Hard Coating Treatment of Aluminum Alloys
AMS-2470	Anodic Treatment of Aluminum Alloys, Chromic Acid Process
AMS-2471	Anodic Treatment of Aluminum Alloys, Sulfuric Acid Process, Undyed Coating
AMS-2472	Anodic Treatment of Aluminum Base Alloys, Sulfuric Acid Process, Dyed Coating
AMS-2473	Chemical Treatment for Aluminum Alloys, General Purpose Coating
AMS-2474	Chemical Treatment for Aluminum Alloys, Low Electrical Resistance Coating

GENERAL STANDARDS

MIL-STD-470	Maintainability program Req.
MIL-STD-471	Maintainability Demonstration
MIL-HDBK-217B	Reliability prediction of Elect. Equip.
MIL-STD-721	Definition of Effectiveness terms for Reliability, Maintainability, Human Factors, and Safety.
MIL-HDBK-472	Maintainability prediction
MIL-STD-415	Test pts. and test facilities for elect. sys. and associated equip. design STD for.
MIL-STD-481	Electromagnetic interference characteristics requirements for equipment.
MIL-W-5088	Wiring, Aerospace Vehicle
MIL-STD-1353	Electrical connectors and Associated Hardware, Selection and use of.
DOD-STD-1678	Fiber optic test methods and instrumentation.
EIA-RS440	Fiber optic connector terminology
DOD-C-85045	Cables, Fiber Optics, Gen. Spec. for
MIL-STD-1472	Human Engineering Design Criteria for Military Systems, Equipment and Facilities.
MIL-STD-171	Finishes for metals and wood.
MIL-STD-189	Rocks, Elect. Equip. 19 inch and associated panels.
EIA-RS-310	Racks, panels, and associated equip.
FED-STD-595	Colors
MIL-M-13949	Plastic sheet, laminated copper, Dual glass-base epoxy.
MIL-STD-1313	Micro circuit terms and definitions.
MIL-STD-130	Ident. marking of U.S. Mil. property.
MIL-F-14072	Finishes for ground sig. equip.
MIL-P-8585	Primer coating, zinc chromate, low-moisture sens.
MIL-STD-285	Attenuation measurement for enclosures

MIL-STD-202	Test methods for electronic and electrical component parts.
ANSl-Y32.14-1973	ANSI STD. Graphic symbols for Logic Diagrams
ANSl-Y14.5M-1982	Dimensioning and Tolerancing
MIL-STD-454	Sstd. Gen. Req. for electronic equipment.
MIL-E-4158	Elect.Equip., Ground, General Req. for
ANSI/IEEE-STD-100-1979	Definition of Elect. terms (IEEE)(American National Standards Institute).
Standard 142	Grounding of industrial power systems (IEEE)
MIL-B-81705	Barrier materials; flexible electrostatic-free, Heat shrinkable.
DOD-STD-1686	Electrostatic Discharge Control program for protection of electrical and electronic parts. Assemblies and equipment.
DOD-HDBK-263	Electrostatic discharge control handbook.

DIGITAL DATA STANDARDS

EIA-RS-232	Interface between data terminal and communication equipment.
EIA-RS-42	Electrical Characteristics of balanced voltage digital interface circuits.
EIA-RS-423	Electrical characteristics of unbalanced voltage digital interface circuits.
EIA-RS-449	General purpose 37 position and 9 position interface for data terminal equipment, serial binary data interchange.
IEEE-488	Digital interface for programmable instrumentation
IEEE-583	Modular instrumentation and digital interface system (CAMAC)
IEEE-595	IEEEStandard Serial Highway Interface System (CAMAC)
IEEE-596	IEEE Standard PArallel Highway Interface System (CAMAC)
IRIG-123-72	Instrumentation timing systems brochure
IRIG-104-70	IRIG Standard time formats
IRIG-106-60	Telemetry standards
MIL-STD-188	Military Comm. System Tech. Std.
USAS-X3.4.1977 (ANSI)	Info. Intechange, Code for

Organic Finishes Reference Documents

FED STD 595	Colors
MIL-STD-171	Finishing of Metal and Wood Surfaces
MIL-HDBK-132	Protective Finishes
TT-E-489	Enamel, Alkyd, Gloss (For exterior and Interior Surfaces)
TT-P-1757	Primer Coating, Zinc Chromate, Low-Moisture-Sensitivity.
MIL-C-5410	Cleaning Compound, Aluminum Surface, Non-Flame-Sustaining
MIL-C-5541	Chemical Conversion Coatings on Aluminum and Aluminum Alloys
MIL-C-8514	Coating Compound, Metal Pretreatment, Resin-Acid.
DOD-P-15328	Primer(Wash), Pretreatment (Formula No. 117 for Metals)
MIL-C-43616	Cleaning Compound, Aircraft Surface
DES 7.11.4	Tropicalizing (Fungus Proofing)
88-3102	Painting, Varnishing and Lacquering

Steel Structure near-White Blast Clearing Painting.

Council SSPC-10

Guidelines for selection of organic finishes for military using systems are presented in MIL-STD-171, Type I exposure. Cured finishes shall conform to the requirements of manufacturing Standard 88-3102.

3.1.6 Safety Standard Options

Currently, NASA Shuttle Programs tailor their safety standards and regulations from the two standards mentioned above. It is, therefore, reasonable to expect a separate NASA Handbook (NHB) specifically related to all segments of the Space Station Program.

Contractor Program Plans for individual NASA sites evolve from a tailoring process which begins with a baseline Military Standard on the specific area and ends with a Program Plan for individual contractors. Initially, a general Military Standard is chosen as a baseline reference document for each discipline. Specific design guidelines and specifications in the standard are then tailored to the tasks that need to be described in the NASA handbook for an overall Program in NASA, such as the Space Shuttle Program, or the Space Station Program. The Contractor Program Plans are, in turn, tailored from the NASA Handbook in accordance with specific needs at each NASA site.

Since this has been the method NASA has utilized for other program standards, it is reasonable to expect them to follow suite with respect to safety standards for the Space Station Program. Therefore, it is recommended that SSDS limit its comparison of safety standard options to MIL-STD-882 and NHB 5300.4(1D-2).

A preferred option is one that consolidates the comprehensiveness of MIL-STD-882 with NHB 5300.4(1D-2) elements 1D200-4 (Organization), 1D200-8 (Mishap Investigation and Reporting), 1D200-9 (Risk Management), 1D201-6 (Hazard Reduction Precedence Sequence), and 1D201-9 (Human Engineering). A preferred option is one that also incorporates specific tasks during the operations and support phases to assure sustained safety of systems through the operational life period and to forecast timely replacement/refurbishment prior to excessive degradation in safety.

3.1.6.1 Mil-Standard 882

This document provides uniform requirements for developing and implementing a system safety program of sufficient comprehensiveness to identify the hazards of a system and to impose design requirements and management controls to prevent mishaps by eliminating hazards or reducing the associated risk to an acceptable level. Selective application and tailoring of this military standard must be accomplished to specify the extent of contractual and in-house compliance.

This standard provides uniform requirements for developing and implementing a system safety program to identify the hazards of a system and to impose design requirements and management control to prevent mishaps by eliminating hazards or reducing the associated risk to a level acceptable to the Managing Activity (MA).

Tasks described in this standard are to be selectively applied in contract-definitized procurement,requests for proposal (RFP), statement of work (SOW), and Government in-house developments requiring system safety programs for the development, production, and initial deployment of systems, facilities, and equipment.

Task Descriptions are to be tailored as required by the MA governing regulations and as appropriate to particular systems or equipment program type, magnitude, and funding.

Application guidance and rationale for selecting tasks to fit the needs of a particular system safety are included in Tables 3.1.6-1, 3.1.6-2, and 3.1.6-3 in the appropriate Data Item Description (DID) and statements of work.

Current use of this standard is with Revision B which went into effect 30 March 1985. The first revision, A, was released on 27 June, 1977.

Specific referenced documents required to supplement this military standard are called out in the following DIDS:

DI-H-7047A	DI-H-7048A
DI-H-7049A	DI-H-7050A

3.1.6.2 NHB 5300.4(1D-2), SAFETY, RELIABILITY, MAINTAINABILITY, AND QUALITY PROVISIONS FOR THE SPACE SHUTTLE PROGRAM

This NASA Handbook (NHB) establishes common safety, availability, maintainability, and quality provisions for the Space Shuttle Program. NASA Centers shall use this publication both as the basis for negotiating safety, reliability, maintainability, and quality activities at the Centers.

This NHB provides common safety provisions for the Space Shuttle Program to individual NASA Centers to assure that applicable provisions of this NHB are imposed in lower tier contracts.

Applicable safety requirements and tasks shall be included in the basic management systems, design verification documents, overall system analysis, system engineering requirements definition, and design review practices.

Application guidance and rationale for selecting tasks to fit the needs of a particular safety program are discussed in the following sections of the NHB:

a. ID200, SAFETY MANAGEMENT
b. ID201, SYSTEM SAFETY
c. ID202, INDUSTRIAL SAFETY
d. ID203, TEST OPERATIONS SAFETY

This standard was originally released in August, 1974 as 5300.4(1D-1) and revised in October, 1979 as an administrative updating of 5300.4(1D-1) to incorporate changes approved by the Program Director.

This NH incorporates provisions of NASA documents:

NHB 1700.1, NASA SAFETY MANAUAL, VOL. 1
NHB 5300.4 (1A), RELIABILITY PROGRAM PROVISIONS FOR AERONAUTICAL AND SPACE SYSTEM CONTRACTORS
NHB 5300.4 (1B), QUALITY PROGRAM PROVISIONS FOR AERONAUTICAL AND SPACE SYSTEM CONTRACTORS.

3.1.7 Reliability Standard Options

The options associated with Reliability Standards apply to the extent that existing standards will be used and if so, to what extent each may be employed. This options report is limited to the two most widely used Reliability Standards in military and/or NASA space programs. These two Reliability Standards are:

a. MIL-STD-785B, RELIABILITY PROGRAM FOR SYSTEMS AND EQUIPMENT
b. NHB 5300.4(1D-2), SAFETY RELIABILITY, MAINTAINABILITY AND QUALITY PROVISIONS FOR THE SPACE SHUTTLE PROGRAM

3.1.7.1 MIL-STD-785B, RELIABILITY PROGRAM FOR SYSTEMS AND EQUIPMENT

This standard consists of basic application requirements, specific reliability program tasks, and application, guidance and rationale for task selection. This standard is structured to discourage indiscriminate blanket application. Tailoring is forced by requiring that specific tasks be selected and that certain essential information relative to implementation of the task be provided by the procuring activity. The tasks can be tailored to meet specific program needs for space and terrestrial segments.

This standard provides general requirements and specific tasks for reliability programs during the development, production, and initial deployment of systems and equipment.

Tasks described in this standard are to be selectively applied in contract-definitized procurements, request for proposals, statements of work, and Government in-house developments requiring reliability programs for the development, production, and initial deployment of systems and equipment.

Task descriptions are intended to be tailored as required by governing regulations and as appropriate to particular systems or equipment program type, magnitude, and funding.

Application guidance and rationale for selecting tasks to fit the needs of a particular reliability program is included in Table 3.1.7-1.

This standard was revised to 785A in March 1969 and to 785B in September, 1980.

3.1.7.2 NHB 5300.4 (1D-2); SAFETY, RELIABILITY, MAINTAINABILITY AND QUALITY PROVISIONS FOR THE SPACE SHUTTLE PROGRAM.

This NASA Handbook (NHB) establishes common safety, reliability, maintainability, and quality provisions for the Space Shuttle Program.

The reliability provision in NHB 5300.4(1D-2) sets forth reliability task requirements. Shuttle Program contractors utilize the reliability provision as a guideline for program reliability conduct. Furthermore, NASA centers use the reliability provisions as a basis for negotiating reliability tasks with Shuttle Program contractors.

Tasks described in the NHB reliability provision are applied as an integral part of the design and development process. The design and development process includes the evaluation of hardware reliability through analysis, review and assessment. These tasks are also intended to be tailored as required by, and in accordance with, the applicable Data Requirement Documents (DRD).

TABLE 3.1.7-1 APPLICATION MATRIX FOR SYSTEM PROGRAM DEVELOPMENT

TASK	TITLE	TASK TYPE	PROGRAM PHASE CONCEPT	VALID	FSED	PROD
100	System Safety Program	MGT	G	G	G	G
101	System SAfety Program Plan	MGT	G	G	G	G
102	Integration/Management of Associate Contractors, Sucontractors and All firms	MGT	S	S	S	S
103	System Safeaty Program Reviews	MGT	S	S	S	S
104	SSG/SSWG Support	MGT	G	G	G	G
105	Hazard Tracking and Risk Resolution.	MGT	S	G	G	G
106	Test and Evaluation Safety	MGT	G	G	G	G
107	System Safety Progress Summary	MGT	G	G	G	G
108	Qualifications of Key System Personnel	MGT	S	S	S	S
201	Preliminary Hazard List	ENG	G	S	S	N/A
202	Preliminary Hazard Analysis	ENG	G	G	G	GC
203	Susystem Hazard Analysis	ENG	N/A	G	G	GC
204	System Hazard Analysis	ENG	N/A	G	G	GC
205	Operating and Support Hazard Analysis	ENG	S	G	G	GC
206	Occupational Health Hazard Assessment	ENG	G	G	G	GC
207	Safety Verification	ENG	S	G	G	S
208	Training	MGT	N/A	S	S	S
209	Safety Assessment	MGT	S	S	S	S
210	Safety Compliance Assessment	MGT	S	S	S	S
211	Safety Review of ECP's and Waivers	MGT	N/A	G	G	G
212	Software Hazard Analysis	ENG	S	G	G	GC
213	GFE/GFP System Safety Analysis	ENG	S	G	G	G

NOTES:

TASK TYPE

ENG - System Safety Engineering
MGT - Management

PROGRAM PHASE

CONCEPT - Conceptual

VALID - Validation

FSED - Full-Scale Engineering Development

PROD - Production

APPLICABILITY CODES

S - Selectively Applicable

G - Generally Applicable

GC - Generally Applicable to Design Changes only

N/A- Not Applicable

C-2

Application guidance and rationale for selecting tasks that satisfies the needs of a particular reliability program is included in applicable DRD's and statements of work.

NHB 5300.4(1D-2) was revised to the current NHB 5300.4 (1D-2) in October 1979. As such, NHB 5300.4(1D-1) was canceled in October 1979.

NHB 5300.4(1D-2) reliability provision incorporates information from NHB 5300.4(1A), RELIABILITY PROGRAM PROVISION FOR AERONAUTICAL AND SPACE SYSTEM CONTRACTORS.

3.1.8 Logistics Standards Options

Options associated with Logistics relate to the extent that the Logistic Support Analyses (LSA) guidelines and requirements are applied to the Space Station Program.

LSA guidelines and requirements are established by Department of Defense (DOD) Instruction 5000.2, Major System Acquisition Procedures, and DOD Directive 5000.39, Acquisition and Management of Integrated Logistic Support for Systems and Equipment. The requirements of this standard are applicable to major and less-than-major system/equipment acquisition programs, major modification programs, and applicable research and development projects. The goal of this standard is single, uniform approach by the Military Services for conducting those activities necessary to (a) cause supportability requirements to be an integral part of system requirements and design, (b) define support requirements taht are optimally related to the design and to each other, (c) define the required support during the operational phase, and (d) prepare attendant data products. LSA is the selective application of scientific and engineering efforts undertaken during the acquisition process, as part of the system engineering and design process, to assist in complying with the supportability and other Integrated Logistic Support (ILS) objectives through the use of an iterative process of definition, synthesis, tradeoff, test, and evaluation.

This standard provides general requirements and descriptions of tasks which, when performed in a logical and iterative nature, comprise the LSA process. The tasks are structured for maximum flexibility in their application. In addition to the general requirements and task description sections, this standard contains an application guidance appendix which provides rationale for the selection and tailoring of the tasks to meet program objectives in a cost effective manner. This document is intentionally structured to discourage indiscriminate blanket applications. Tailoring is forced by requiring that specific tasks be selected and that certain essential information relative to implementation of the selected tasks be provided by the requiring authority. Additionally, the user must be aware that when the LSA process, or a portion thereof, is implemented contractually, more than the LSA statement of work and LSA deliverable data requirements must be considered. Readiness and

supportability requirements and objectives must be appropriately integrated and embodied in specifications, general and special contract provisions, evaluation factors for award, instructions to offerors, and other sections of the solicitation document.

Defense system acquisitions are directed toward achieving the best balance between cost, schedule, performance, and supportability. Increasing awareness that supportability factors, such as manpower and personnel skills, are a critical element in system effectiveness has necessitated early support analyses, the establishment of system constraints, design goals, thresholds and criteria in these areas, and the pursuit of design, operational, and support approaches which optimize life cycle costs and the resources required to operate and maintain systems. This standard was prepared to identify these early analysis requirements and foster their cost effective application during system acquisitions.

Individual tasks contained in this standard shall be selected and the selected task descriptions tailored to specific acquisition program characteristics and life cycle phase. Application guidance and rationale for selecting tasks and tailoring task descriptions to fit the needs of a particular program are included in appendix A. This appendix is not contractual and does not establish requirements.

Unless otherwise specified, the following standards and handbooks of the issue listed in that issue of the Department of Defense Index of Specifications and Standards (DoDISS)) specified in the solicitation form a part of this standard to the extent specified herein.

Military Standards

MIL-STD-1366	Material Transportation System Dimensional and Weight Constraints, Definition of.
MIL-STD-1388-2A	Logistic Support Analysis Data Element Definitions.
MIL-STD-1629	Procedures for Performing a Failure Mode, Effects, and Criticality Analysis.

(Copies of specifications, standards, drawings, and publications required by contractors in conjunction with specific procureement functions should be obtained from the procuring activity or as directed by the contracting officer.)

3.1.9 Maintainability Standards Options

This options report identifies Maintainability standards which are applicable to Maintainability Programs of the Space Station Data Systems. This options report is limited to two most widely used Maintainability Standards in Military and/or NASA Space Programs. These two Maintainability Standards are:

MIL-STD-470A, MAINTAINABILITY PROGRAM FOR SYSTEMS AND EQUIPMENT NHB 5300.4(1D-2), SAFETY, RELIABILITY, MAINTAINABILITY AND QUALITY PROVISIONS FOR THE SPACE SHUTTLE PROGRAM.

3.1.9.1 MIL-STD-470A MAINTAINABILITY PROGRAM FOR SYSTEMS AND EQUIPMENT

This standard consists of basic application requirements, specific tailorable Maintainability program tasks, and application, guidance and rationale for task selection. This standard is structured to discourage indiscriminate blanket application. Tailoring is forced by requiring that specific tasks be selected and that certain essential information relative to implementation of the task be provided by the contracting activity. The tasks can be tailored to meet specific program needs for space and terrestrial segments.

This standard provides task descriptions for maintainability programs. The tasks, as tailored, will be applied to systems and equipment development, acquisitions and modifications. Software maintainability is not covered by this standard.

Tasks described in this standard are to be selectively applied in Department of Defense contract-definitized procurements, request for proposals, statements of work, and Government-in-house developments requiring maintainability programs for the development and production of systems and equipment.

Task descriptions are intended to be tailored as required by their users as appropriate to particular systems or equipment program type, magnitude, and funding.

Application guidance and rationale for selecting tasks to fit the needs of a particular maintainability program are included in attached Table 3.1.9-1.

This standard originated in March 1966 and was revised in January 1983.

Government Documents The following documents, of the issue in effect on date of invitation for bids or request for proposal, form a part of this standard.

Standards

Military

MIL-STD-280,	Definitions of Item Levels, Item Exchangeability, Models, and Related Terms
MIL-STD-471,	Maintainability, Verification/Demonstration/ Evaluation
MIL-STD-721,	Definitions of Terms for Reliability and Maintainability
MIL-STD-785,	Reliability Program for Systems and Equipment Development and Production
MIL-STD-1388-1,	Logistics Support Analysis
MIL-STD-1629,	Procedures for Performing a Failure Mode, Effects and Criticality Analysis

Publication None

Military Handbook MIL-HDK-472, Maitainability Prediction

3.1.9.2 NHB 5300.4(1D-2); SAFETY, RELIABILITY, MAINTAINABILITY AND QUALITY PROVISIONS FOR THE SPACE SHUTTLE PROGRAM

This NASA Handbook (NHB) establishes common safety, reliability, maintainability, and quality provisions for the Space Shuttle Program. NASA centers use this publication as a basis for negotiations with Shuttle Program contractors and as the guideline for conduct of program safety, reliability, maintainability and quality activities at the centers.

The maintainability provision in NHB 5300.4(1D-2) sets forth maintainability task requirements. Shuttle Program contractors utilize the maintainability provision as a guideline for program conduct. Furthermore, NASA centers use the maintainability provisions as a basis for negotiating maintainability tasks with Shuttle Program contractors.

Tasks described in the NHB maintainability provision are applied as an integral part of the design and development process. The design and development process includes the evaluation of hardware maintainability through analysis, review and assessment.

Tasks described in the NHB maintainability provision are intended to be tailored as required by, and in accordance with, the applicable Data Requirement Documents (DRD).

Application guidance and rational for selecting tasks that satisfies the needs of a particular maintainability program is included in applicable DRD's and statements of work.

NHB 5300.4(1D-2) was revised to the current NHB 5300.4(1D-2) in October 1979. As such, NHB 5300.4(1D-1) was canceled in October 1979.

NHB 5300.4(1D-2) Maintainability provision incorporates information from NH 5300.4(A), Reliability Program Provision for Aeronautical and Space System Contractors.

3.1.10 Procurement Standards Options

In the past, Government prime contracts were awarded in accordance with the applicable Government Agency regulations in effect at time of prime contract solicitation and award. These individual agency regulations were incorporated as the result of statutes, laws, regulations, Federal acquisition regulation, Agency supplements, etc., developed over the years.

On April 01, 1984, NASA, DOD and other Government Agencies adopted Uniform Federal Acquisition Regulations commonly known as FAR. The FAR, with agency supplemental regulations, replaces Federal Procurement Regulations (FPR), the Defence Acquisition Regulation (DAR) and NASA Procurement Regulation (NASPR). It provides a uniform regulation for use by all Federal Executive Agencies in their acquisition of systems, supplies and services with appropriated funds. The FAR system has been developed in accordance with the requirements of the office of Federal Procurement Policy Act of 1974, as amended by Public Law 96-83. The FAR is issued within applicable laws under the joint authorities of the Administrator of General Services, the Secretary of Defense, and the Administrator for the National Aeronautics and Space Administration, under the broad policy guidelines of the Administrator for Federal Procurement Policy.

The FAR is published in (1) the daily issue of the Federal Registar, (2) cumulated form in the code of Federal Regulations (CFR) and (3) a separate loose-leaf edition. The FAR is issued as Chapter 1 of Title 48, CFR. Subsequent chapters are reserved for agency acquisition regulations that implement or supplement the FAR.

The FAR provides for coordination, simplicity and uniformity in the Federal acquisition process. It includes changes recommended by the Commission on Government Procurement, the Federal Paperwork Commission, various congressional groups and others.

In short, the FAR has been developed to make the Federal procurement process more streamlined, efficient, yet still protect the Government's interests when dealing with private enterprise or the Commercial Sector.

It is highly unlikely any Federal Agency would deviate from the FAR when contracting for services as such action would be contrary to the intent of Public Law and create problems of legality or procurement policy implementation problem for the Federal Agency.

A. Options

There are no viable options when dealing with large Federal Agency Procurement.

Private industry utilizes the Uniform Commercial Code (UCC) as a standard for commercial procurement activity; however, it is totally inappropriate for Government procurements.

All companies have their own "Commercial" terms but these also have broad differences in terms of content and consistency and do not constitute any sort of standard.

3.1.11 Video Standards Options

There are various standards relating to video systems which are potentially candidates for use in SSIS. Since operational scenarios have not been explored in any significant detail, the purpose of the compilation below is to give the salient characteristics of some of the "standard" candidates and advantages and disadvantage for various SSIS applications.

It is worthwhile noting that a video subsystem that might be used to support a docking and berthing operation would require a much higher resolution than a video subsystem used to act as surveillance within a module. Similarly, robotic setups of an experiment (attached or remote from the space station) would probably require a full, real time picture, which it might be acceptable to receive a slow scan (or even frozen frame) picture from that same camera once the experiment is under way.

Another factor in selecting from the candidate video systems is in the control of the video systems. Video which is used to monitor payloads will probably be under control of the mission specialist or personnel at the POCC; video which is used for onboard core/Space Station operations will probably be managed by the responsible crew personnel using a control console/studio like switcher and mixer. These are to be custom devices, although they should accommodate standard electrical interfaced, signal levels, losses, etc.

Finally, the selected candidates should, conceivably, be selected so that various signals can be "meshed" within a channel hierarchy.

Candidate 1. RS170A, NTSC, EIA Standard – This is presently utilized at JSC mission control center, for an in-house 525 live, scan rate, color system. In compressing the three primary color channels into one signal, some video detail is sacrificed, which still yields acceptable entertainment quality.

Advantages

* Video is distributed over single coax. interface.
* All equipment is available off-the-shelf at the present and probably will be in the future.
* Ease of set up and maintainability.
* Video signals can be modulated on RF carriers and distributed for viewing on color receivers.
* Signals would be compatible with the existing NASA ground facilities and television networks, including satellite transmission links.
* Cost effective.

Disadvantages

* Requires video switching equipment to route multi-video sources to multi-users.
* User requires a control module to access the video switch.
* Horizontal resolution is reduced.

Candidate 2. RS170, EIA Standard (monochrome) - Standard broadcast monochrome studio facilities operate at a 525/60 scan rate. The specifications are similar to that of the RS170A system (above) but a subcarrier is not used.

Advantages

* Video is distributed over single coax interface.
* Equipment is available off-the-shelf.
* Ease of setup and maintainability.
* Video signals can be modulated on RF carriers and distributed for viewing on monochrome receivers.
* Signals compatible with existing NASA facilities.
* Equipment is very cost effective.
* Improved horizontal resolution.

Disadvantages

* Black-white only
* Grey level control is difficult.

Candidate 3. TTL Video - Used for certain computer terminal applications or TTL - compatible levels. Equipment is available off-the-shelf. This candidate requires three separate interfaces (red, green, blue composite) or four for a non composite signal plus synch; a subcarrier is not necessary. This type of system is a good candidate for alphanumeric high resolution display.

Candidate 4. RGB Color - This routes red, green , blue over separate cables, which provides display detail better than is available via traditional encoded broadcast color. The green signal channel is usually the composite. Some RGB systems use a non composite signal format an separate color component cables and relay on a separate synch interface.

Advantages

* High resolution displays.
* Color
* Equipment available off-the-shelf.
* Can be compatible with standard NTSC television.
* Utilizes video format.

Disadvantages

* Requires triple video switching equipment to route multi-sources to multi-users.
* Control modules required to access video switch.
* Requires three or four cable interface for routing video signals.
* Not cost effective.
* Setup and maintainability more difficult due to timing of the separate RED, GREEN and BLUE signals.

Candidate 5. RS330A, EIA Standard - This type of system, primarily used for closed circuit TV (CCTV), operates at the 525/60 scan rate. It is used where the signal is generated locally, e.g., on the Space Station, and where picture quality can be controlled, as by the C&T system.

Candidate 6. RS353A, EIA Standard - This system is for monochrome CCTV, used at rates other than 525 line scan. Recommendations are for 675, 729, 875, 945, 1023 lines, all at 60Hz for the field with a 2:1 interface. JSC Mission Control uses the 945 line scan rate.

Advantages

* High resolution.
* Equipment available off-the-shelf for most scan rates.
* Single table interface.
* Ease of setup and maintainability.
* Cost effective.

Disadvantages

* Monochrome displays.
* Not all equipment available as off-the-shelf.
* Not compatible with any other standard.

Option 7. NASA Standard SE-36661

Display Generation Equipment.

DGE converts computer language data into dynamic raster-video displays containing both alphanumeric (A/N) and graphic information. Equipment refreshes continually the last information received or until updated by the computer. DGE output is a digital video signal, 1.4 volt P/P composite for display on 945 line TV monitors.

This equipment is presently utilized in the JSC Mission Control Center for converting shuttle tracking data into usable displays which are available to flight controllers monitoring Shuttle Missions.

Audio Standards

There are a wide variety of Audio Standards that will pertain throughout the SSDS, particularly on Ground Segment communication. These standards are described in the following table (3.1.11-1).

A typical standard Audio (Voice) Analog interface is a 3002 (FCC tariff No. 260) and is as follows:

Table 3.1.11-1
Telephone Conditioning Parameters

	Non-Conditioned 3002 Channel	With C1 Conditioning	With C2 Conditioning	With C4 Conditioning
Frequency Range In Hertz (Hz)	300-3000	300-3000	300-3000	300-3000
Attentuation Distorting (Net Loss at 1000 Hz)	Freq. Range: 300-3000; Deci. Vari.: -3 to +12 Freq. Range: 500-2500; Deci. Vari.: -2 to +8	Freq. Range: 300-2700; Deci. Vari.: -2 to +6 Freq. Range: 1000-2400; Deci. Vari.: -1 to +3 Freq. Range: 300-3000; Deci. Vari.: -3 to +12	Freq. Range: 300-3000; Deci. Vari.: -2 to +6 Freq. Range: 500-2800; Deci. Vari.: -1 to +3	Freq. Range: 300-3200; Deci. Vari.: -2 to +6 Freq. Range: 500-3000; Deci. Vari.: -2 to +3
Delay Distortion in Microseconds	Less than 1750 s from 800 to 2600 Hz	Less than 1000 s from 1000 to 2400 Hz. Less than 1750 s from 800 to 2600 Hz.	Less than 500 s from 100 to 2600 Hz. Less than 1500 s from 600 to 2600s Hz. Less than 3000 s from 500 to 2800 Hz	Less than 300s from 1000 to 2600 Hz. Less than 500 from 800 to 2800 Hz. Less than 1500 s from 600 to 3000 s from 500 to 3000 Hz.
Signal to Noise (dB)	24	24	24	24
Non-Linear Distortion Signal to 2nd Harmonic (dB)	25	25	25	25
Signal to 3rd Harmonic (dB)	30	30	30	30

3.1.12 Data Interfaces Standards Options

Specifications or Standards which apply to Data interfaces are as follows:

Federal Standards

Fed-Std-1003	Telecommunications Synchronous Bit Oriented Data Link Control Procedures
Fed-Std-1020	Telecommunications Electrical Characteristics of Balanced Voltage Digital Interface Circuits
Fed-Std-1030	Telecommmunications Electrical Characteristics of Unbalanced Voltage Digital Interface Circuit

EIA Standards

RS-334
RS-363
RS-366
RS-234
RS-422
RS-423

Mil Standards

Mil-Std-188-100	Common Long Haul and Tactical Communications System Technical Standards
DOD-C-85045	Cables, Fiber Optics, General Specifications
Mil-Std-188-144	Electrical Characteristics of Digital Interface Circuits

International Standards

15-1745-1975	Basic Mode Control Procedures for Data Communications

CCITT Standards

V. 2	Power Links over Telephone Lines
V. 4	General Structure of Signals for Data Transmission over Public Telephone Network.
V.10 (X.26)	Electrical characteristics for unbalanced double-current interchange circuit general use with integrated circuit equipment in the field of data communications (and provisional amendments, May 1977).

V.11 (X.27)	Electrical characteristics for balanced double-current interchange circuits general use with integrated circuit equipment in the field of data communications (and provisional amendments, May, 1977).
V.15	Use of acoustic coupling for data transmission.
V.19	Modems for parallel data transmission using telephone signaling frequency.
V.20	Parallel data transmission modems standardized for universal use in the switched telephone network.
V.21	200-bit/s modem standardized for use in the general switched telephone.
V.22	Standardization of data signaling rates for synchronous data transmission general switched telephone network.
V.22bis	Standardization of data signaling rates for synchronous data transmission leased telephone-type circuits.
V.23	600/1.2K bit/s modem standardized for use in the general switched telephone network.
V.24	List of definitions for interchange circuits between data terminal equipment data circuit terminating equipment (and provisional amendments).
V.25	Automatic calling and/or answering equipment on the general switched telephone network including disabling of echo suppressors on manually established.
V.26	2.4K/1.2K bit/s modem standardized for use on four-wire leased circuits.
V. 26bis	2.4/1.2K bit/s modem standardized for use in the general switched telephone network.
V. 27	4.8 Kbit/s modem standardized for use on leased circuits.
V. 27bis	4.8 Kbit/s modem with automatic equalizer standardized for use ono leased equipment.
V. 27ter	4.8K/2.4K bit/s modem standardized for use in the general switched telephone network.

V. 28	Electrical characteristics for unbalanced double-current inter-change circuits.
V. 29	9.6 Kbit/s modem for use on leased circuits.
V. 31	Electrical characteristics for single-current interchange circuits controlled contact closure.
V. 35	Data transmission at 48 Kbit/s using 60-to-108 KHz group bit/s circuits.
V. 36	Modems for synchronous data trans-mission using 60-to-108 KHz group circuits.

3.2 SYSTEM MANAGEMENT

The purpose of this paper is to describe options for management and interfacility coordination of the end-to-end Space Station Data System (SSDS), which includes the following:

- Processing nodes, distributed among Space Station program elements (SSPE's), space and ground
- Local Area Networks in space
- Uplink/downlink TDRSS links
- Wide Area Communication Network
- Regional Data Centers (RDCs), Data Handling Centers (DHCs), Payload Operations Control Centers (POCCs), etc.
- Local Area Networks on the ground

Each of the elements, facilities, and the links between them must be managed. Primary management functions are:

- Network Control (including Scheduling)
- Network Monitoring
- Network Administration and Configuration Management
- Network Maintenance (including Emergency Management)
- Customer/SSDS Interface

Options for the above are presented in this paper. Most of the options describe the types of decisions which need to be made, since definitive option definition will depend on specific SSDS subsystem designs and ongoing NASA policy decisions.

3.2.1 Network Control

Network Control options relate to which components of the SSDS will be controlled and in what manner. This control includes scheduling and prioritized access to SSDS resources and services.

Control coordination will also be needed for the TDRSS and DOMSAT links/antennae, the existing White Sands NASA Ground Terminal, the proposed New TDRSS Ground Terminal, and the NASCOM TDRSS network, etc.

Primary options relate to the enhanced control functions to be implemented within the existing Network Control Center (NCC) versus elsewhere within the SSDS. The extent of this enhancement of functionality within the existing NCC versus more distributed responsibility for NCC-type control is an important NASA programmatic option. Additional analysis is needed to determine which network control functions can be acceptably distributed, in conjunction with system responsiveness, cost/benefit, security, etc. tradeoffs.

Another option is to develop a new NCC instead of extensively upgrading the existing facility. As increasing network control function can be feasibly automated, the NCC can focus more on network management, fault detection, etc.

Another important programmatic option is the extent of distribution of command and control responsibility. Restricted commands, for example, may need to pass through a central control and verification node, whereas many individual nodes may have authority to validate and send unrestricted commands to the different SSPE's.

'Restricted' commands, and commands affecting flight SSPE's will require more command and control verification and checking than unrestricted commands or commands for ground elements.

3.2.1.1 Real Time Control

Options include the extent of real-time control to be provided, for which functions, and for which SSPE's. Their implementation may also vary in their degree of centralization, their space/ground distribution, etc.

Assessments are needed as to the degree of function centralization versus the ability to perform them in real-time. Functions specific to a particular SSPE and autonomously implemented there may be more easily handled in real or near real-time. Functions which require coordination among SSPE's or other resources, however, may require some centralized control and may only be able to be performed in near real-time.

In emergency situations, for example, it may be necessary to re-direct traffic, reallocate resources, or even shut down some users. These are necessary SSDS control functions. Critical control functions therefore must have alternative or redundant implementations in the event of failure or the primary control function path. Redundant functions may often have somewhat lesser capabilities than their primary implementation, since they will be invoked only in rare or emergency situations. This must not occur, however, for critical control functions with real-time requirements. Critical real-time control functions must also be fully supported in redundancy mode.

3.2.1.2 Intermediate Time Controls

Many network control functions may not be real-time critical, such as equipment reconfiguration, long term resource scheduling, etc. In these instances there may be a wide tolerance in acceptable response times. It is therefore anticipated that the redundant implementation of these functions will satisfactorily be able to meet their required response.

3.2.1.3 Communications Link Scheduling/Prioritization

Optional prioritization schemes for communication link utilization and conflict resolution are possible. Emergencies and mission critical situations presumably will have first priority, and link allocation must be expedited in these situations.

In addition, a system of dynamically allocated priorities, in which link utilization is automatically and dynamically scheduled according to current traffic loads and priorities, is desirable. The use of automated software capabilities for automated scheduling is an attractive option if technologically mature by IOC. Cost and reliability criteria need to be evaluated. Increased automation should reduce the required ground support, however, and may facilitate function decentralization.

The required security and availability of the SSDS communication traffic and links must be maintained. Prioritized usage of alternative links when primary links are not available needs coordination and control since capacities of alternate links may be less than that of primary links. Verified update procedures are important to maintain system integrity as well as system transparency to updates. The breadth of checking of updates required is an important factor.

The everyday scheduling and control of network links will depend on details of the network design and decisions regarding prioritization of service. The selection of options will depend on SSDS design features and on NASA policy decisions related to network allocation priorities and function implementation alternatives.

3.2.1.4 Long Term Planning & Scheduling

Long term planning and scheduling of missions, resource utilization, and data management are important areas of system management. Options for these areas are discussed throughout the rest of this paper; in general little time-criticality is associated with long range planning functions.

3.2.2 Network Monitoring

The extent of network monitoring and performance assessment is another programmatic option. Each node can be assigned different responsibilities for monitoring and assessing its performance. This distribution in the network monitoring and performance assessment function will need to be counterbalanced with a need for centralized responsibility for overall network monitoring and assessment. The decision to maintain responsibility for overall network performance, analysis, and statistics availability with the Network Control Center (NCC) will need to be reassessed in terms of the breadth of location and functionality of the SSPE's and the enhanced capabilities within each SSPE.

3.2.2.1 Network Monitoring Data Management

This section describes the options for the collections of system management data from the network, to be used for the following example purposes:

- to support network control (from real-time controls to long term network planning and scheduling)
- to support trouble shooting and repair
- to support network usage and billing
- to assess ongoing network performance and ongoing resource utilization

Options related to many of these issues are presented in the 'White Paper' on 'Network Monitoring and Performance Assessment', covering options category 2.6. Additional options are discussed below.

3.2.2.1.1 Data Collection

High level decisions are:

- o What types of data are to be collected for each facility, subsystem, and component?

Data types will range from resource utilization and performance statistics to ongoing monitoring and maintenance data. These data can be reported for each SSPE, for each SSPE subsystem, and/or for each subsystem component (individual processors, etc.).

- o How will the data be transported to the location where analysis is to be performed?

How much of the data will require real-time or near real-time electronic transmission versus periodic offline delivery? Some data will require transport to the central monitoring node, i.e., possible security breach information, etc. Ongoing performance statistics (not resulting in a fault detection), on the other hand, may only need to be reported weekly or monthly.

Extensive data collection could impact the performance of the data system itself. The extent of data collection and monitoring needs to be evaluated with respect to the overall performance requirements and capabilities of the SSDS.

The extent of error detection and correction performed at each node is another option. The techniques for determining the bit error rates and how these differ by data type are discussed in the 'Network Monitoring and Performance Assessment' white paper.

It is also important that the types of data collected should have a direct relationship to the controls implement, i.e., one should have a real time, short term, or a long term response based on the evaluated results of the measurement data. This may not always be possible. Where it is possible, it suggests the groundwork for future automation of the data system.

3.2.2.1.2 Data Reporting

There are options in the periodicity, scope, and breadth of dissemination of management and network performance information. Certain types of information will require more regularity, greater detail, or wider distribution than other types. Although extensive reporting might affect SSDS performance, reporting is generally expected to represent a small overhead compared to its contribution to efficient system management.

3.2.2.2 Data Analysis

The techniques to analyze the data should be determined prior to its collection. The types of reports to be made and to whom, the anticipated

results of analyses, and the types of corrective measures indicated need to be decided.

3.2.2.3 Cost Accounting

The breadth of accounting performed is another option. The breadth and depth of resources monitored will determine the amount of data required.

An important area is the customer accessibility to up-to-date cost information for services he desires or has used. Standard services such as CPU utilization, data base management and data storage will normally be monitored but what will be the accounting procedures to accommodate specialized services, or do we presume the SSDS will know all possible service requests before each mission?

Important criteria in deciding the limits of information tracking and reporting are cost versus responsiveness. Highest SSDS responsiveness and data accountability are desired on the part of the user community. It is nevertheless impractical to track and report on each quantum or packet of data and/or to track the performance of each component. Development and operations costs and likely performance degradation will probably be the primary criteria in limiting the extent of information tracking and accounting.

A broader tradeoff is the relative funding priority of the entire function of data administration and accounting versus choices for better (and more expensive) technology, increased function redundancy, increased mission support, etc.

3.2.3 Network Administration

Typical network administration functions are:

- o Configuration Control

This includes functions such as resource allocation tracking, system access authorization control, resource prioritization algorithms, command management tables, etc.

- o Integration and Initializlation of Hardware/Software

This includes system startup configuration, tracking of the integration and implementation of new capabilities, etc.

- o Hardware/Software Updates Management

This includes management and control of system updates in hardware and software, documentation control and requirements, maintenance of test procedures for updates, etc.

Network administration involves integrating and initializing software, making updates, tracking hardware and software version numbers, etc. Two important subfunctions involved in administration are:

- o maintenance and update of databases which record the current data system configuration, for each facility, subsystem, and component

- o administration of the software which provides the actual data system functions—modifying routing tables, etc. In addition, checking must be performed to insure that parameters, software versions, etc., are correct, up-to-date, and compatible with other facilities, subsystems, and elements.

3.2.3.1 Centralized Versus Distributed Options

Configuration databases may vary from totally centralized to totally distributed. A centralized database is easier to control and update; it may be difficult to access by remote sites, however. A fully distributed database is more accessible to each facility, but if there are overlapping needs for the same database information, it may be next to impossible to keep their information synchronized. Distributed implementation may also incur increased costs due to replication of hardware and software.

Similarly, the degree of centralization of network administration can vary. A centralized administration has the advantage of locating all the relevant expertise and information in one or few places — reducing chances for error and the required number of staff. Communication costs may increase, however, if the central administration staff need to access any node in the network. Increased distribution in network administration will generally result in a larger staff and greater costs, but more autonomy and self-direction.

3.2.3.2 Configuration Control Options

Configuration control is the tracking of hardware and software modifications or updates and administering an orderly management scheme for these functions.

The control of SSDS updates, both onboard and on the ground, is important. Options include:

1. The extent of simulation and/or checking required before authorization is granted for insertion or replacement of hardware or software.

2. The amount of concurrency in running the updates simultaneously with the existing version until adequacy of the update is assured.

3. The documentation and update control procedures, including the extent of corroboration of NASA personnel and customers before update insertions are approved.

The options chosen will depend on the exact functions being updated, their criticality, the cost of extensive checking, etc.

Optional configuration management procedures can be implemented.

For critical, potentially life-threatening functions, a centralized, highly controlled configuration monitoring may be necessary. Distributed update and function migration responsibilities may be possible, however, in the areas of payload reconfiguration, non-critical core subsystem functions, etc.

The management of function migration from ground to space (which will be an ongoing process in the Space Station program) is an important area, discussed below.

3.2.3.3 Space/Ground Functionality Transfer Procedures

In addition to regular updates due to error detection and correction, enhanced capability insertion, etc., a Space Station program goal is to evolve to an ever more autonomous Space Station base, with fewer and fewer functions required on the ground (with associated large ground support staff).

Updated functionality, implemented both in hardware and software, will need to migrate from ground to space as the functionality achieves a sufficient degree of compactness, reliability, and verification. A broader spectrum of testing and organizational approval may be required, however, before deciding to migrate a ground-based function onboard. Efficient and thorough configuration management of ground-to-space upgrades will require precise record keeping and adequate check out procedures.

3.2.3.4 Network Administration/TMIS Interface

The TMIS network will be primarily responsible for program management, configuration control, and data communications in the areas of documentation and systems engineering support.

The overlap of functions and required data between the TMIS and the SSDS needs to be assessed as the TMIS definition becomes mature. Options regarding the procedures and controls for information exchange between the two systems need to be developed.

The breadth of data interchange and replication and the privacy of each data type will be key options. The degree of compatibility between TMIS security and SSDS security in areas of management data will require coordination.

3.2.4 Network Maintenance

In addition to updates management, maintenance of hardware and software and the repair of detected malfunctions are an important SSDS function. An expanded discussion of software maintenance options related to the SSDS is presented in the options white paper on 'System Development', Section 3.5. Options for hardware maintenance and selected management issues are presented here.

3.2.4.1 Redundant Function Implementation Options

In coordination with maintenance and repair procedures, the extent of redundancy in certain functions, and the amount of space/ground replication need to be decided. Function replication to provide redundancy may be preferred over system repair or replacement options. Factors affecting these choices are cost, weight and space (onboard considerations), and/or function criticality. For critical functions, replication may be required in order to minimize possible downtime of these functional capabilities.

3.2.4.2 Trouble Shooting & Repair

3.2.4.2.1 Emergency Management Options

The extent of backup recovery options upon subsystem or component failure will depend upon mission criticality and/or crew and Station safety concerns.

The extent of crew training and responsibility during emergency recoveries versus the amount of online ground support is an important option area for each emergency type.

An assessment needs to be made of the relative likelihoods of various types of subsystem failure before an intelligent set of emergency procedures options can be formulated.

3.2.4.2.2 Separate Vs Duplicate Maintenance Functions

Another option is whether maintenance functions are implemented within the same data system hardware and software as that which provide the data system functions, or whether separate hardware and software are used.

Use of the same hardware and software reduces costs. However, failures of key components might hinder fault detection, maintenance, or repair. For example, if the same communication path is used for maintenance as for normal operation, there may be no alternate way to reach the failed link or node for testing and fault isolation. The provision of separate access ports and separate diagnostic software may insure reliable access to processing nodes.

3.2.4.3 Hardware Maintenance Options

The primary options here relate to the extent and periodicity of ongoing hardware maintenance checking and subsequent repair procedures when problems occur.

Maintenance checking breadth and periodicity relate to the extent of hardware devices monitored and the frequency with which this monitoring is performed. Critical functions will require more thorough monitoring and usually more frequent monitoring, depending on the time-criticality of associated repairs.

The repair procedures once problems occur have options of switchover to replica components or to alternative function/service implementation, as discussed in Sections 3.2.4.1, 3.2.4.2, and 3.2.5.5. For certain non-critical functions, it might be acceptable to have no backup redundancy but to repair the problem online as it occurs (if little risk occurs for delayed repairs due to unforeseen problems) performed.

An important aspect of hardware maintenance is the ability to quickly and accurately isolate problems. Normal diagnostics packages, often provided by vendors, may be adequate to detect the large majority of problems which may occur. Some unusual problems may not be detected, however, and it can be vital to isolate such faults. One procedure, which may be useful especially for critical functions, is to have interface diagnostics available between each hardware component or subsystem. The interface diagnostics would provide the capability to send a spectrum of inputs to the device and monitor the expected outputs. Assuming a sufficiently broad spectrum of inputs to completely test the functionality of the device, this procdure may help to isolate faults which vendor diagnostics packages may not detect.

The implementation of such interface diagnostics will occur during the design and assembly of the hardware systems. A decision to incorporate such procedures needs to be made during the design process to assure efficient and thorough implementation of this capability.

The other key aspect of hardware maintenance is the set of procedures to repair detected faults, subsequent to redundant function or component implementation. For onboard components, a key issue will be whether to have the crew repair particular components versus waiting for replenishment of the components on the next scheduled Shuttle rendezvous. For critical components which fail, decisions will need to be made as to whether single redundancy is sufficient since the intervals between Shuttle dockings may be many weeks apart. Will triple redundancy for some components be required, and will this even be adequate for some items, especially ones for which the crew will be incapable of repairing.

Key tradeoff factors in these decisions will be the criticality of the hardware component, the cost, weight, and space requirements of the spare components, and the likely ease of repair by the crew.

3.2.4.4 S/W Maintenance Options

See options paper on 'System Development", options section 3.5.

3.2.5 Customer/SSDS Interface Options

Options for interfaces between the SSDS and the customer are presented here. The options related to interface with the SSDS Simulation Center (SC), mission integration, the Software Support Environment (SSE), onboard training, and alternative service restoration (in case of failure) are discussed.

3.2.5.1 Simulation Center

Basic options relate to the location, responsiveness, and breadth of Simulation Center (SC) services to support customer tests of operational procedures; software, payload, hardware, and core services interfaces; and performance estimation and modeling. For each of these functions, the extent of services provided, the user-friendliness, and cost are key tradeoff criteria.

The responsiveness, diversity of locations, and breadth of the SC services are SC cost drivers — rapid response requires potent CPU capabilities; wide location diversity increases replication costs for hardware and software; breadth of services implies extensive software development.

3.2.5.2 Mission Integration Planning

Key options are which procedures and system services will be provided to assist in the integration of new missions. Examples are 1) the provision of simulation capabilities to check-out instrumentation prior to onboard integration and 2) a customer interface organization to handle management and negotiation of new mission requirements and goals. This latter area is discussed in the 'Wide Area Communications' white paper, Section 2.5.2.2.10. The extent to which new missions will need to be validated (at the Simulation

Center, etc.) before flight approval may involve a tradeoff between SSDS user-friendliness and the need for assured and adequate system operations. The development of SSDS software and operations support hardware of sufficient responsiveness and thoroughness may be an important option. Simplified checkout procedures and minimal checkout times are desired, however.

Information on the availability and associated costs of using system resources will be valuable. Other important factors include the capability for extensive checkout of equipment and software by the customer at his home facility and the degree and ease of use of standardized interfaces and procedures.

3.2.5.3 SSE Interface

An interface is needed for the transfer of customer developed (at the SSE) software to the SC. Both the SSE and the SC functions may be distributed to a number of interfaces may be needed. Functionally, a part of the SC may be within the SSC, i.e., the individual payload models developed by the customer. Each individual link is presumably a minor system design constraint since only utility or user software needs to be transferred between the two. The performance requirements of each link are small compared to the rest of the SSDS. The logical connectivity is impacted by the extent of function distribution.

3.2.5.4 Onboard Training

The extent of onboard training supported by the SSDS versus customer-provided is another option. Onboard training will normally be more expensive than ground-based simulation; sometimes, however, the SC development and execution costs may exceed the onboard performance testing costs. Another important cost factor in onboard training is the loss of usage of the Space Station resources and crew during the training period.

Live video onboard training support represents a desirable option from a user-friendliness standpoint, but expensive and resource intensive with respect to communications uplink and downlink.

3.2.5.5 Service Restoration versus Service Repair

An important distinction is between the operation of the data system and the customers perception of the data system. One can restore service before one repairs the failure.

For example, the customer may have very high reliability requirements, resulting in a very short time allowed to repair failures. What the customer is interested in, however, is that service be restored within a short period — and not necessarily that a particular component be replaced. Thus, in the event of a failure, an alternative to finding and repairing the specific failed component, is to locate the problem down to a subsystem, port, or equipment chain, and replace or switch in a whole new subsystem or equipment chain. Service is thus restored quickly. The actual locating and repair of the failed component or subsystem may not occur for some time afterwards. The system's service availability appears uninterrupted to the customer. A high degree of redundancy may be required to achieve this, however.

In summary, system management is a critical function in order that the SSDS provide adequate services and meet customer needs. Many of the subsequent tradeoff decisions will depend on specific subsystem design options and NASA programmatic and policy decisions.

3.3 DELETED

3.4 DELETED

3.5 SYSTEM DEVELOPMENT

3.5.1 Hardware Procurement

Hardware Procurement is that activity associated with the selection of hardware products to meet program requirements and the selection of suppliers for those products. Procurement involves not only the selection and associated strategies but also the definition and administration of those tasks required to insure product suitability, i.e. acceptance and qualification testing.

Discussion of significant procurement issues from the SSP perspective are provided in the following.

3.5.1.1 Hardware Commonality

Hardware commonality refers to utilization of specific configuration (or configuration family) hardware in differing applications. The advantages of commonality include operational gains of reduced spares and support requirements, narrower expertise requirements, and perhaps reduced procurement costs. The disadvantages include the inevitable compromises required in providing blanket solutions for variant requirements and the enhanced program risk that reliabilities of the selected configuration(s) may be less than projected and may impact system availability.

This subject is explored at length in the Standardization/Commonality Options White Paper, Item 4.3.1, and will not be further discussed here.

3.5.1.2 Qualification Levels

The Space Station represents a departure from prior space projects since the primary elements will be assembled/activated on-orbit, thus the operational environment of the DMS hardware will be relatively benign, particularly that hardware in the Station modules. All of the SSPE space hardware, however, will be subjected to thermal, mechanical and pressure dynamics during the NSTS boost to orbit and during subsequent assembly/activation operations. In addition, the near earth radiation environment and its effects must be addressed. Actual operating environments will differ significantly based on hardware application (Space Station, COP, POP) plus their resultant orbital parameters (altitude, inclination, sun angle, etc.). A rigorous acceptance/qualification program must therefore be performed on all flight hardware to insure its mission suitability.

Systems Engineering will analyze potential equipment environmental exposures during launch, deployment, and operational life to establish appropriate acceptance/qualification test levels. Operational failure modes must also be considered in this analysis to define potential worst case environments. The environments to be considered, as indicated in the reference (1) NASA Technical Memorandum, are thermal, pressure, shock/vibration, radiation, and EMI.

The Space Station and COP will be boosted into a nominal 500Km, 28.5^{o} inclination orbit; the POP orbit will utilize a polar (98.25^{o}) orbit with an nominal altitude of 705Km. The natural environments for the assembly and operational phases of these three elements are provided in Table 3.5.1 - 1; note that the POP will be assembled and activated at the Space Station then transferred to its operational orbit.

Table 3.5.1 - 1

Space Station, COP and POP Natural Environments

		Station & COP		POP	
		Assembly	Operation	Assembly	Operation
Thermal	Source:	$400 BTU/ft^2-hr$	———	———	———
	Sink:	0^oK	———	———	———
Pressure		10^{-9} torr	———	———	0^{-10} torr
Vibration		(Boost to orbit environment - see note 2)			
Radiation		.63rad(Si)/day	———	———	5-70rad(Si)/day

Note 1. Atm pressure will be dynamic depending on solar activity

Note 2. NSTS boost to orbit vib'n environment estimated to be 10 -12g rms, 10 - 2000Hz.

Note 3. Daily radiation based on estimates of solar min. and solar max. dose for nominally shielded equipment.

There is an inherent dilemma for the procurement process in that the SSDS hardware definition will specify state of the art designs and technologies which are not generally available today in space/flight qualified (or qualifiable) configurations. Procurement options for this case are to 1) rework/redesign hardware to meet projected environments, or 2) drive the SSPE designs/operational planning to accommodate less rugged/less radiation tolerant hardware. The rework activity of option 1) could be significant since it will include comprehensive review and appropriate corrective rework of:

a) vendor processes and materials for acceptibility
b) thermal management techniques and materials
c) component/circuit card natural frequencies and deflections
d) passivation techniques, including zero-g effects
e) enclosure applicability

This rework approach must be carefully evaluated with the supplying vendor to assure that the net gains can be achieved.

This section briefly discusses each environment and the significant procurement considerations, then proposes options addressing those procurement issues.

3.5.1.2.1 Thermal Environment

3.5.1.2.1.1 Description

The space environment provides a near earth solar radiation source of approximately 440 BTU/ft^2-hr and a space sink of approximately 0^{o}R. Thus extreme case temperature variations can be achieved by unprotected equipment depending on orbit beta angle, equipment orientation and power dissipation. Mature thermal management techniques utilizing wrapping materials, heaters, active cold plates are available, however, to maintain equipment case temperatures within acceptable bands during stable operation. Also, although station and platform build-up scenarios can include transient conditions such that non-operating equipment may be subjected to significantly wider temperature ranges, experience indicates that special handling equipment and techniques can be employed to accommodate less thermally rugged units. This thermal manageability provides a wide lattitude of hardward options with corresponding potentials for cost savings even if special cannisters must be provided. In summary, thermal constraints are not considered to be a significant driver for the DMS hardware therefore no distinct procurement options are identified.

3.5.1.2.2 Pressure Environment

3.5.1.2.2.1 Description

At 400Km, the atmospheric pressure is in the range of 10^{-9} torr, which is a virtual vacuum for the SSDS equipment. This environment represents a potential design/procurement problem for platform and truss hardware since most off-the-shelf equipment is operated with a nominal internal (air) pressure of 15 psia although actual internal pressure requirements may be much

less for adequate convective cooling. There appear to be three options for consideration to overcome this problem: 1) provide equipment with sealed/pressurized enclosures, 2) provide a pressurized environment for the equipment and 3) redesign the equipment to operate with unpressurized.

A qualitative assessment of these options to generic criteria is provided in Table 3.5.1 – 2.

Table 3.5.1 – 2

PRESSURE ENVIRONMENT

Option Impacts on Evaluation Parameters

	OPTION #1	OPTION #2	OPTION #3
RISK PARAMETERS	PROVIDE PRESS'D ENCLOSURES	PROVIDE PRESS'D ENVIRONMENT	REDESIGN FOR VACUUM
COST	MODERATE	LOW	MODERATE-HIGH
SCHEDULE	MODERATE	LOW	MODERATE
PERFORMANCE	LOW	LOW	MODERATE
RELIABILITY	LOW	LOW	MODERATE
MAINTAINABILITY	LOW	LOW	LOW
SAFETY	LOW	LOW	LOW

NOTE: Costs of Qual. Program not considered in evaluation.

3.5.1.2.2.2 Options Characterization

a. Seal/Pressurize Enclosures

This option reworks/replaces the equipment enclosure to provide a pressure seal. The internal pressure should be some fraction of 15 psi in order to limit the strength required by the enclosure, while retaining adequate convective cooling. Displays/controls mounts must also be sealed. Qualification testing will demonstrate the suitability of the final configuration. The advantage of this approach is that it provides a potential "quick fix" with minimal disturbance of existing design/performance; the disadvantage is the inevitable leakage of seals, relief valves, etc., that must be resolved by constant purging, or periodic re-pressurization.

b. Provide Pressurized Environment

Operationally pressure sensitive equipment utilized on the Station structure or on the COP or POP could be enclosed in a pressurized shroud to eliminate the problem of application in compatibility. The advantage of this option would be the reduced impact on the individual hardware; the disadvantage would be the potential finite leak rate that must be addressed as in the previous option.

This approach appears to be fairly straight forward and could provide a common solution for spatially local hardware sets. Thermal management of the enclosed equipment would not present any significant problems; however, a shroud with sufficient strength to contain a few psid must be designed to prevent weight concerns.

c. Redesign Equipment

This option fully redesigns the selected equipment to operate in a vacuum, utilizing conduction paths to its enclosure/cold-plate for thermal management. The advantage of this approach would be the cleaner solution of providing an environment tolerant design. The disadvantage would be the significant cost of redesign, and the potential impacts on unit performance.

3.5.1.2.3 Mechanical Environment

3.5.1.2.3.1 Description

The mechanical environments for the SSPE equipment consist of the vibration, shock and acoustic environments of NSTS launch and staging, handling operations during acquisition and maintenance, and potentially, the dynamic affects of Orbiter docking/berthing. It is anticipated that the orbital boost operation will provide the most severe environment. NSTS lift-off vibration and acoustic levels are well defined and are provided in Tables 3.5.1 - 3 and 3.5.1 - 4 attached. Typically, the dominant low frequency vibration environment is mechanically transmitted to the equipment through the orbiter longerons while the high frequency vibration environment is acoustically induced. The vibration response is a function of the equipment configuration and method of mounting. The response of the SSDS equipment to package inputs from all sources will be derived from tests and/or analysis by Systems

Engineering. Based on MDAC MSIS experience with the PAM program, the composite flight levels are not expected to exceed 10-12 grms across a defined 10 - 2000 Hz profile for trunnion mounted packages. These levels are expected to envelope all vibration, shock and acoustic exposures and are not considered to be severe for "ruggedized" hardware.

As indicated earlier, the equipment must also undergo some level of qualification however, the levels and durations will be reduced from criteria defined in the standards (i.e., DOD-E-8983E) in order to accommodate protoflighting.

Again, the probability of qualifying available commerical or GFE hardware may be low; the available the options are:

1) rework existing equipment to beef up enclosures and component mounting while providing appropriate isolators at launch package mounts.

2) completely redesign equipment to meet projected profiles.

Table 3.5.1 - 5 provides a qualitative assessment of these options.

Table 3.5.1 - 5

MECHANICAL ENVIRONMENT

Option Impacts On Evaluation Parameters

	OPTION #1	OPTION #2
RISK PARAMETERS	REWORK OTS EQUIPMENT	REDESIGN OTS EQUIPMENT
COST	MODERATE	MODERATE - HIGH
SCHEDULE	MODERATE	MODERATE - HIGH
PERFORMANCE	LOW	LOW- MODERATE
RELIABILITY	LOW - MODERATE	LOW
MAINTAINABILITY	LOW - MODERATE	LOW
SAFETY	LOW	LOW

NOTE : Costs of Qualification Program not considered in evaluation.

3.5.1.2.3.2 Options Characterization

a. Rework Off-The-Shelf Equipment

This option addresses the rework of existing equipment that has all the required attributes (configuration, performance, etc.) except it is not sufficiently rugged to survive the launch and deployment environment. The concept is to ruggedize enclosures and internal components. This effort includes:

a) analysis of circuit card natural frequencies providing stiffening as required

b) analyzing component mounts and adding additional strength (staking) and conformal coating as required

c) strengthening the enclosure with increased wall thicknesses and gusseting as required.

Since this is a compromise approach, vibration/shock isolators may also be required at the equipment mount within its launch package. The advantage of the approach is the potentially low cost of the rework; the disadvantage is the resulting risks in qualifying the resulting configuration and the potential impacts on performance and reliability.

b. Redesign Off-the-Shelf Equipment

This option comprehensively redesigns the equipment to MIL-Spec criteria such that qualification testing to uncompromised levels can be confidently performed. The advantage of this approach is the achievement of a cohesive, well engineered product; the disadvantage is the high cost of the redesign/redevelopment.

3.5.1.2.4 Radiation

3.5.1.2.4.1 Description

High energy charged particles that proliferate the terrestrial space region with their effects on semiconductor technologies, present perhaps the most severe environment in terms of long term equipment compatibility. These particle fluences include high energy protons and heavy nucleii from galactic cosmic rays, protons, electrons and alpha particles produced by solar flare activities, and trapped protons and electrons of the Van Allen radiation belts. The galactic particles have the highest energy distributions, up to 10^{10} eV, however all three sources have sufficient energy distributions such that their particles can penetrate through normal spacecraft skin and hardware enclosures to deposit energy within the semiconductor material. The resulting effects on electronics are:

1) the 'single event upset' (SEU) phenomenon where-in a logic state (single bit) change occurs in a logic latching device, or semiconductor memory, caused by a high energy particle hit at a critical circuit point. and, 2) a gradual degradation of device characteristics proportional to the total dose energy (radiation) accumulated,

The predominant cause of single event upset is heavy nuclei and high energy protons that deposit their energy in the form of ionization in the IC material. If the ionization occurs near a depletion region then the holes and electrons may be collected to produce a charge increase of sufficient magnitude to cause the circuit to change state. In most cases the phenomenon is transient and the circuit will operate normally thereafter, however, in some cases, parasitic 4-layer paths can be stimulated to 'latch-up' resulting in IC burnout. In most cases the fault tolerance design features, i.e. error correction codes, parity checks, voting mechanisms, and redundancies will maintain SSPE operation, but hardware is becoming more sensitive to the problem as LSI densities increase and feature size decreases. The SEU phenomenon must be comprehensively addressed in hardware and software designs to minimize potential burdens on fault tolerance mechanisms and on-orbit servicing.

Electron and proton fluences constitute the primary total dose radiation problem for spacecraft. The accumulation of this radiation is proportional to the particle fluxes encountered and is somewhat statistical, based on orbital parameters and solar flare activity. Its unit of measure is the 'rad', defined as 100 ergs/g and it would not be unusual for the total dose of a polar orbiting spacecraft with minimal natural and intentional shielding to reach 25K-50K rads(Si) during a solar maximum year. This dose would significantly degrade transistor gains, increase junction leakages and shift thresh-hold voltages in MOS transistors of typical hardware. The over-all result would be a general decrease in performance on digital circuits and a decrease in accuracy of analog circuits such that mission requirements could not be achieved.

Optical fibers are also degraded by radiation to varying degrees. The affects can be a drastic increase in attenuation (dB/km) depending on the fiber type.

High neutron fluences and dose rates associated with nuclear detonation are not considered as part of the mission environment and are not addressed. The availability of semiconductor technologies with a radiation tolerance of 1K rads(Si) and above is extremely limited as shown in the Figure 3.5.1 - 1 test data. Fortunately for the Space Station and Co-orbiting platform, the natural magnetic field of the earth provides an effective shield for the galactic and solar flare particles in the lower inclination orbits up to approximately 40^{o}, and altitudes below 500Km, such that with normal enclosure material, negligible radiation will be accumulated from these sources. These orbits can still provide measurable radiation, however, from passes through the 'South Atlantic Anomaly', a region centered at 35^{o} south latitude and 35^{o} west longitude where the protons and electrons from the Van Allen belt are in closer proximity to the earth due to the main magnetic field dipole offset. Since this region is localized, the dosage will remain fairly small. The reference [2] study has shown, for example, that with a shielding thickness of 4mm Al, the daily radiation for the nominal Station and COP 28.5^{o} orbit is .628 rad(Si)/day or approximately 230 rads(Si)/yr. This environment can be tolerated by a wide selection of components and will not be a procurement issue.

At higher orbital inclinations, shielding from the magnetic field is reduced as the orbit approaches the magnetic poles such that a polar orbit intercepts the free-space Solar Cosmic Ray (SCR) particle fluence over approximately of its orbit. POP Hardware with 3mm Al shielding, for example, would accumulate approximately 2K rads(Si) per year under solar minimum conditions and approximately 25K rads(Si) per year during solar maximum conditions yielding an estimated total 10 year dose of 100K rads(Si). This environment represents a severe design/procurement hardship.

Full shielding is not a viable solution because of weight and volume penalties. Figure 3.5.1 – 2 provides data on the effectiveness of shielding and indicates, for example, that the reduction of the 100K rads(Si) referenced above to a more tolerable 1K rad(Si) dose would require shielding thickness of approximately 4.5 ins Al. Aluminum is typically the shield material for these studies however, other more efficient materials may be developed. The problem is complex in that secondary effects must also be considered in the low Z/high Z shielding equation.

The options therefore available are: (1) select hardware (components and circuit designs) to exceed the specified requirements, (2) provide selective shielding, or (3) perform periodic hardware replacement. Table 3.5.1 – 6 provides a qualitative assessment of these options.

Table 3.5.1 – 6

RADIATION ENVIRONMENT (TOTAL DOSE)

RISK PARAMETERS	OPTION #1 SELECT TO MEET REQ'TS	OPTION #2 SELECTIVE SHIELDING	OPTION #3 PERIODIC REPLACEMENT
COST	HIGH (3)	LOW	HIGH (1)
SCHEDULE	HIGH (3)	LOW	LOW
PERFORMANCE	HIGH (3)	HIGH (2)	LOW
RELIABILITY	LOW	LOW	LOW
MAINTAINABILITY	LOW	LOW	HIGH (1)
SAFETY	LOW	LOW	LOW

1) Due to repetitive (replacement) effort and costs
2) Feasibility risk
3) High development risks

3.5.1.2.4.2 Options Characterization

a. Select Hardware To Meet Requirements

This option addresses the task of providing components and hardware with a tolerance to the projected total dose requirements and that exhibit an avoidance of latch-up when exposed to the radiation environment. Unfortunately, as the component industry moves toward increasing densities to satifisfy the industry needs for faster, more integrated parts, the resulting susceptibilities to radiation, particularly the single event upset, increases. There is a concern for the developing VHSIC technologies for example, in that the reduces feature sizes will provide a susceptibility that is orders of magnitude higher than that of previous parts.

The validation of radiation tolerant parts requires a comprehensive test and analysis program on samples of the target lots. This effort will be required on components no matter what level of tolerance must be guaranteed, however, there appears to be a considerably lower probability of success in meeting the higher levels. This effort will clearly be one of the key development issues for the SSPE's and in particular the polar platform.

b. Selective Shielding

Total shielding, as discussed in the description section, is not viable, however, it may be feasible on a very limited basis on the polar platform to provide shielding on selective components that cannot meet the required radiation tolerance. The advantages of this approach is that it can salvage an otherwise unacceptable design. The disadvantage is the additional design effort, weight and volume penalties of the shielding.

c. Periodic Replacement

This option addresses replacement/refurbishment of susceptible hardware on a periodic basis to preclude system failures. This approach would be compatible with the servicing programs established for platform hardware. In addition, technology insertion programs may also take advantage of improvements in radiation susceptibility. The net effect is that hardware may not, in reality, be required to demonstrate a 10 year total dose tolerance.

3.5.1.2.5 Spacecraft Charging

Space craft in orbit can build up electric potentials to thousands of volts with reference to ambient plasma, such that large differential voltages can appear between packages/circuits and structure. Electrical arcing can result to permanently damage the associated electronics. The required conditions for this phenomenon are a complex combination of high altitude (geo-synchronous), sun-angles, particle flux densities and magnetic storm activity. At the lower altitudes of the SSPE's, this phenomenon is not as applicable such that use of appropriate conductive materials and effective grounding practices will eliminate the potential problem. No particular procurement issues have been identified for this subject.

3.5.1.2.6 Electro-Magnetic Interference

This subject addresses the emission of noise levels either radiated or conducted on signal or power busses that may interfere with other equipment and the susceptibility of equipment to radiated and conducted noise. There are spectral bands in the solar spectrum that must be reviewed in the design/qualification tasks, however, the primary RF spectral regions are the man-made earth based and on-board sources. No particular issues have been identified. Sound engineering practices will be employed, coupled with adequate EMI controls such that there will be no significant risk to SSPE operational performance.

3.5.1.3 Prototypes/Test Beds

From the viewpoint of procurement activities, the test beds may serve a function during the selection process of insuring compatibility and performance of candidate hardware/software products. The only identifiable issue is that of the availability of test-beds and their utilization within the program controls requirements, i.e. quality control, configuration management, etc.

3.5.1.4 Procurement Strategies

There are four basic procurement strategies in hardware/software procurement: 1) commercially available (off-the-shelf) products, 2) contractor/vendor provided hardware, 3) Government Furnished Equipment, and 4) second sourcing. Table 3.5.1 - 7 provides a qualitative assessment of these options; the merits and applicability of each is discussed below.

TABLE 3.5.1 - 7

PROCUREMENT STRATEGY OPTIONS

RISK PARAMETERS	OPTION #1 OTS PRODUCTS	OPTIONS #2 CONTR/VENDOR PROVIDED	OPTION # GFE	OPRION #4 2ND SOURCE
COST	LOW	MOD - HIGH	LOW	MOD - HIGH
SCHEDULE	LOW	MODERATE	LOW	MODERATE
PERFORMANCE	MODERATE	LOW	MODERATE	LOW
RELIABILITY	MODERATE	LOW	MODERATE	LOW
MAINTAINABILITY	MODERATE	LOW	MODERATE	LOW
SAFETY	MODERATE	MODERATE	LOW	LOW

NOTE. Qualification costs not considered in evaluation

3.5.1.4.1 Off-The-Shelf Products

This option addresses utilization of commercially available products that match Space Station requirements to a reasonable degree. The major difficulty, as discussed earlier, is the suitability of such hardware to the potential space environments. Special installations and/or handling equipment will in general be required to utilize such hardware; however, the additional effort may be a relatively small cost, thus rendering a clear advantage to this approach.

3.5.1.4.2 Contractor/Vendor Provided Hardware

This option is typical of special project hardware where-in the products are uniquely specified and competed between several contractors. The effort generally requires concept definition, documentation and progress review plus some level of effort management overview. the costs are significantly higher, particularly for low production run efforts, however, the resulting products have the unique configurations/capabilities/suitabilities required by the program.

3.5.1.4.3 Government Furnished Equipment

This option addresses hardware that is provided by the government with no development/procurement activity by the integrating agency. Equipment in this category may be immediately available 'off-the-government-shelf' or may require additional production runs by associated contractors/vendors. In either case, the design/development effort is complete, therefore acquisition of such hardware can represent a significant cost savings to the program provided the appropriate qualification can be successfully completed. Additional potential advantages are that the government procurred equipment will generally be ruggedized and may also adhere to SSP applicable standards. Another potential scenario is that the hardware may be procured by NASA and imposed on the Space Station program to enforce particular programmatic issues, i.e. commonality, standardization, etc.

3.5.1.4.4 Second Sourcing

This strategy is regularly employed by both industry and government in cases where a product produceability risk is projected or has been demonstrated to be moderate to high. Production problems may be due to materials, state of the art processes, or even limited vendor/contractor resources, that are not applicable to the second supplier.

The disadvantages of this approach are the start-up costs for the second source, the additional resources required to oversee two suppliers, and the arbitration of configuration/performance change requests.

3.5.1.5 Summary

The above design/procurement issues will all be resolved with normal (as required) trade-offs in program costs, performance and operational requirements, plus execution of appropriate options.

The issues of commonality and the goal of replicating subsystems with the SSPE's may be in jeopardy however, particulary when addressing the radiation environments.

3.5.1.6 References

[1] W. W. Vaughan "National Environment Design Criteria For the Space Station Definition And Preliminary Design (First Edition)", TM-86460, NASA, Marshall Space Flight Center, Alabama, September 1984.

[2] K. A, Pfitzer, W. R. Yucker "Extravehicular Crewman Work System Study - Final Draft", MDC-H1660, MDAC, Huntington Beach, CA, Dec 1978.

[3] R. E. Smith, G. S. West, "Space and Planetary Environment Criteria Guidelines For Use In Space Vehicle Development", 1982 Revision (Volume 1), NASA, Marshall Space Flight Center, Alabama, 1983.

Table 3.5.1-3 a)

Orbiter Cargo Bay Random Vibration
Trunnion Supported Payloads - On P/L Keel Pin

Payload Weight *Less Than 10,000 Lbs.

o All Axes	20 to 60 HZ	.0023 G^2/HZ
	60 to 100 HZ	+9 dB/OCT
	100 to 300 HZ	0.01 G^2/HZ
	300 to 2000 HZ	-9 dB/OCT
	OVERALL = 1.9 GRMS	

Payload Weight *Greater Than 10,000 Lbs.

o All Axes	20 to 480 HZ	.0023 G^2/HZ
	480 to 2000 HZ	-9 dB/OCT
	OVERALL = 1.2 GRMS	

The associated time duration is 20 seconds per flight which includes a fatigue scatter factor of 4.

*Total payload weight is irrespective of the number of mounting points.

Table 3.5.1-3 (c)

Orbiter Cargo Bay Random Vibration
Longeron/Adapter Supported Payloads - At Orbiter Interface

o X Axis	20 to 100 HZ	+6 dB/OCT
	100 to 500 HZ	.03 G^2/HZ
	500 to 2000 HZ	-4 dB/OCT
	OVERALL = 5.4 GRMS	
o Y Axis (Fwd of Sta. Xo = 919)	20 to 40 HZ	+12 dB/OCT
	40 to 100 HZ	.06 G^2/HZ
	100 to 170 HZ	-6 dB/OCT
	170 to 600 HZ	.02 G^2/HZ
	600 to 2000 HZ	-9 dB/OCT
	OVERALL = 4.5 GRMS	
o Y Axis (Aft of STA. Xo = 919)	20 to 40 HZ	+12 dB/OCT
	40 to 500 HZ	.06 G^2/HZ
	500 to 2000 HZ	-4 dB/OCT
	OVERALL = 7.8 GRMS	
o Z Axis	20 to 100 HZ	+6 dB/OCT
	100 to 2000 HZ	.03 G^2/HZ
	OVERALL = 7.6 GRMS	

The associated time duration is 20 seconds per axis per flight which includes a scatter factor of 4.

Orbiter Cargo Bay Random Vibration
Trunnion Supported Payloads - on P/L Trunnion

Payload Weight *Less Than 10,000 Lbs.

o X Axis	20 to 50 HZ	.0015 G^2/HZ
	50 to 125 HZ	+9 dB/OCT
	125 to 300 HZ	.025 G^2/HZ
	300 to 2000 HZ	-9 dB/OCT
	OVERALL = 3.0 GRMS	
o Y Axis (Fwd of Sta. Xo = 919)	20 to 68 HZ	.004 G^2/HZ
	68 to 100 HZ	+9 dB/OCT
	100 to 380 HZ	.013 G^2/HZ
	380 to 2000 HZ	-9 dB/OCT
	OVERALL = 2.5 GRMS	
o Y Axis (Aft of Sta. Xo = 919)	20 to 68 HZ	.004 G^2/HZ
	68 to 125 HZ	+9 dB/OCT
And	125 to 300 HZ	.025 G^2/HZ
o Z Axis	300 to 2000 HZ	-9 dB/OCT
	OVERALL = 3.0 GRMS	

Payload Weight *Greater Than 10,000 Lbs.

o X Axis	20 to 50 HZ	.0015 G^2/HZ
	50 to 80 HZ	+9 dB/OCT
	80 to 480 HZ	.0063 G^2/HZ
	480 to 2000 HZ	-9 dB/OCT
	OVERALL = 2.0 GRMS	
o Y and Z Axes	20 to 68 HZ	.004 G^2/HZ
	68 to 80 HZ	+9 dB/OCT
	80 to 480 HZ	.0063 G^2/HZ
	480 to 2000 HZ	-9 dB/OCT
	OVERALL = 2.4 GRMS	

The associated time duration is 20 seconds per axis per flight which includes a fatigue scatter factor of 4.
***Total payload weight is irrespective of the number of mounting points.**

Table 3.5.1-4 ORBITER CARGO BAY INTERNAL ACOUSTIC ENVIRONMENT

1/3 OCTAVE Band Center Frequency (Hz)	Sound Pressure Level (dB) ref. $2X10^{-5}$ N/m^2	
	Lift-off	Aeronoise
	5 Seconds/Flight*	10 Seconds/Flight*
31.5	122.0	112.0
40.0	124.0	114.0
50.0	125.5	116.0
63.0	127.0	118.0
80.0	128.0	120.0
100.0	128.5	121.0
125.0	129.0	122.5
160.0	129.0	123.5
200.0	128.5	124.5
250.0	127.0	125.0 **
315.0	126.0	125.0 **
400.0	125.0	124.0 **
500.0	123.0	121.5
630.0	121.5	119.5
800.0	120.0	117.5
1000.0	117.5	116.0
1250.0	116.0	114.0
1600.0	114.0	112.5
2000.0	112.0	110.5
2500.0	110.0	108.5
Overall	138.0	133.5

*Time per flight does not include a scatter factor.

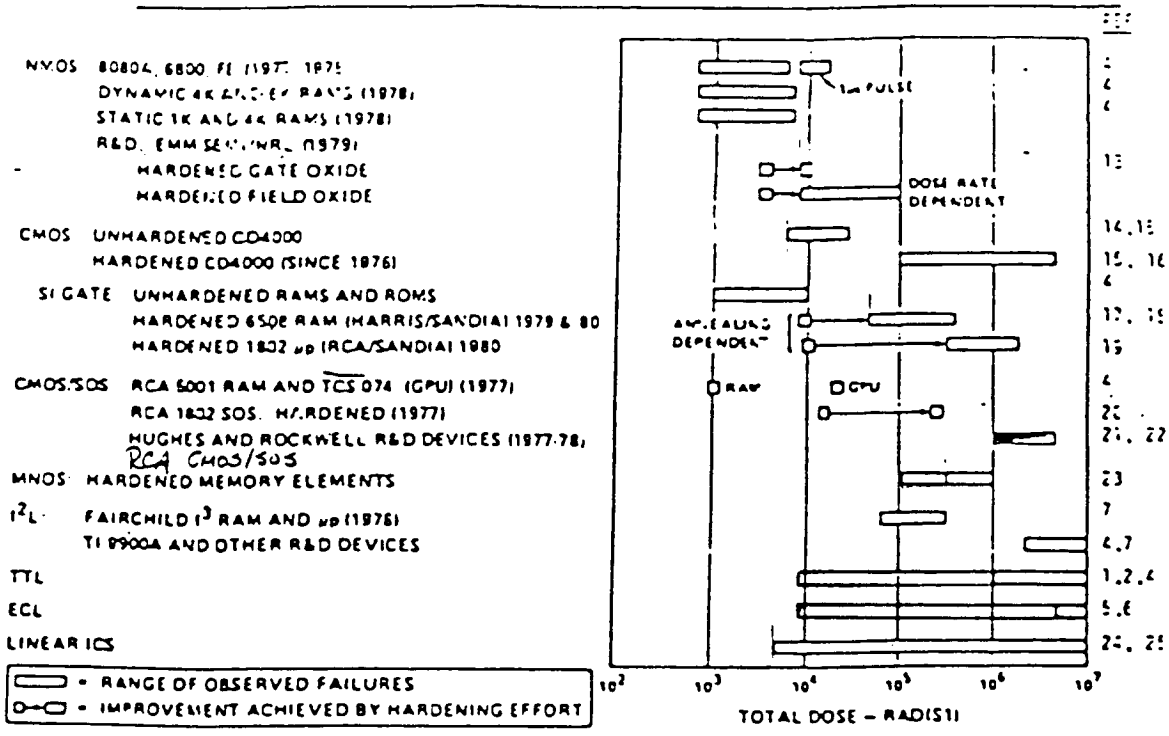
**NOTE: Narrow band discrete noise is radiated from the cargo bay vent doors during transonic/low supersonic flight. The noise radiated from any one vent is described below:

This environment is not intended for full payload exposure but only to those areas of the payload adjacent to a cargo bay vent opening.

One-third Octave Band Center Frequencies, Hz	Sound Power Level dB re 10^{-12} watts
	8 Seconds/Flight
250	128
315	136
400	130

Figure 1

TOTAL DOSE HARDNESS



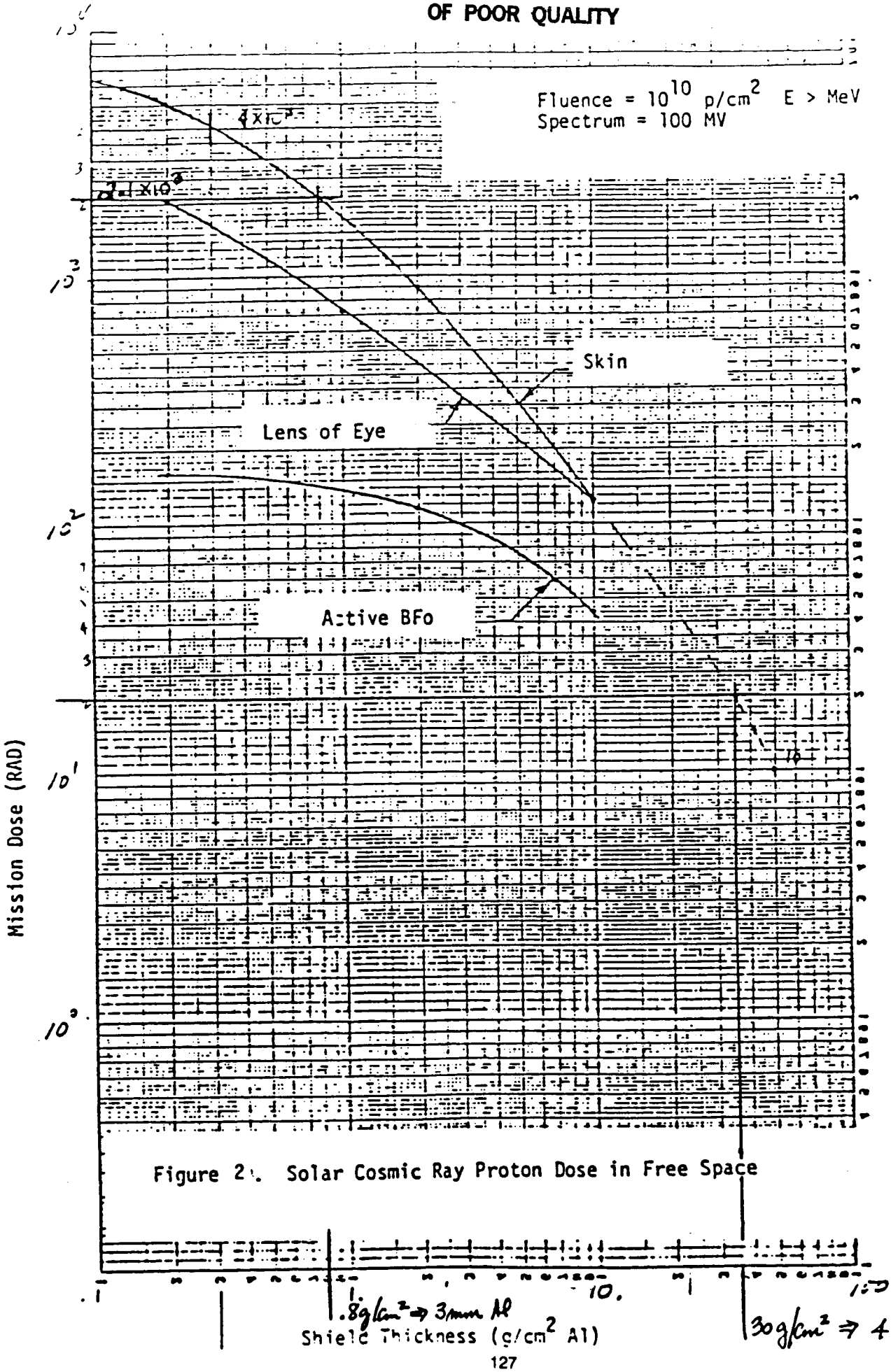


Figure 2. Solar Cosmic Ray Proton Dose in Free Space

3.5.2 Software Development Options Paper (Task 2)

The Phase B RFP for the Space Station states that "for software, commonality of programming languages, support software, operating systems, and user interface languages are major goals. The SSE (Software Support) Environment) is defined as the common software required for the development of applications software for Space Station flight and ground systems."

The goal of this paper is to identify those tools currently used by the software population which have demonstrated a significant productivity increase. In addition, where possible, specific aspects of the software development tools which show promise for future productivity gains will be stressed. Also, a certain amount of 'blue-sky-dreaming' will be incorporated, tempered with real-world activities, to project where software development will be in the near future.

Since the users of the SSE will be a large population of analyst, designer, programmer, tester, and tracker experts it must be emphasized, that these tools and controls should be integrated to provide an environment which facilitates their use.

Many tools examined here have evolved over the last decade to aid the software development process and many more integrated tool sets are on the horizon. Along with the tools, management and acquisition strategies options are characterized. The characterization of (1) tools, management and acquisition strategy options, (2) options for facilities to host the SSE, and (3) ergonomic issues are included.

3.5.2.1 Software Engineering

Software engineering is the disciplined application of methods, principles, procedures, and tools to ensure the development of reliable, understandable, modifiable and efficient software. This process as depicted in figure 1, is an iterative process of requirements definition, design, code, test and release. Because of the rising cost of software development, initiatives

within NASA and DoD combined with industry and academic efforts have provided much study of the software engineering phases. These efforts have provided insights into the phases of the development effort, the life cycle costs of those processes, and ways to automate and integrate the functions. The Ada* language and environment, the STARS initiative (Software Technology for Adaptable and Reliable Systems) and the DoD-STD-SDS (Joint Logistics Commanders, Joint Policy Coordinating Group on Computer Resource Management) are three examples of DoD initiatives which will have large impacts on software engineering.

*Ada is a registered trademark of the Ada Joint Program Office

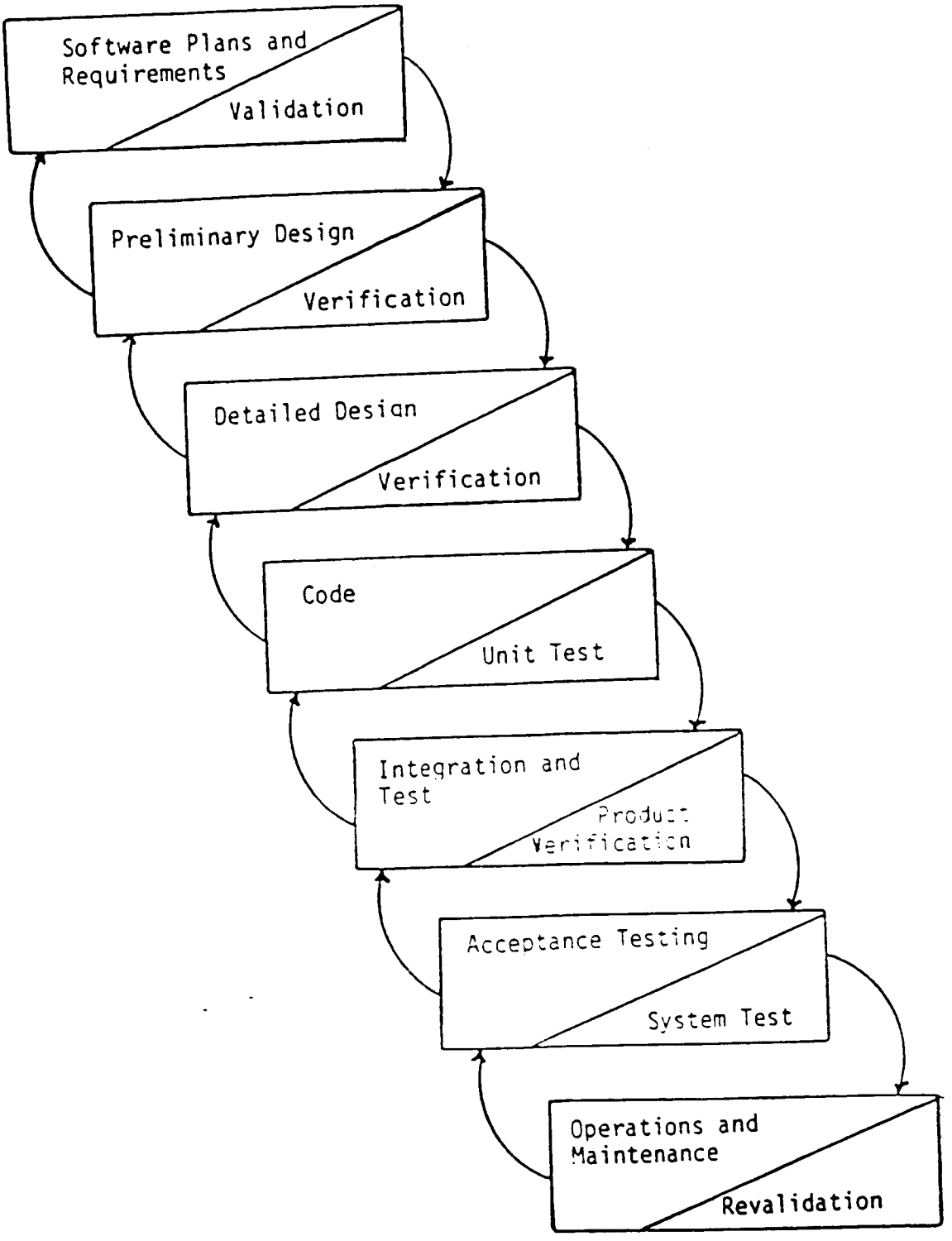


Figure 1 – The Software Life Cycle

ADA

There are approximately 38 Ada compilers currently in various stages of development, test, and use today (February 1985). In 1984, 14 compilers passed the DoD certification tests. Two significant DoD contracts for Ada software development environments are currently being worked. These are the AIE and the ALS. The ALS compiler has been validated and a preliminary MAPSE is being installed on VAX systems. Preliminary indications are that an AIE will soon appear which is designed for the IBM 370/VM architecture. Tools for the APSEs can be generated independently, and because they are done in Ada, they can be transported to all APSEs. This wide market feature will greatly encourage the development of APSE tools and should ensure a rich tool set for Ada users.

Software Technology for Adaptable and Reliable Systems (STARS)

The STARS program plans to look at all phases of the software life cycle, from both technical and management viewpoints. Through the STARS program, the DoD is seeking an integrated and automated software environment which covers the software life cycle. Technically, the program uses Ada and its environment as a foundation. Beyond that, it will address management practices, software acquisition strategies, increased personnel skill levels, increase the use of tools, and make advances in both software systems methodologies and software theory.

A product of the STARS program, the Software Engineering Institute (SEI), has already been formed. It is intended to be a vehicle through which emerging technologies will be developed, validated, and brought into practice.

DoD-STD-SDS

This is a joint services effort to standardize the way software is developed. DoD standard 2167 was recently released to address this issue. This standard will be used to describe how all software will be developed on DoD contracts. It identifies the produced documents, required methods, and techniques used when developing DoD software. This standard was initiated to avoid redundancy, improve productivity and increase management visibility for software development.

Environments to support program development are toolsets which attempt to address all the functions which must occur in a large software development project.

No toolsets automate all aspects of software engineering. (See Figure 2.) Major toolsets like UNIX have taken many years to mature. Many new toolsets are currently being developed and are attempting to present an integrated environment to the user. Some examples are: SEF by IBM/FSD, TAGS by Teledyne Brown, TEAMWORX by Cadre Technologies, FPE by SofTool, USE.IT by HOS, APCE by PRC, Structa by Tektronix, and ProMod by GEI.

Current NASA and DoD initiatives are concentrating on the Ada language and environment. The Ada Programming Support Environment (APSE) is currently being defined by the DoD. It will eventually address the total software life cycle model. It will be many years before full APSE's are addressing all phases of software engineering.

The minimal toolset or MAPSE has been defined and many are in varying stages of development. According to the DoD's STONEMAN, the MAPSE includes a text editor, formatted printer, translators, linkers, loaders, set-use static analyzer, control flow static analyzer, dynamic analysis tool, terminal interface routines, file administrator, command interpreter, and configuration manager.

Ada compilers are becoming common place and will be maturing rapidly. It was originally predicted that this would happen in 1980. Ada MAPSE's are, in the author's opinion, 3-5 years from mature releases. Other, more mature development environments such as UNIX and VM/CMS must be considered for the Space Station's initial SSE. Most software life cycle costs are not code related (rather specification, test, maintenance, CM) and can be addressed in these more mature environments. The Ada compilers could be immediately integrated into one of these environments and MAPSE tools added later as they mature.

	Requirements		Design	Implementation						Checkout								Maintenance				Management					Documentation									Multipurpose			
Function / Toolset	Requirements Tracing	Requirements Language	Design Support	Compiler/Assembler	Linker/Loader	Conditional Compilation	Control Flow Analyzer	Report Generator	Preprocessor Generator	Statement Execution Monitor	Interface Simulator	Source Debug	Formal Verification System	Test Case Generator	Symbolic Executor	Instruction Level Simulator	Environment Simulator	Global Cross-Reference	Disassembler	Call Structure Analyzer	Timing/Performance	Configuration Management	Project Control	Fault Report	Standards Auditor	Graphics Generator	MIL/Spec Generator	Word Processing	Typesetter	Documentation Templates	Interface Documenter	Text Primitives	Speller	Data Base File Manager	Text Editor	Pretty Printer	File Compare	Mailbox
DoD MAPSE				X	X					X	X	X						X			X	X					X						X	X	X	X	
DoD APSE	X		X	X	X					X	X	X	X					X			X	X	X	X			X	?					X	X	X	X	
AF SDVS				X						X						X	X					X															
NASA HAL/S			X	X	X			X									X	X	X		X	X											X			X	
Softool 80				X	X		X		X	X		X						X		X	X	X			X					X	X						
Navy MIASS				X	X											X																					
Army ADA				X	X							?						?				X						X						X	X			
BITS				X	X	X												X	X	X						X	X	X						X	X	X	X	
Navy SDL		X	X	X	X							X									X		?										X				
AF CSDP			X	X	X		?	X				X		X								X	X	X				X						X			X
TI ISS				X	X		?			X		X		X							X	X		X				X	X					X	X			X
GTE Phoenix	X			X	X									X									X														
Boeing SS				X	X	X										X		X		X		X											X	X	X	X	
FASP			X	X				X		X						X							X										X	X			
KERN				X					X																		X				X		X	X			
UNIX				X	X	X			X	X		X						X			X							X	X			X	X	X	X	X	X

Figure 2. Contemporary Toolsets vs Functional Content

The DoD Ada Programming Support Environments that are under-going development include the Ada Integrated Environment (IBM 370/VM) the Ada Language System (VAX 11/780/VMS), and the Ada Language System/Navy (VAX/VMS).

The Ada Integrated Environment (AIE)

In April 1982, the Air Force (Rome Air Development Center) contracted Intermetrics, Inc. to implement a Minimal Ada Programming Support Environment (MAPSE) entitled "Ada Integrated Environment" (AIE). The AIE is designed for use in the development of embedded computer system software and can accommodate a variety of users, skilled and unskilled, from project managers, program designers and developers to documentors and clerical personnel. The AIE contains a virtual operating system called the kernel or KAPSE (Kernel Ada Programming Support Environment) that isolates tools (both system and user) from hardware dependencies. The system tools consist of a production quality Ada Compiler and symbolic debugger; a program integration facility with program library management and linking/loading tools used to develop Ada programs; a data base manager with a complete file management system; and a command language processor which allows user interaction with tools and other operating system routines. The AIE will be hosted on the IBM 370 architecture and can co-exist with other operating systems (e.g., OS/VS1, CMS, UTS, etc.).

The Ada Language System (ALS)

In April, 1980, the Army contracted SofTech, Inc. to develop the Ada Language System (ALS), an integrated programming environment designed to aid in the development and maintenance of Ada programs. The ALS is designed to support large software systems throughout their life cycle. In particular, the ALS was designed with the requirements of embedded computer system development in mind. The three major components include a file structure (called the environment database), a set of tools, and a mechanism through which the tools are invoked (i.e., the command language interpreter). The ALS will be hosted on DEC VAX 11/780 architectures.

Ada Language System/Navy (ALS/N)

The Ada Language System/Navy is a minimal Ada programming support environment designed to provide support for program generation and execution of Ada application programs targeted for Navy standard embedded computers and peripherals. The system is composed of extensions to the Army's ALS that are the minimum required to support projects using the Navy's standard embedded computers. [5]

3.5.2.1.1 Requirements Analysis Tools

3.5.2.1.1.1 Requirements Analysis Description

Traditionally, requirements for software systems have been hard to produce. Prose specifications are generated by developers and reviewed by customers. However, written requirements cannot reveal important aspects of the final system such as performance, function, usability and reliability. Not until system integration can it be demonstrated that the proposed system meets the customer's requirements. By then, problems are costly to detect and correct.

Ideally, requirements analysis tools should allow for a precise and verifiable specification of software requirements and allow an automatic analysis of the features of the system. This will keep faulty design decisions from being propagated through the system during implementation. Tools should also allow for automatic tracking of requirements through system design, implementation and testing of the final product.

Currently all requirements analysis tools support a methodology for developing or demonstrating software analysis.

The following options are discussed:

- SA - Structured Analysis
- SREM - Software Requirements Engineering Methodology
- PSL/PSA - Problem Statement Language/Problem Statement Analysis
- Warnier/Orr

Table 1 compares key characteristics of these options.

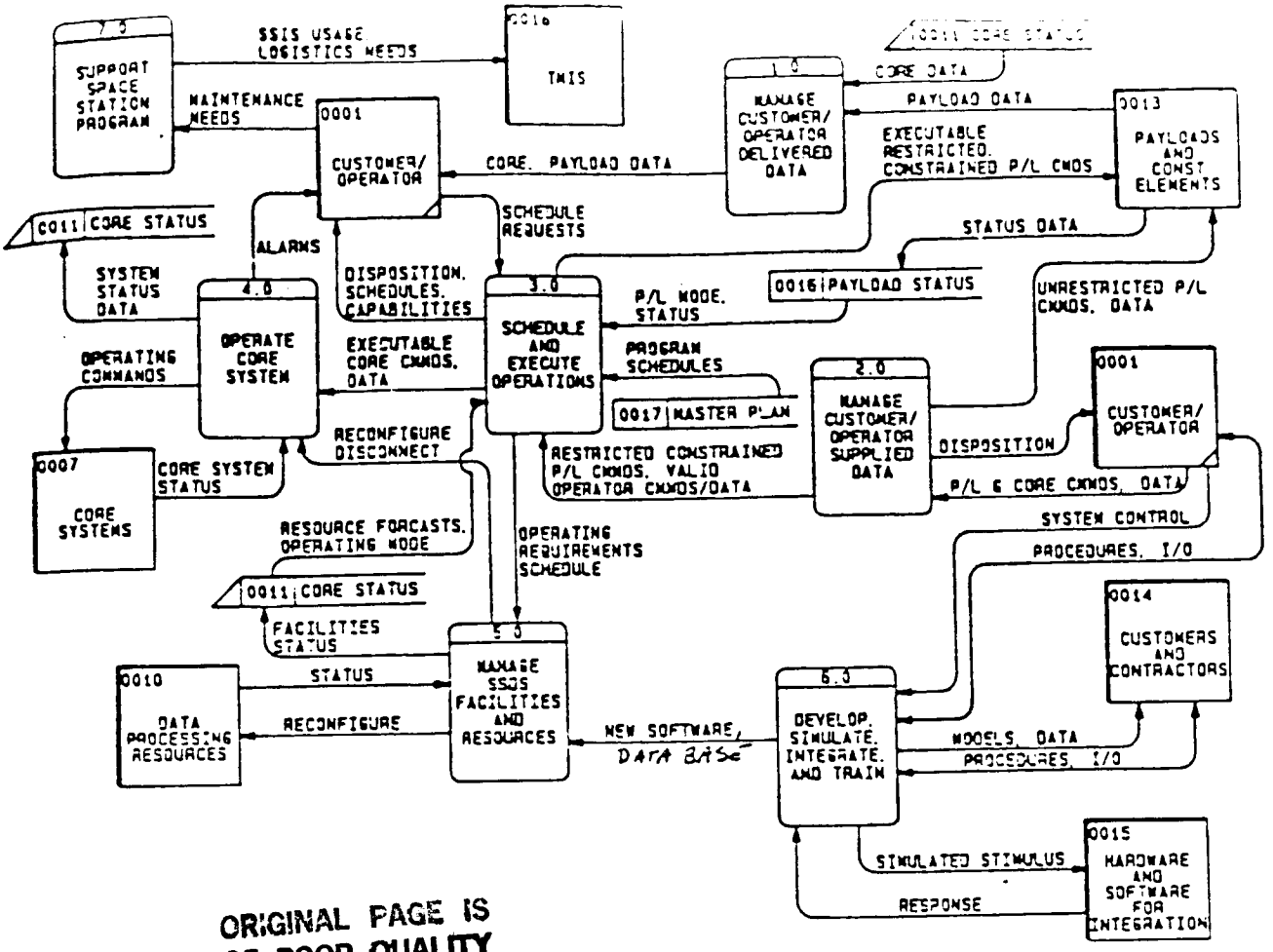
Feature	SA	SREM	PSL/PSA	WARNIER/ORR
Host System	IBM PC VAX APOLLO	VAX	IBM VAX HP UNIX	IBM PC
Maturity	High	Medium (10 years)	Low (6 years)	Low (2 years)
Key Features	Methodology uses data flow diagrams Iterative refinement	Control flow methodology Data base of processes and transfer function Consistency checks Simulation SREM users group	Object oriented methodology Maps real world system into data base of entities and relationships Provides graphical depictions Code generation may be coming	Formal requirement specifications Code generation Data structure diagrams Program structure chart
User Experience	Can be used in conjunction with PSL/PSA User's group (SDT) Many companies have classes Easy to modify	Thorough methodology Initial manual process Not easy to modify Complexity similar to PSL/PSA User's group is addressing user friendliness TRW has classes	Time consuming Large resource user Lots of options Expertise required for effective and efficient use New graphic interface	Formal classes required Graphic interface
Cost	$900-$30,000	$10,000/year	$45,000	$2000

Table 1 – Requirements Analysis Tools

3.5.2.1.1.2 Requirements Analysis Tools Options Characterization

- Structured Analysis - SA is a discipline incorporating simplified system modeling and the early use of data oriented techniques. [1][7] The underlying concept is the building of a logical (nonphysical) model of a system, using graphical techniques that enable users, analysts, and designers to get a picture of the system and how its parts fit together to meet the user's needs. This is accomplished with logical data flow diagrams (DFD) (See Figure 3) that specify precisely "what" the system has to do, leaving the designer free to specify "how" it can be done. The methodology involves building a system model top-down by successive refinements, first producing an overall system data flow, then detailed data flows, and finally defining the data structure and process logic. The DFDs and related documentation (data dictionaries, data immediate access diagrams, and process logic) make up a comprehensive account of a system in terms of a logical, function specification. They also provide the basis for step-wise refinement of requirements in a structured and controlled environment. Structured Analysis has become extremely popular and as such, many tools to support the technique have been developed or are in development. The following is just a few examples of current systems.

Company	Tool	Host	Cost
Tektronix, Inc.	Structa	VAX/VMS/UNIX	$14.2K
Cadre, Inc.	Analyst Workbench	Apollo	$24K
StructSoft, Inc.	PCSA	IBM/PC	$900
McDonnell Douglas	DFD Draw	IBM/PC	$500
Yourdon, Inc.	Analyst Toolkit	Wang PC	$3.5K
Intech, Inc.	Excelerator	IBM/PC	$10K
GEI, Inc.	ProMod	VAX/VMS IBM/PC	$25K



ORIGINAL PAGE IS
OF POOR QUALITY

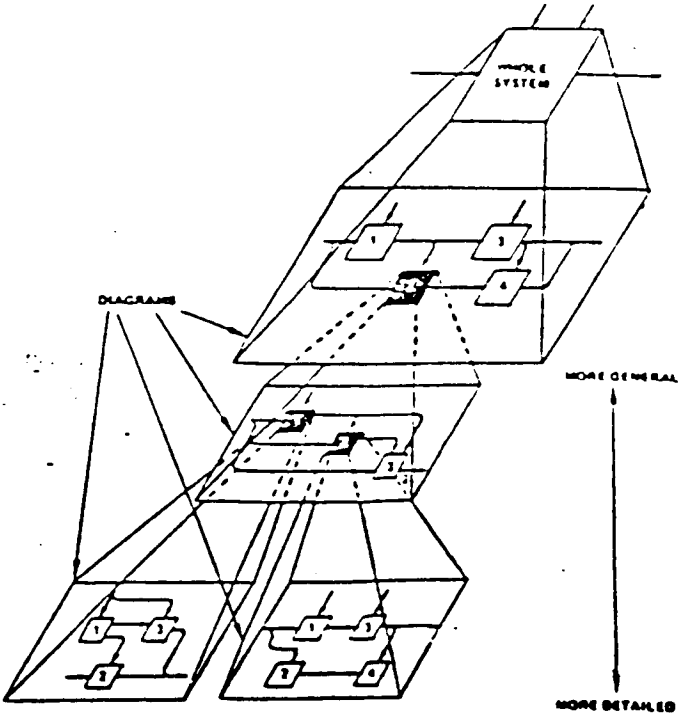


FIGURE 3 – DATA FLOW DIAGRAMS (STRUCTURED ANALYSIS)

- SREM – SREM (software requirements engineering methodology) [2] is the requirements definition tool in the DCDS (distributed computer design system) tool set. DCDS provides a software engineering and development environment for the definition of requirements, specifications, designs, code, aids for verification and validation, and documentation of real-time software. It consists of unified systems of methodologies, supporting software tools, utilities, and analysis techniques. The system covers the entire data processing life cycle, starting with a definition of the system requirements and ending with the tested data-processing hardware and software, including operation and maintenance requirements.

  DCDS procedural techniques define the sequence of steps to be taken in the development of software systems (See Figure 4). These steps represent a formalism which identifies the data base contents, produces outputs in increments, and provides the criteria for completeness/correctness of outputs. The steps are

  A. System Requirements Engineering Methodology (SYSREM): defines system requirements.
  B. Software Requirements Engineering Methodology (SREM): defines system software requirements.
  C. Distributed Design Methodology (DDM): develops a distributed design.
  D. Module Design Methodology (MDM): defines detailed design.
  E. Test Design Methodology (TDM): defines test plans and procedures against SYSREM and SREM requirements and records test results.

- PSL/PSA – PSL/PSA is a tool to aid in the precise definition of system specifications [3]. These specifications can include both requirements and design. PSL/PSA is composed of two components. PSL (problem statement language) is a language used to specify software systems requirements and designs. The model as defined in the PSL is maintained in an entity relationship (E/R) data base. The PSA tool (problem statement analyzer) is used to inspect the model for consistency and completeness. PSA supports report generation and query capabilities. See Figure 5.

A system is defined with PSL by mapping its objects and relationships into the entities (e.g., process, processor, event, set, input and output) and relationships (e.g., generates, performs, interrupts, and collection of) of a conceptual PSL E/R model. The objects and relationships are stored in the data base in a way which defines their dependencies and interactions and allows PSA to automatically act on them.

ISDOS now has a data flow diagramming tool to interface with PSL/PSA to graphically represent the analysis of a software system. The tool is called STRUCTURED ARCHITECT.

- Warnier/Orr - The Warnier/Orr technique for developing, analyzing, and representing software systems involves a technique for decomposing the system observing the data and data structures. This technique often called Data Structured Systems Development (DSSD) has been an accepted method since 1970. DSSD incorporates the data architecture or data structure approach to design and has evolved into a complete systems development methodology.

The tool which supports the DSSD technique is called STRUCTURE(S) and is used throughout the software lifecycle. It automates the production and maintenance of systems from analysis through code generation.

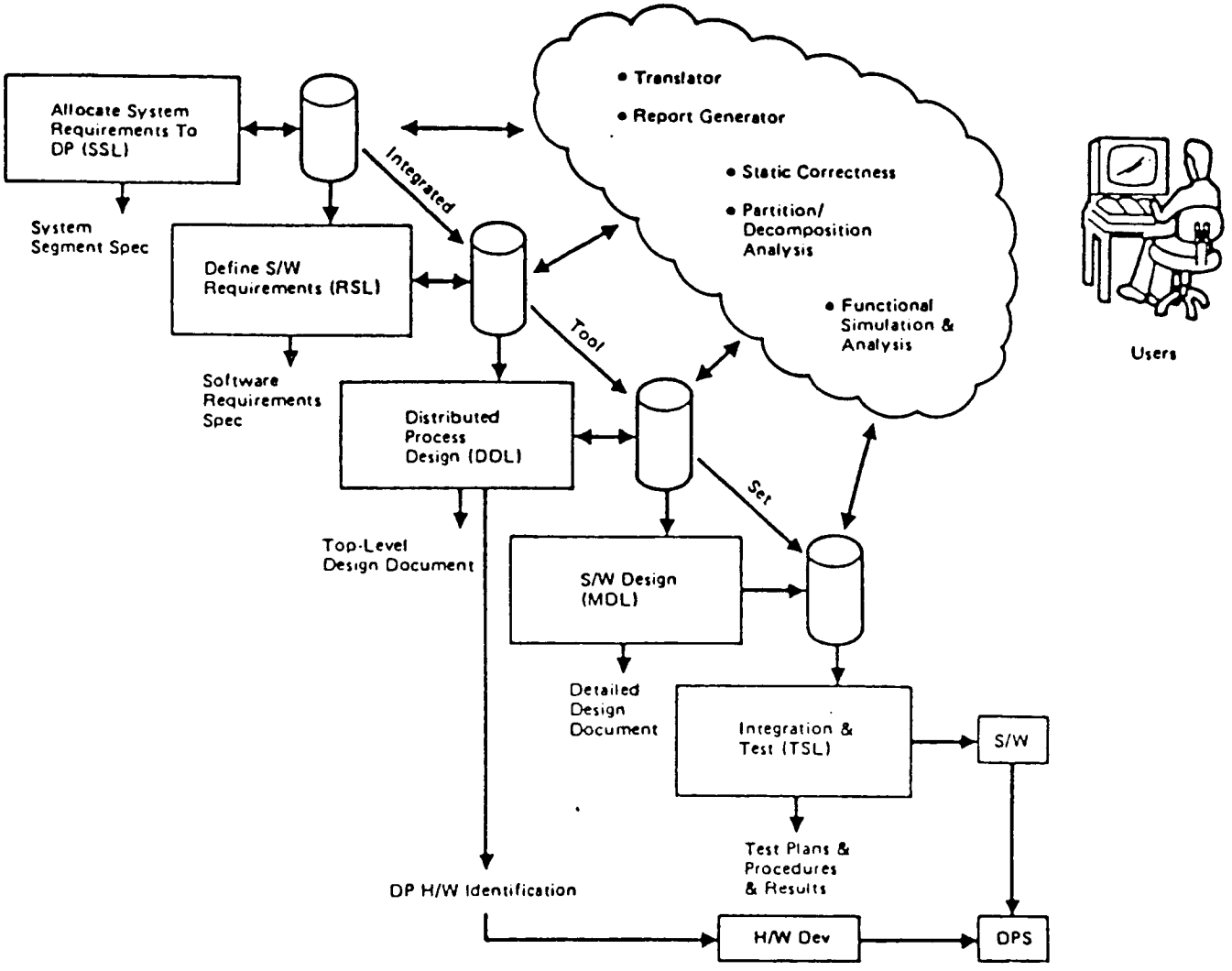


Figure 4 - Overview of DCDS

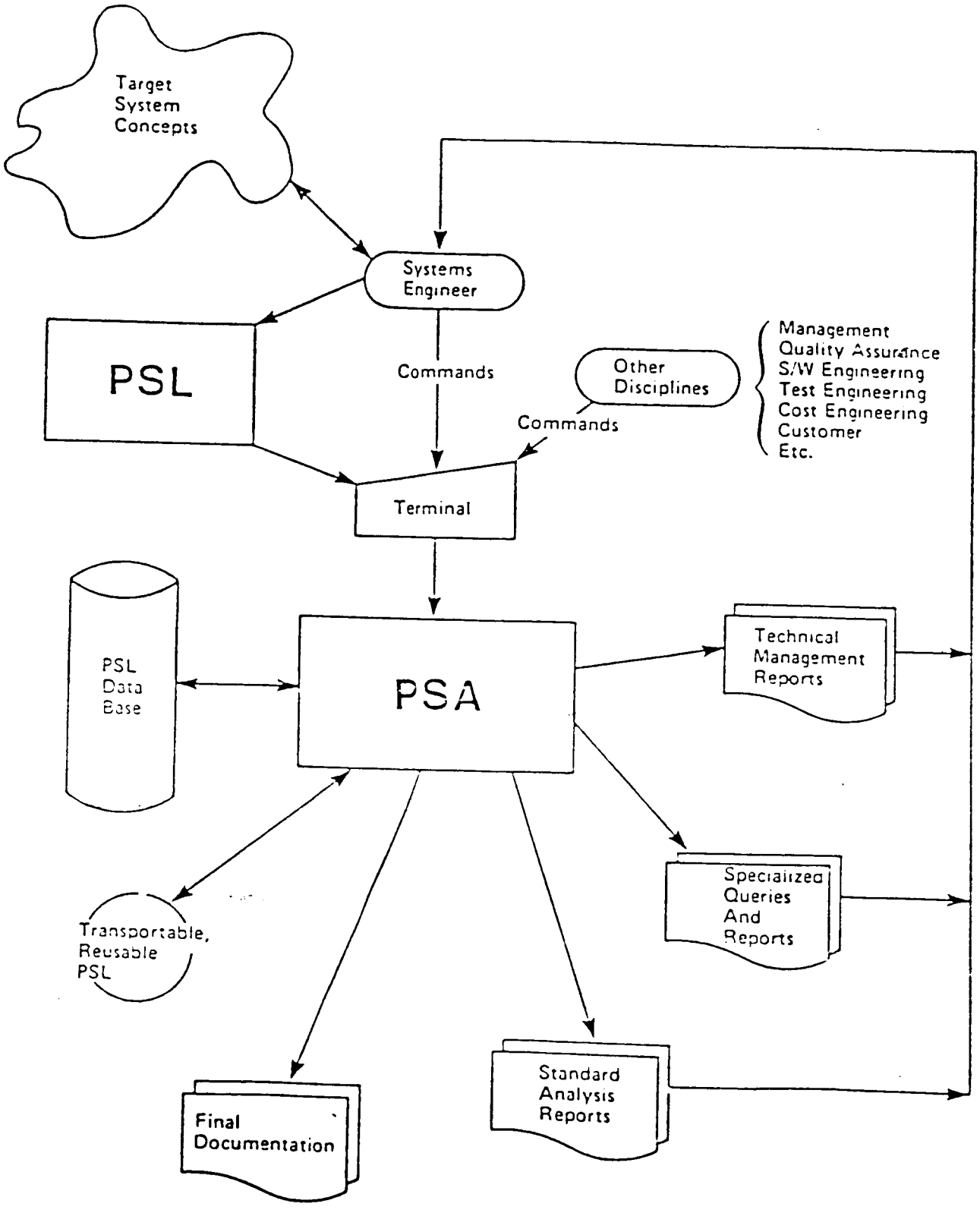


Figure 5 – The PSL/PSA System

3.5.2.1.1.3 Requirements Analysis Tools Projected Capabilities

Great strides in requirements analysis for both systems and software systems are certainly on the near horizon. Many colleges, universities and private corporations are developing Formal Specification Languages (FSL). One such FSL is being generated by Reasoning Systems, Inc., A. in Palo Alto, California. These FSLs will demonstrate before implementation the exact functions a system will have, then the system will automatically be generated from the specification.

With the advent of knowledge based and artificial intelligent systems within 5 years time an analyst will be able to sit at an automated workbench and verbally input the system and have it checked for accuracy and consistency. This type of system will even be able to generate a prototyped system for further evaluation by the analyst/customer.

3.5.2.1.1.4 Requirements Analysis Tools References

[1] Chris Gane and Trish Sarson, Structured Systems Analysis: Tools and Techniques McAuto, McDonnell Douglas Corp., St. Louis, Missouri, 19/7

[2] Mark Alford, "Requirements for Distributed Data Processing Design", IEEE, 6/79, CH1445, 001500.75

[3] W. E. Beregi, "Architectural prototyping the software engineering environment", IBM Systems Journal 23, No. 1, 4-18 (1984)

[4] Pieter Mimno, "A New Technology for Mathematically Provable Software", Computerworld, Oct. 11, 1982

[5] DoD APSE Analysis Document Version 1.0, 18 September 1984

[6] DoD "Stoneman" requirements for the APSE

[7] Tom DeMarco, Structured Analysis And System Specification, Yourdon Press, 1978.

3.5.2.1.2 Design Tools

3.5.2.1.2.1 Design Tools Description

Design tools assist the designer in implementing an interpretation of the system from WHAT the system should do into HOW the system will implement the user's needs. This activity of transformation is the design process, of which there are several well defined and mature techniques or methods. These methods all have automated tools except in the case of the object oriented technique developed to support the Ada environment. With the advent of the APSE tools, automated support of object oriented design will be commonplace.

Design tools tend to be graphic in nature, since we deal with pictures better than text, and represent the relationships of the system and its structure.

The following options are discussed:

- Structured Design
- Object Oriented Design
- Data Structures Design
- DCDS
- Rapid Prototyping
- PDL/Ada

3.5.2.1.2.2 Design Tools Options Characterizations

- Structured Design - Structured (or Composite) Design is a software design methodology which seeks to characterize the problem decomposition or modularization process [1]. It does so by analyzing the fundamental types of decomposition: source/transform/sink, transactional, functional, and data structure. Structured design proponents feel that certain types of decomposition yield systems which are easier to implement and maintain.

Structured Design also addresses the relationships between modules and the strength of modules (See Figure 6 - Structure Chart). Module coupling (content, common, external, control, stamp and date) is an analysis of all the kinds of ways that modules may be dependent on other modules. Module strength (coincidental, logical, classical, procedural, communicational, functional and informational) is an analysis of the relationships among the elements within a single module. Once again good structured design implies creating modules with high strength and using informed judgement when deciding on the type of coupling to create between modules.

The strength of Structured Design is its 'rules' for evaluating a given design. A designer can refine the requirements into a design. Then a tool can assess the design against a set of criteria. [7]

Tektronix, Inc. and Cadre, Inc have tools in development which support the Structured Design methodology. Hughes Aircraft Company has proprietary tools which support the Structured Design methodology, and Intech, Inc. has developed a tool which supports the drawing of structure charts, but does not support the 'rules' for Structured Design.

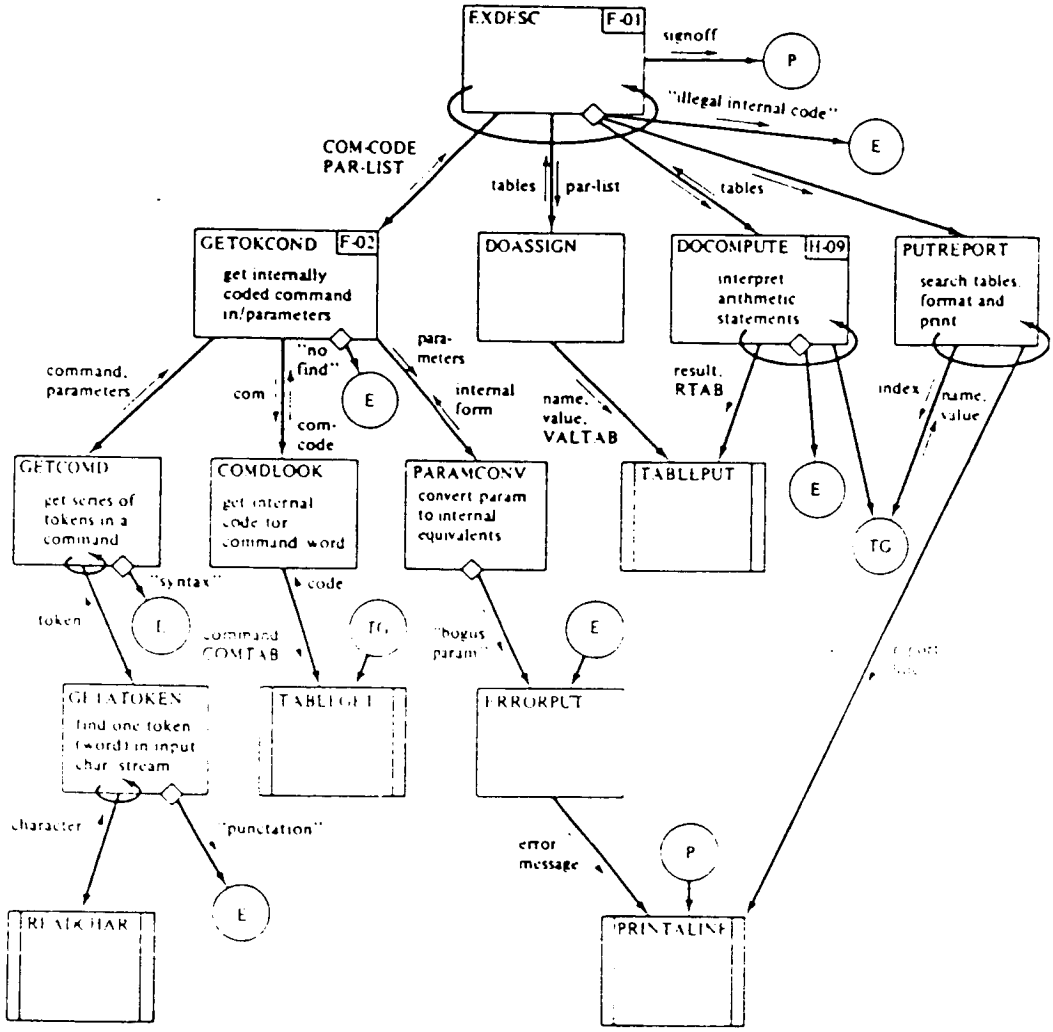


Figure 6 – Structure Chart

- Object Oriented Design – Object oriented design is a software design methodology which emphasizes data objects and design decisions in the modularization process rather than function and processes [2]. All functions that process the objects or reflect the design decision are then included in one module. The goal is to have changes impact only the single module.

  It has been shown that decomposing a system based on a flowchart of a process produces modules that are coupled in ways which cause change to be more difficult [2]. However by modularizing around the data and the manipulation of the data, the difficult design decisions (which are likely to change) are hidden from the rest of the system. These modules however will not correspond to the flowcharted steps of the process and execution implies many more calls and returns among modules than in a flowchart decomposed system. Efficiency concerns can be addressed by having the development environment create the run time flowchart oriented system by collecting the necessary code and data from the object oriented modules.

- Data Structures Design – This decomposition technique is based on the premise that the program's structure should reflect the correspondences between the structures of the input and output data of the program. The program decomposition, therefore, is based not on data flow, but on data structure of the input and output streams. The technique is primarily oriented towards the design of the program logic rather than the drawing of boundaries defining module interface. [3] [1] [6]

  The Warnier/Orr technique supports this methodology. The STRUCTURE(S) tool supports the data structures design by describing a design by means of a diagramming technique.

- DCDS – (See SREM under requirements and analyses tools)

- Rapid Prototyping - Rapid prototyping [4] is the process of building a model of a final system. The model exhibits the external and some internal characteristics of the system and is generated in a fraction of the time of a conventionally built software system. The model is then used to verify that the design meets the users requirements effectively. This allows systems designers to experiment with more options and reduces design errors which are very expensive to remove if not found until the implementation phase. Prototyping has become an excellent method to test, review, and evaluate portions of the system under development [8] and is used extensively, especially to demonstrate the user interfaces and timing conditions of a system. There are several tools in development which are being used for system prototyping. The following companies are actively involved: DARTS Technology, General Dynamics, San Diego, California and USE, Anthony Wasserman, University of California at San Francisco.

  If the rapid prototyping is done in a language such as Ada then much of the prototype may actually be used in the final product after addressing areas which the model indicated would not meet performance requirements.

- PDL/ADL - PDL (Program Design Language) is a method of design specification which uses English language to represent the design. PDL is used as a detail design technique, once a module decomposition process has taken place the data and processes of each module are specified in a PDL. PDL has the advantage of being totally text oriented and is very easy to automate compared to pictorial design specifications such as structure charts.

  The following tools are some of those available for processing PDL;

  - PDL Formatter - Caine, Farber and Gordon
  - SDDL - Jet Propulsion Labs/Cal Tech, Pasadena, California
  - PDL Formatter - Software Engineering Facility, IBM/FSD

ADL – Ada Design Language uses Ada as the PDL which has the added advantages of enforcing the object oriented design methodology, providing the strong data typing capabilities, promoting the program library reusable code aspects of Ada, and providing the package oriented program structure [5]. Ada is a high level language and also a candidate for a prototyping language. When the appropriate tools evolve, doing the design in ADL allows for prototyping and a very fast and error free program development phase. One of many ADL's on the market is BYRON by Intermetrics, Inc., Cambridge, Mass.

3.5.2.1.2.3 Design Tools Projected Capabilities

The transition from analysis to design (from the WHAT to the HOW) has become increasingly easier. With the advent of rapid prototyping, design evaluation and graphic oriented tools the design phase of software development is becoming shorter and more refined. Surely, we will see within the next few years a tool which takes the analysis of a system and produce a 'first cut' of the design. In fact, GEI's ProMod tool already does this in textual form. Until we can add the knowledge based rules to a design, the intuitive process of the knowledge worker is necessary. The five year prediction certainly will include automated tools to produce a design based upon the analysis, along with the required documentation.

The impact of reusable software packages, specifically the Ada packages, will see a distinct shortening of the design phase as evidenced by the projects looking into common software which is built everyday. (Common Ada Missile Packages, McDonnell Douglas Corp. – a DoD contract). When Ada packages become commonplace, we will see an amazing productivity increase both in shortening the development time and the increase of 'futuristic' software development.

3.5.2.1.2.4 Design Tools References

[1] G. F. Myers, Composite Structured Design, New York, N.Y., Van Nostrand Reinhold, 1978

[2] D. L. Parnas, "On the criteria to be used in decomposing systems into modules", Communications of the ACM 15, No. 12, 1053-1058 (1972)

[3] M. A. Jackson, Principles of Program Design, London: Academic Press, 1975

[4] W. E. Beregi, "Architectural prototyping in the software engineering environment", IBM Systems Journal 13, No. 1, 4-18 (1984)

[5] Grady Booch, Software Engineering with Ada, Menlo Park., CA., Benjamin/Cummings, 1983

[6] Ken Orr, Structured Requirements Definition, Topeka, Kansas, Ken Orr and Associates, Inc., 1981

[7] Ed Yourdon, Larry Constantine, Structured Design, Englewood Cliffs, N.J., Prentice-Hall, Inc., 1979.

[8] Bernard Boar, "Applications Prototyping", SDFV Presentation, February 1984.

3.5.2.1.3 Code Generation

3.5.2.1.3.1 Code Generation Description

The process of code generation takes the detailed design specification and creates source code in a programming language which executes on a target machine to perform to system requirements. Certain characteristics of code generating techniques arise independent of the programming language used. Some of these techniques are characterized here:

- Interactive
- Automated
- Production libraries

3.5.2.1.3.2 Code Generation Options Characterization

- Interactive - Interactive code generation is a process where an automated tool aids the programmer in generating code by providing a template to help define the structure of the program. The coding process is speeded up and made more error free because the programmer has less work to do in specifying the loops (IF-THEN-ELSE, etc). The Language Sensitive Editors (LSE) from Digital Corporation support several languages, including FORTRAN, Ada, C, and Pascal. The LSE in the IBM/FSD Software Engineering Facility is called DSS. DSS is table driven and supports several languages including Ada and PDL/Ada. The VI editor which is the standard editor on the UNIX systems supports interactive code generation for the LISP language. All LSE's are aware of the language syntax required for the code being entered on an automated system. All correct prefixes, suffixes, structure and specific syntax for the given language is automatically entered by the LSE during the entry of the code by the programmer. This eliminates the minor problems of misplaced or missing syntax which ultimately cause major compile or runtime errors. Easy access to terminals is a requirement of this option.

- Automated code generation, seemingly futuristic, is in use today by many private companies and universities. The code generation field includes at least two major areas; (1) Formal specification languages which can be checked for accuracy and completeness – USE.IT from High Order Software is an examples of this type of code generator. (2) DARTS Technology from General Dynamics is an example of a code generator which is able to interpret the changes which must take place when an additional function is made to a given system. The code is then generated and an entirely new system is ready for use.

- Production libraries – Certain programming environments like Ada and Unix support a production library capability [1]. This allows all programs to be entered into a data base as they are generated. The details of their inputs, outputs and functions provided are maintained with the data base. This allows others to make use of the designs and programs that already exist. Reusing software in this way optimizes the coding process by taking advantage of previous design, code, and testing efforts.

The entire field of reusable software has mushroomed recently. The productivity increases which can be derived from the reuse of design, code, test, and all parts of the development is fundamental. Several programs and projects which will be contributing to this arena are:

Project	Company or Agency
SEF (Software Engineering Facility)	IBM/FSD
CAMPS	McDonnell Douglas
Reusable Software Implementation Program (RSIP)	Navy Research Laboratory
DARTS Technology	General Dynamics
DRACO	University of California at Irvine

3.5.2.1.3.3 Code Generation Projected Capabilities

All of the above options are being used today. The least mature is the automated generation of code from formal high level specifications. Several systems exist today and by 1987, it can be expected that others will be available and more target languages and machines will be supported.

Ada will also advance the production library technologies as it matures in the 1987 timeframe.

3.5.2.1.3.4 Code Generation References

[1] Michael Ryer, "Developing on Ada programming Support Environment", Mini-Micro Systems, Sept 1982, 223-226

3.5.2.1.4 Software Test, Integration and Verification

3.5.2.1.4.1 Software T, I & V Description

Testing is the examination of program execution behavior to ensure that requirements are satisfied. Testing of the Space Station software will be very important because of its life/mission critical nature. Facilities must be provided in the SSE for testing because of the inability to observe the software in actual use in a safe environment (i.e. on the ground). These test facilities must support a more cost effective approach to testing than has been achieved in past manned space programs. This description of the testing function will apply to the most rigorous testing. For less critical, less complex software less rigorous testing may be done.

Testing may be performed by the developer or by an independent agency. Independent verification and validation (IV&V) makes sense many times at the integration and systems level. IV & V is the procedure of evaluating the quality of the software and demonstrating that the functionality of the requirements have been satisfied by an autonomous agency. This agency should be involved from requirements specification time, and is independent of the development team and therefore, less affected by any biases. This agency looks at the system from the user's perspective.

Three levels of testing are described:

- Unit test
- Integration testing
- Systems testing

3.5.2.1.4.2 Software T, I & V Options Characterization

- Unit testing – The purpose of unit testing is to find errors in a single module of software. Errors found in this phase of testing can be corrected more cost effectively than in the other two phases.

Unit Test Tools - Tools that facilitate unit testing include:

Logic Flow Graph Generators - A tool of this type reads the source code for the program to be tested and generates a directed graph that represented the execution flow of the module (in some sense this is similar to a flow chart). One use of these digraphs is to analyze test coverage.

Data Flow Graph Generators - A tool of this type reads the source code for the program to be tested and generates a graph that would represent the flow of data through a program. Some uses of these graphs are to analyze set/use of variables and debug other variable usage errors.

Complexity Measuring tools - A tool of this type reads the source code for the program to be tested and generates a number that represents the complexity of sections of the code. This information is useful in identifying high risk areas of the code. These areas would require a higher test coverage.

Stub/Driver Generator - A tool of this type takes the source code of a module and generates skeleton stubs and drivers necessary for execution of the program being tested. This would need to be done, for example, when the module calls another procedure that has not yet been coded.

Test Data Generators - A tool of this type generates test data to be used to test the program. An example of how these test data values would be generated would be the following. The test data generator reads the source code and/or digraph and generates a set of input data that will force execution of the program to go down a certain logic flow path.

Source Level Debugger – A tool of this type provides diagnostic capability during test execution of the software. These capabilities would include start-stop at a given statement, display a variable's value and change a variable's value.

Test Design Language – This is a language to be used to design the test case procedures. It should be the interface to the source level debugger. This should be the same language that will be used for integration and system testing. To support the design and development of test case using this Test Design Language, a set of development tools very similar to program development tools should be provided. These tools include a syntax directed editor, static analysis aids, cross reference data generators, etc.

Performance Monitoring Tools – A tool of this type is used to estimate CPU and space usage of the module.

A Symbolic Execution Tool – A tool of this type symbolically executes the program being tested. This means that a simulation of the programs execution is performed with symbolic values placed in the program variables.

- Integration Testing – The purpose of integration testing is to find errors in the interfaces and communication between software modules. Integration and integration testing of Space Station software will be a more important aspect of the software engineering process than in previous manned space efforts. This is because of the distributed nature of the system as well as the extensive use of advanced technology hardware. The integration testing should proceed in small well defined steps. At each step the system execution should be less controlled and more realistic than the previous step.

Integration Test Tools – Several tools facilitate integration testing. These tools include:

Interface analyzers – A tool of this type reads the interface portion of the software being tested and either analyzes for potential problem areas or prints out a summary of interface information for manual analysis. This information would include cross reference data, calling sequences and parameter lists.

Integration Test Environment – This is a set of hardware and or software tools to provide diagnostics during execution of the software being integrated. This tool provides the same support that the source level debugger provides for unit testing. This tool may or may not be the same set of tools used as a source level debugger. If it is not the same set it may have pieces that are common with the source level debugger.

– System Test – The purpose of system testing is to take a completely integrated software system and execute it in an environment that is close to the real operating environment in order to find errors in the system. These errors may be code, design or requirements errors. This phase of testing includes the final acceptance testing of the system. This testing includes performance as well as functional testing.

System Test Tools – System testing tools are not so widely available as unit and integration test tools and most current system test tools are specific to the system being tested. The system test to be provided for testing of the Space Station software should be flexible enough to be used for system testing of other space systems and flexible enough to provide for technology insertion. The types of tools that should be provided are:

A System Test Environment - This tool provides the same function as the source level debugger and the integration test environment. The two significant differences between the system test environment and the other two tools are the following. The system test environment should provide a cost effective means of executing the entire software system with sufficient diagnostic capabilities. The system test environment should provide a means of executing the system in a way that is very close to the expected real operating environment.

Data Logging and Reduction Tools - A tool of this type saves data obtained from a system test and format, reduce and summarize the data. These type of tools may be used for unit and integration testing but they are most necessary in system testing because of the amount of data saved from a system test.

Data Analysis Tools - A tool of this type takes outputs from the data reduction tools and provide various kinds of automatic analysis. These include software programs that perform engineering calculations and compare the results with the data from the system test, software programs that correlate simulator data with data calculated by the system, software programs that format data into graphical form, graphics hardware and software packages, and expert systems that use rule base knowledge to analyze results.

Each of the above mentioned testing techniques uses one or all of the following types of test and test options which are included in the tools.

- Static Analysis
- Path Analysis
- Auditing
- Flow Analysis

Fortran Programming Environment (FPE) by SofTool, Inc., incorporates all the above testing options. In addition to these it also includes compilation, optimization, instrumentation, tracing, and guarantees thorough test coverage.

The Maintainability Analysis Tool (MAT) by Science Applications, Inc., also supports Fortran and incorporates the above options plus (1) locating module interface problems, (2) configuration management, (3) parsing and analysis of the source code, (4) produces calling trees, and (5) quantifies the maintainability of modules.

3.5.2.1.4.3 Software T, I & V Projected Capabilities

Software testing and verification along with integration is the most time intensive phase of software development. Developers agree that if the beginning phases can guarantee the accuracy of the requirements and design, this phase could be greatly reduced and the reliability of the software will increase. Therefore, because the automated tools to assist in the beginning phases of software development are becoming more robust, and testing will be incorporated into the development phase we will see these levels of testing accomplished in a smaller portion of the software life cycle.

3.5.2.1.4.4 Software T, I & V References

[1] Gerald M. Berns, "The Mat Program", Proceedings of the DECUS Symposium, Dec. 1984

[2] David Hamilton, "Space Station Integration Test Environment", IBM FSD Houston, 1985.

3.5.2.2.1 Acquisition Strategies

3.5.2.2.1.1 Acquisition Strategies Description

A combination of methods for acquiring software will be utilized on the Space Station because of the wide variety of systems required. Where possible, commercially available software products and software from previous NASA projects can be used when justified by lower life cycle cost projections. Otherwise new software may be developed under contract.

The following options are discussed:

- Commercially available
- Software recovery
- Multiple Contractor
- Single Contractor

Table 2 compares key characteristics of these options.

3.5.2.2.1.2 Acquisition Strategies Options Characterization

- Commercially Available - Commercially available off-the-shelf software (COTS) is the least expensive acquisition method. Total life cycle costs and schedule risks are much lower than other methods. Growth and technology advancements can be accommodated easier with commercially available software. See Table 3 for examples of some commercially available software.

  When acquiring software from the commercial world several issues must be addressed:

  - Performance and resource constraints
  - Changing requirements
  - Licensing the software
  - Use of the software in the final product
  - Multiple locations of the SSE

COMPARISON OF SOFTWARE ACQUISITION OPTIONS

FEATURE	COTS	RECOVERY	MULTIPLE CONTRACTOR	SINGLE CONTRACTOR
Relative Cost	LOW	MEDIUM	HIGHEST	HIGH
RISK	LOW	MEDIUM	HIGHEST	HIGH
ADVANTAGES	- Growth accommodation - Fastest development	- Fast development - Builds on itself	- Varied resources	- Fewer communication problems - Single NASA interface
Problems	- Changing requirements - Performance - Integration of commercial products	- Prejudices - Requirements mismatch - Logistics - Availability of maintenance skills	- Communication - Multiple customer interfaces - Less able to exploit commonalities	- Risk of limited resources - Possible narrow vision - Lack of diversity or expertise

Table 2

COMMERCIALLY AVAILABLE SOFTWARE

Type of Software	Example of Product
Operating Systems	VM – IBM MVS – IBM VAX/VMS – VAX NOS – CONTROL DATA MCP – BURROUGHS GCOS – HONEYWELL PC/DOS UNIX
Data Base Systems	DB II – IBM DMS II – Burroughs DM IV – Honeywell SQL/DS – IBM Accent R – DEC ADABASE – IBM AMBASE – DEC CLIO – IBM, DEC DBMS II – DEC, PDP DMS – SPERRY
Communication Systems	NDL – Burroughs CICS/VS – IBM ADR – IBM CMS 1100 – SPERRY Com–Plete – IBM DNS – HONEYWELL

Table 3

Software recovery - Related projects often have related software requirements. Ideally, software generated by one should be usable on another. In reality many factors make software recovery more difficult than it sounds. Many times the "not invented here" syndrome creates a negative atmosphere for software recovery. However, if it can be determined that an existing software package satisfies or nearly satisfies a new project's requirements, then it may be the cost effective approach. Costs and schedule risks will not be as low as a commercial product. [1] An example of a software product which has been successfully reused is the RTX real time operating system which was developed initially for NASA, but has been used on several DoD S/W development projects. [2]

- Multiple Contractors - If custom software must be developed to support unique requirements then NASA will accomplish this via contractors. In the Space Station program, there will be four NASA centers each with contractors performing software development functions for NASA. The multiple contractors approach is, therefore, not an option, but a given for the Space Station program. Interface and dependency problems must be addressed in this environment. Schedule risks are greater. Communication must be enhanced in order to maintain standards and to exploit commonality between elements.

- Single Contractor - A single contractor, whenever possible, holds the most promise for low cost, quality custom software. Interface and dependency problems are contained within a single management structure. Planning, scheduling and tracking also take place within that structure. A single interface for NASA exists. Standards are easier to define and enforce. Commonality is easier to exploit. Single contractors, however, run the risk of hitting many pitfalls such as narrow vision, lack of diversity and expertise, and the "not invented here" syndrome.

3.5.2.2.1.3 Acquisition Strategies Projected Capabilities

The key factor here is commercially available software and the single most important projected capability in the 1987 timeframe is the Ada programming language and its support environment. The Ada environment has the following advantages:

- Standard development tool set
- Portability of the software (tool sets and developed software)
- Future reusable packages library
- Designed for real-time embedded systems

The availability of Ada will have a significant impact on how NASA acquires and manages software. When Ada becomes commercially available the acquisition of its tools will limit the risk and cost of constructing the SSE itself. Ada then would be used to increase the quality of contractor generated custom software for the Space Station data management system.

3.5.2.2.1.4 Acquisition Strategies References

[1] R. C. McCain, Software Reusability Study Report, FSD Houston, 3/26/84

[2] George Gaxiola, "Commonality of Real Time Command and Control", IBM FSD Technical Directions, 1971, Vol. 7, No. 3.

3.5.2.2.2 Configuration Management

3.5.2.2.2.1 Configuration Management Description

Edward Bersoff [2] defines configuration management as "The discipline of identifying the configuration of a system at discrete points in time for purposes of systematically controlling changes to this configuration and maintaining the integrity and traceability of this configuration throughout the system life cycle". Figure 7 depicts configuration management as 4 elements. [3]

All large software development projects require a mechanism for controlling changes to the product. Historically the change process has evolved from a simple programmer controlled process at the beginning of the project to a sophisticated customer managed, data base controlled configuration management (CM) process at the maturation of the project. These CM systems have normally been developed along with the software product and were tailored to the environment present, i.e. control boards, change request forms, problem reporting forms, program library structures, etc.

Current trends in configuration management tools are similar to other tool areas - more general purpose tools commercially supplied and integrated with the other software development tools and data bases.

Configuration Management comes in several different 'flavors' and each type comprises minimum through maximum amount of control over the software.

- Support Software - the management of developed or acquired software which is used in the SSE to develop additional software. In particular, this can include; compilers, editors, linkers, testers, etc.

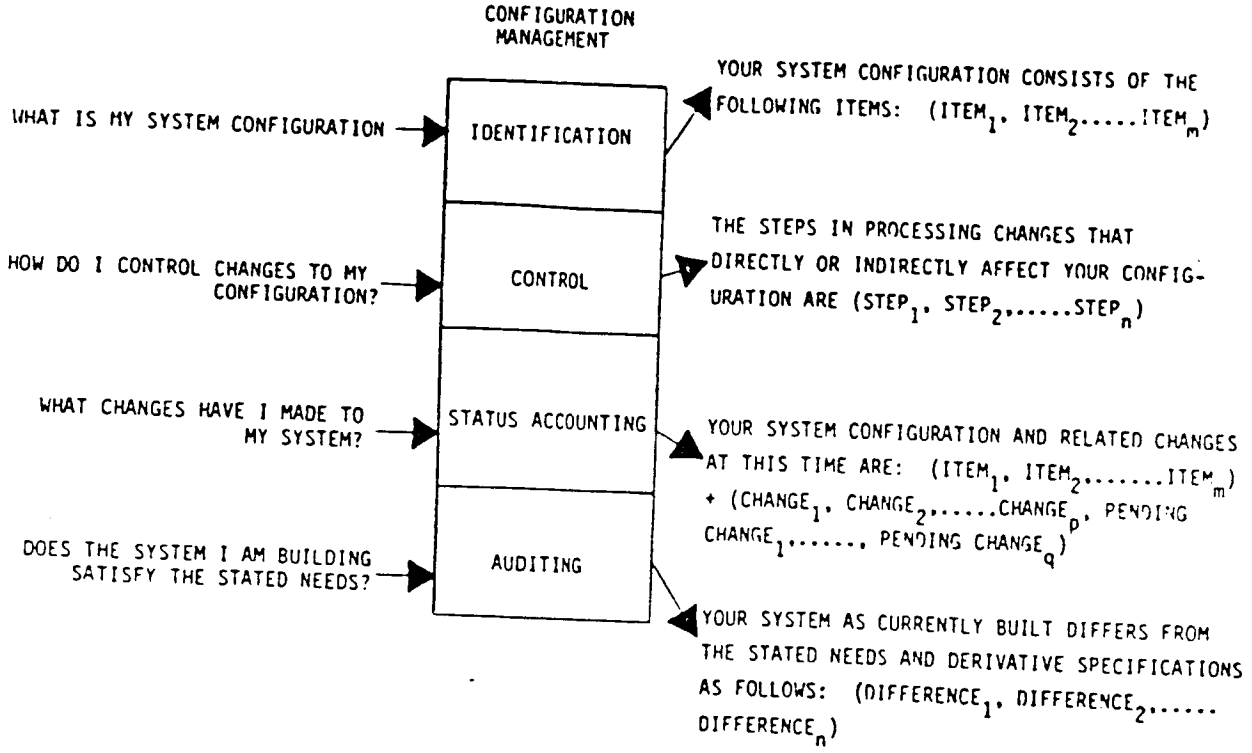


Figure 7. The Four Component Elements of Configuration Management

- Product Software - the management of software developed or acquired for the Space Station Program activity.

- Documentation - the management and control of all documentation generated, or delivered for all software.

Controls over the software and documentation will and should take on different levels at different locations of this program.

A minimum amount of control is desired over Space Station customers that generate autonomous software (i.e. the programs do not interface with other Space Station DMS subsystems and the DMS is not dependent on the programs). More controls, such as integrated testing and means of tracking problems and changes, will be needed if their software is not autonomous. The most control will be exercised over core and system level software packages. Also to be considered are aspects of configuration management as they apply to whatever onboard support of changes is eventually provided in the Space Station DMS. Location of control will be a factor in Space Station software configuration management. With four NASA centers contracting software development, the CM process must be tailorable to some extent to each center's special requirements.

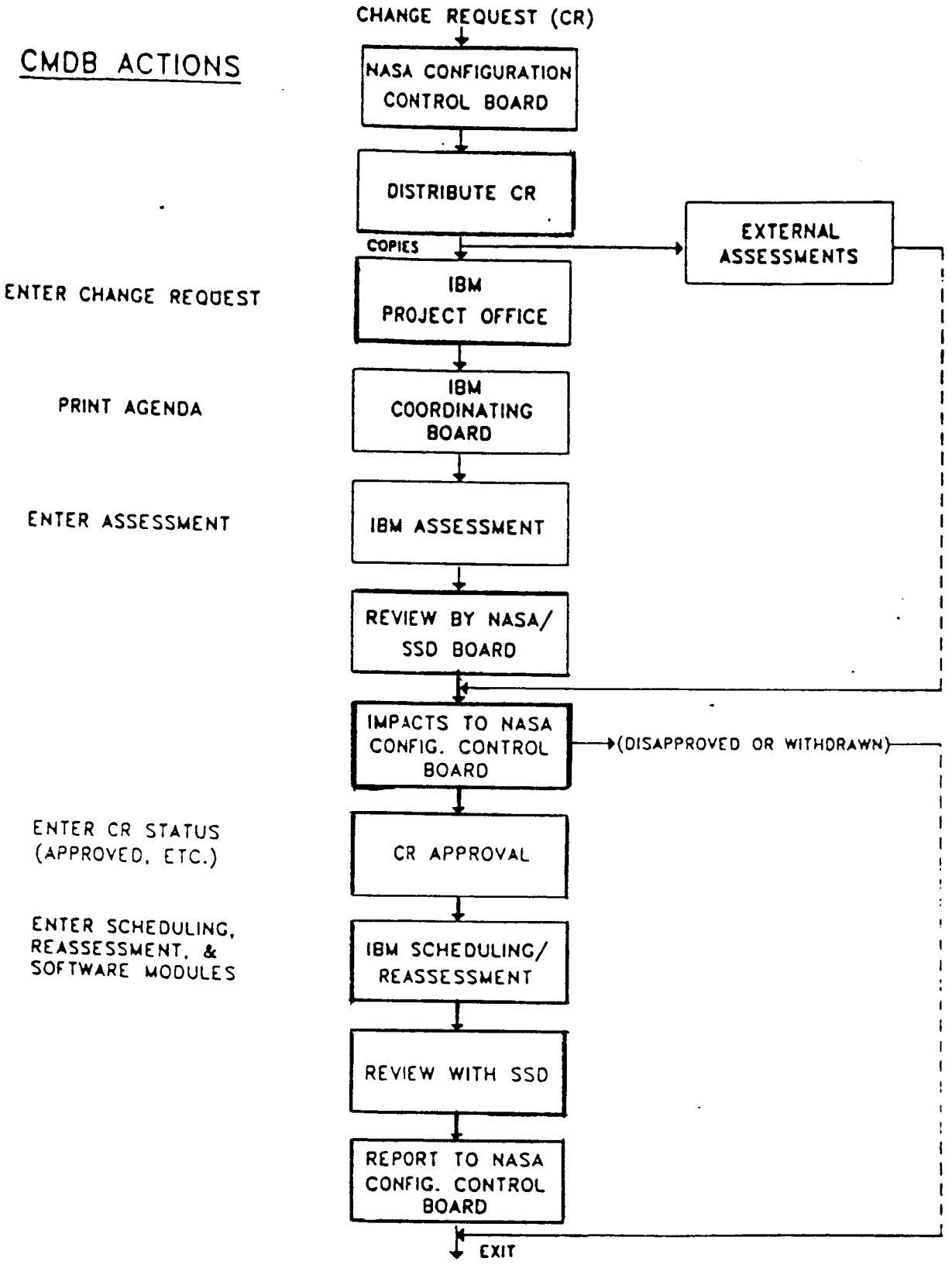
The following options are discussed:

- SPF
- MMS
- CCC
- Source Tools

3.5.2.2.2.2 Configuration Management Options Characterizations

- SPF - The Software Production Facility is the tool used to produce the Primary Avionics Software System (PASS) for the Space Shuttle. It is comprised of commercial and custom hardware and software products. Configuration control is maintained over the commercial software, the custom development tool software and PASS software.

  Configuration control of software produced within the SPF (both the SPF and the PASS) is effected by an extensive set of tools, which are integrated into a package to support the planning, development, build and test phases of a software project. The elements which are under configuration control are the source modules, the executable load modules, and descriptor data for each source module (e.g. library residence, language translators, linkage editor module type, cross reference data, and program function). All changes must be associated with one or more 'Control Instruments'. There are many different types of control instruments (Change Request (CR), Discrepancy Reports (DR), Program Change Authorization (PCA), etc.). All control instruments are stored in an IMS data base called the Configuration Management Data Base (CMDB). A set of control boards is associated with each control instrument and the control instrument must receive approval from each of its boards before the affected software is baselined into the system. See Figure 8.



CHANGE REQUEST PROCESSING FLOW

Figure 8

Interactive panels are provided using the Application Development Facility (ADF) to allow the users to enter the control instruments, board approvals, affected modules, and module submission. Access to the panels is controlled through profiles, so that only authorized users can enter board approvals, schedules, etc. Verification that approval has been given is done automatically by the build tools. If an update has been submitted for a control instrument that is not approved, it will be automatically withdrawn from the build.

A large number of printed reports are available from the CMDB and several products exist to facilitate additional report generation from the IMS data base.

A change in the commercial software must be preceded by the creation of a Change Request (CR) or an Incidence Report (IR). These are stored in an information management data base. CRs are normally generated by the user community and must be approved by the Change Control Board staffed by the customer and users prior to implementation. Incidence Reports are created as the result of a problem being reported to the central 'Help Desk' which is staffed by the organization responsible for the commercial products. The software modifications are provided by the commercial vendors. These are incorporated onto a set of system disk packs which are then frozen, tested and then released into production.

- Module Management System (MMS) is a product of Digital Equipment Corporation which supports the configuration of a software library system. This includes the changes of versions for different site locations, rebuild capabilities, change tracking, and avoids redundancies in the database. The tool uses flat files and runs on VAX hardware.

- Change and Configuration Control (CCC) is a product of Softool which provides a comprehensive change control environment for a software library system. CCC controls who can make changes, handles source code, object code, test data, and documentation and can deal with any

language. The programs comprehensive features include automatic reconstruction of previous versions, problem tracking, difference reports, management reports, access control, archiving, compression, encryption, and automatic recovery.

- SourceTools is a product of Oregon Software which supports the development and maintenance of software program libraries. It consists of a collection of programs which function with any computer language, coordinates changes made by several programmers jointly, and uses standards text files and documentation. The tool set consists of (1) control, creation and modification of source files, (2) a mechanism to build an entire system, (3) a maintenance program.

3.5.2.2.2.3 Configuration Management Projected Capabilities

The future of CM systems appears to be general purpose tailorable systems working on data in a distributed data base (distributed to the extent of the development environment). These systems are available today and should be run as a background process to track the software changes and documentation. The metrics for evaluating and assessing configuration issues will also be included by 1987.

3.5.2.2.2.4 Configuration Management References

[1] System Engineering Tools Compendium - IBM FSD Bethesda, MD 1983

[2] Edward Bersoff, "Software Configuration Management", Prentice Hall, Englewood Cliffs, NJ 07632, 1980

[3] Daniel Roy, "Software Tools and Methodology Study for NASA MSOCC", Century Computing, Laurel, MD 20707, June 1984

3.5.2.2.3 Standards Definition and Enforcement

3.5.2.2.3.1 Standards Definition and Enforcement Descriptions

A common SSE will provide NASA with the best chance for defining and enforcing standards. Definition and enforcement are very different issues and must be approached using distinct processes. Standards, while necessary for communication and uniform implementation, should not lead to restricting productivity. Therefore, care needs to be taken defining and enforcing standards only where history has proven the need.

Standards should be defined for many different aspects of software development. It is first necessary, however, for an analysis of the software development process to be done to decide where standards show the most promise of helping to attain program goals. Once determined, standards must be documented. This can be accomplished by publishing the most rigid standards in a formal standards document which is carefully change controlled. These standards specify design and code methodologies to be followed, review and testing processes, documentation required, and compliance and deviation aspects. Less rigid standards and conventions can be documented and distributed in a more dynamic manner.

Standards for software requirements and design documentation are often a function of criticality of the software. Critical software systems may be required to maintain documentation through the detailed design specifications for the life of the project.

Automated standards enforcement will truly improve productivity in software development standards, since both the software and documentation can be updated at the same time, eliminating the need for dual changes.

A unique problem that the Space Station DMS system will have to address is the definition and enforcement of standards to apply to the customer supplied software that will execute in the SSDS resources. NASA has recently published a directory of 25 standard documents and guidelines from a variety of source: IEEE, National Bureau of Standards, DoD and ESA. [5]

The following options are discussed:

- Control boards
- Inspections
- NMI 2410.6
- ICD's

3.5.2.2.3.2 Standards Definition & Enforcement Options Characterization

- Control boards - Control boards are a means of assigning a group of "concerned" individuals the responsibility of reviewing submitted material to ensure it meets the standards applied by the control board. For this method to work the control boards must have authority, knowledge, and time to do the job. Submission must be a requirement and submitters must view the board as informed and fair. A process for review, pass or fail, and tentative resubmission must be set up.

- Inspections - Inspections address standards enforcement via peer review. Any level of software engineering (i.e., requirements, design, code, etc.) has products which may be reviewed for conformance to applicable standards.

  Inspections are a proven manner to catch errors and insure standards conformance since their introduction in 1973 by IBM [3]. Inspections have taken place in varying degrees of formality. Their success has been recognized and inspections are being used more often. [4]

  Standards enforcement is only one reason for inspections. Inspecting for product correctness, performance concerns, efficiencies, etc., are other reasons for inspections [1].

- NMI2410.6 - NMI2410.6 is a NASA document which specifies standards to be applied to software development for mission critical systems developed for NASA. [2]. NM12410.6 specifies a plan that contains a management approach section and a technical approach section. Within the management approach section should be information about the following:

    1) Identification of all S/W elements and corresponding organizational responsibilities
    2) Approach to categorization and classification policies
    3) Management mechanisms in the areas of: requirements development and control, schedules development and control, resource development and control, internal review concepts, and external review concepts
    4) What documents will be generated
    5) What CM techniques will be used
    6) What quality assurance plans will be made
    7) Deviation and waiver procedures
    8) Method of maintaining the management plan

Within the technical approach section should be information about the following:

1) Top level functional requirements
2) Hierarchical view of system elements
3) Plan for software requirements definition process
4) Plan for software design & implementation process
5) Plan for test and delivery process
6) Plan for maintenance & updating process
7) Plan for software engineering approach

Finally NM12410.6 provides for specific reviews of the project management plan.

- ICD's - Interface Control Documents (ICD's) are documents that specify the details of hardware and software interfaces in an exact manner. All affected parties have to agree on the details and thereafter the interface is fixed and becomes a standard which must be adhered to by all users.

3.5.2.2.3.3 Standards Definition & Enforcement Projected Capabilities

The Standards Definition and Enforcement process will be aided by the automation of the software development process. As programmers, designers and testers begin to utilize the paperless electronic tools being developed, several of the problems with standards will be lessened. Communication among inspection items, for example, will be facilitated by the SSE and inspections can be done on line with comments and corrections being fed back automatically to the programmer. With electronic documentation, all reviews by control boards for standards compliance will be made more effective. Interface control standards will be easier to track and verify, especially when the interfaces are between centers where long distance communication is involved in the definition and compliance process.

Means to aid these communication problems in standards definition and enforcement are available today and should be incorporated into the SSE.

One tool which enforces standards is the Fortran Programming Environment (FPE) by SofTool, Inc. FPE guarantees that all source code meets required standards by auditing the code, incorporating user defined standard formats, and includes PDL statements as developed during the detailed design phase. This last feature supports the inspection team concept, so that evaluation of the code can be accomplished by comparing it to the accepted detailed design.

3.5.2.2.3.4 Standards Definition & Enforcement References

[1] Weinberg, Gerald M. and Freedman, Daniel P., "Reviews, Walkthroughs and Inspections," IEEE Transactions on Software Engineering, Vol. SE-10, No. 1, 68-72 (1/1984)

[2] NASA NS/Information Systems Divisions, NMI2410.6, NASA Software Management Requirements for Flight Projects, Feb. 1, 1979

[3] IBM, "Structured Walkthroughs: A Project Management Tool," Bethesda, Maryland, 1973

[4] Edward Yourdon, Structured Walkthroughs, Yourdon Press, New York, NY 1979.

[5] Directory of Software Standards, NASA Office of Chief Engineer, Software Management and Assurance Program, D-DI-D9, January 1985.

3.5.2.2.4 Development Environment Structure

3.5.2.2.4.1 Development Environment Structure Description

Software productivity increases can be mapped to several actual items; (1) use of automated tools, (2) computer facilities and their availability, (3) techniques and methods for development, and the most influencing factor, (4) work environment.

3.5.2.2.4.2 Development Environment Structure Options Characterization

A TRW study headed by Barry Boehm [1] and by the Atlantic Systems Guild, Inc., headed by Tom DeMarco [2] have demonstrated the necessity to evaluate the work environment and ergonomics of the software development group. Areas which must be considered and evaluated are:

- Private work space - recommend 100 square feet
- Furnishings - desk, table, chairs
- Modular vs enclosed rooms
- Communications - electronic mail, telephone
- Terminals, workstations, personal computers, and printers
- Support personnel - secretarial staff, computer operations, etc.
- Support areas - conference rooms, lunch areas, technical libraries

- Programmer team structure – The team structure for the programming group has a significant effect on the efficiency of the group. One option is the chief programmer team which emphasizes a strong technical leader in the programming group. Requirements analysis and top level design is under control of the lead. Programmers in the group produce the detailed design and then implement and test. The Chief programmer coordinates, controls, and enforces standards on the group. The group acts as a team participating in peer group reviews (inspections), testing each others implementation, and exchanging implementation techniques.

3.5.2.2.4.3 Development Environment Structure Projected Capabilities

Since the development environment holds such an important role for software productivity, each of these areas must be studied and the best solution for each should be incorporated into the SSE as part of the entire concept. Powerful new software engineering tools are beginning to emerge and should be encorporated into the SSE as they are available. Other issues, such as quiet time to concentrate, training for tools and techniques, and support groups (in-house gurus), should be considered when dealing with the work environment.

The encouragement of group dynamics, communication techniques, and feelings of ownership must be taken into consideration when setting up a development environment.

3.5.2.2.4.4 Development Environment Structure References

[1] Barry Boehm, Software Engineering Economics, Englewood Cliffs, N.J., Prentice-Hall (1981).

[2] Tom DeMarco, "The 1984 Coding Wars", presented at the Structured Development Forum VI, February 1985, Atlantic Systems Guild, New York, NY.

3.5.2.2.5 Development Facilities

3.5.2.2.5.1 Development Facilities Description

The facility for software development is normally a "host" processor(s). Here the system is built for a target machine where it will ultimately reside. The various types of host facilities all present different perspectives to users. The type of facility may affect the integration functions which occur in the SSE. The means for handling growth of the SSE will be different. It is, therefore, important to critically review the options for SSE facilities. Three options for the distribution of SSE host processing are discussed:

- Centralized
- Distributed with unique hardware environments
- Distributed with common hardware environments

Table 3 summarizes the characteristics of the facilities options.

3.5.2.2.5 Development Facilities Options Characterizations

- Centralized - A centralized SSE implies one large host facility with local and long distance workstation usage. The central host has computing power in one location sufficient to handle peak loads of all users plus any integrated testing/simulation. Long distance access is handled via long distance telephone lines or DOMSAT links.

  A centralized facility implies that the whole set of growth and resource management functions are controlled by a single agency in the most efficient manner. System procurement is facilitated under one agency. Hardware incompatibilities are minimized. A single physical plant exists - one building, A/C, operators etc.

FEATURE	CENTRALIZED	COMMON DISTRIBUTED	UNIQUE DISTRIBUTED
RISK	MEDIUM	LOW	HIGH
GROWTH POTENTIAL	MEDIUM	HIGH	HIGH
H/W COSTS	NO	DIFFERENCES	SIGNIFICANT
INITIAL S/W COSTS	LOW	MEDIUM	HIGH
LIFE CYCLE S/W COSTS	LOW	LOW	HIGH
USER SUPPORT	- Perceived lack of control - Limited site unique uses - No hands on	- More direct user control - Effective support personnel - Allows hands on use	- Limited tool set - Less effective support personnel - Allows hands on use
INTEGRATION SUPPORT	- Best - Good support for reviews and communication	- Good	- Poor - Incompatible systems
DATA MANAGEMENT	- Best support	- Problems: - communications of data - security - dependencies	- Problems: - communication of data - security - dependencies

Table 3 – Comparison of Facilities Options

- In use, a centralized SSE facilitates the speed, ease, and effectiveness of all integrating functions such as intercenter reviews, standards enforcement, interface control, system builds, configuration management, high level planning, scheduling and the TMIS interface. Also, a centralized facility allows for more exploitation of hardware and software commonality. Data bases are more easily integrated together and all users have the same data available on-line and up to date.

- Distributed with unique hardware environments - This option would allow each center to use whatever processing facilities it had or desired to procure. Tools or methodologies which the center and its contractors had found most effective could be carried forward into the Space Station program. The SSE tools would have to be designed and coded to run in all the various facilities. Commonality of hardware and software factors would not be achievable with the resultant savings lost. SSE development would be a much larger challenge with increased cost and schedule risks.

  An advantage of a distributed with unique hardware environment SSE is that each facility would have maximum control over the SSE resource. The least expensive system for that center could be procured. And it could be more effectively tailored to the job assigned to that center.

- Distributed with common hardware environments - A distributed SSE implies a network of host processors integrated together by a controlling center. Each host site has the control necessary to manage the computing resources it feels are needed to handle its user requirements. Access to the distributed sites is via local or long distance lines or domestic satellite links. A lead center provides the overall integrating functions by gathering data from all hosts on the network.

This environment gives the host site control over the resources necessary to accomplish its job with the exception of the SSE system itself which would be supplied by the lead center. All center unique requirements of the SSE system would also be provided or approved by the lead center. This would allow for a maximum exploitation of commonality resulting in this lowest cost full function SSE.

With each center responsible for procuring its own system, a chance exists for making use of existing facilities – within compatibility constraints of the SSE system. Center unique software might be recoverable.

Growth is easily accommodated in a distributed designed SSE. Larger host processors or new host sites can handle increased work loads and network data rates can be increased without significant impacts to the SSE system.

In use, a distributed SSE requires each site to provide the lead center with all data necessary for integrating functions such as intercenter reviews, standards enforcement, interface control, system builds, configuration management, high level planning, scheduling, and the TMIS interface.

Some functions in a distributed SSE might appear slower to the users or have outdated data. Certain intercenter data bases would have to be kept at the lead center to reduce the problems associated with data duplication. This would require a network request to the lead center for data from these data bases.

3.5.2.2.5.3 Development Facilities Projected Capabilities

All capabilities discussed for both distributed and centralized facilities exist today. The future will see greater host processor capabilities for less cost and greater network communication capabilities and standards. One technology sure to play an important role in the future will be intelligent workstations. Whether the SSE is distributed or centralized the intelligent workstation should be considered in the design of the SSE. As these workstations evolve more power and memory they will be able to take over much of the jobs typical SSE users will be repeatedly doing-edits, compiles, and low level testing.

3.5.2.2.5.4 Development Facilities References

[1] Tannenbaum, Andrew S., Computer Networks, Englewood Cliffs, N.J., Prentice-Hall, Inc. 1981

3.5.2.2.6 Project Management Tools

3.5.2.2.6.1 Project Management Tools Description

Project management implies the management of a project using techniques to lay out the plans, evaluate the progress, and provide the "what-ifs" for a project. All aspects of a project are taken into account; resources, tasking, costing, milestones, burden rates, and tracking the actuals against the estimates. The most ideal situation is to have an automated tool which will query the project in someway and automatically generate the reports necessary. This automation should be accomplished without the intervention of a third party, and should be capable of generating accurate reports based upon the needs of the project manager or supervision.

3.5.2.2.6.2 Project Management Options Characterization

All of the project management tools currently available on the market run on a variety of hardware and were developed specifically for business systems. The software development world has been basically ignored as far as project management is concerned. Currently, there is only one (known to this author) available for tracking the project exactly as it progresses — APCE an automated tool developed by PRC, Inc. The unique aspect of this tool is its ability to track the progress of a project without the interference of the management or a dedicated person to track the project. It queries the project data base for expected vs. accomplished, evaluates the current status of a project, charts the critical path, and estimates futures based upon the current status. This is all done by using the typical software project life-cycle from requirements through software maintenance and evaluating the project as it progresses in time.

3.5.2.2.6.3 Project Management Projected Capabilities

Software development has become a major effort in project management and will continue to become more and more important in all future efforts which are automated. As demonstrated in some of the semi-integrated tools available on the market, software project management, along with project metrics, is an extremely necessary and important part of delivering software on-time and within-cost. Therefore, within the next three to five years the addition of software project management tools will see a major increase in their availability.

3.5.2.2.6.4 Project Management Tools Reference

None

Acronym List

AIE	Ada Integrated Environment
ALS	Ada Language System
ALS/N	Ada Language System/Navy
APSE	Ada Programming Support Environment
CCC	Change and Configuration Control
CM	Configuration Management
CMDB	Configuration Management Data Base
COTS	Commercial Off-the-Shelf Software
CR	Change Request
DCDS	Distributed Computer Design System
DFD	Data Flow Diagram
DMS	Data Management System
DR	Discrepancy Report
E/R	Entity/Relationship
HOS	Higher Order Systems
ICD	Interface Control Document
IV&V	Independent Validation and Verification
KAPSE	Kernal APSE
LSE	Language Sensity Editors
MAPSE	Minimum APSE
MMS	Module Management System
PASS	Primary Avionics Shuttle Software
PCA	Program Change Authorization
PSA	Problem Statement Analyzer
PSL	Problem Statement Language
RFP	Request for Proposal
RTX	Real Time Executive
SSE	Software Support Environment
SPF	Software Production Facility
SREM	Software Requirements Engineering Methodology
SA	Structured Analysis
SSDS	Space Station Data Systems
STARS	Software Technology for Adaptable and Reliable Software
SYSREM	System Requirements Engineering Methodology
TMIS	Technical Management Information System

3.5.3 System Test, Integration and Verification

Successful acquisition and operation of any large project is critically dependent on its Test and Verification plan. This plan provides the test sequences and requirements for all levels of hardware and software development/integration to insure the operational effectiveness and suitability of SSDS ground and flight systems. Test and verification on flight hardware of the typical space program is generally extremely conservative and intensive; however, the Space Station program has the opportunity to follow a less conservative and more cost-effective approach. This does not imply a lack of rigor to insure initial achievement of the operational system; but concentrated implementation of standards, commonality, and fault tolerance techniques coupled with the on-orbit accessibility of the flight system will allow consideration of some significant schedule and cost savings options.

Within the traditional "high reliability" program, acceptance tests, (including mission derived environmental and "burn-in" screens) are liberally applied to all hardware from piece parts through assembly (black box) level in order to expose faulty components, processing and workmanship. Special test equipment including mounting fixtures, cabling, etc., is required to support these various levels of tests along with a set of comprehensive functional and environmental test software. Ground hardware is generally subjected to less comprehensive testing without environments because of its accessibility and its benign handling/operation environment. Software products are subjected to comprehensive "verification and validation" testing that begins at the module level and progresses through module integration and system level tests. Hardware qualification (acceptance profiles plus margin) testing is performed on initial production flight assembly samples to insure hardiness of the production hardware. These qualification units are generally dispositioned as "non-flight" since their exposures represent 100% of the assembly demonstrated fatigue life and because such hardware is useful in supporting software development, special testing, etc.

Verification is that effort (test, demonstration or analysis) performed to insure that the system, with all its hardware and software components; complies with its design requirements. Whenever feasible, hardware verification is distributed down to the subassembly level to provide early exposure of deficiencies and minimize cost and schedule associated with their corrective action. Similarly, the software verification is initiated at the module level. Verification activities that do not coincide with acceptance testing are necessary only on the initial hardware/software set.

The final activity is that of system(s) integration which is performed (1) to insure compatibility between all subsystem and system hardware and software interfaces, including payload interfaces, and (2) to complete the verification of the system design and performance requirements. (Note that this document is generic in the sense that it does not address the onboard Space Station Data System (SSDS) interface and interaction with any specific subsystem or payload. All tests, integrations, verifications, and interfaces are intended to apply to all functions of all subsystems and payloads.)

Review of potential test and evaluation scenarios for the SSDS ground segment indicates that its acquisition phase will follow traditional flows therefore no options will be discussed. In contrast, the Space Station represents a departure from the typical space project in that:

- the Station will be assembled, activated and upgraded on orbit,
- the Station is less sensitive to subsystem total weight, thus mechanical strengths of enclosures, etc., can be designed to meet the mechanical environments,
- the Station has no critical, real-time operations required during ascent, re-entry, etc., but only in the remote event of life threatening or major damage situations, thus a higher level of faults can be tolerated,
- the Station will be manned so that hardware replacement, repair, and reconfiguration will be a primary capability,
- the Station will have the capability to modify/upgrade software on-orbit,

- o the Station will have several different levels of criticality in both hardware and software, thereby allowing for the possibility of some verification to be completed on-orbit.

These factors stimulate a number of options to the normal acquisition and growth phase for the onboard SSDS that can result in meaningful schedule and cost savings.

3.5.3.1 Test Options

3.5.3.1.1 Description

There are three areas in the testing (pre-system integration) sequence where deviations from the traditional test sequence offer significant cost savings with manageable risk. The first option addresses the elimination of selected testing at the lower hardware levels based on test sequences of subsequent levels. The second addresses the elimination of selected environmental tests based on their marginal effectiveness and the third addresses the modification of qualification testing to accommodate protoflighting.

Specifically, these identified options are:

- o Deferred Module Testing
- o Selective Environmental Testing
- o Modified Qualification Testing

3.5.3.1.2 Options Characterization

Deferred Module Testing

The typical waterfall hardware test and evaluation sequence performs acceptance testing at each level of hardware. Modules (Circuit Card Assemblies, etc.) are subjected to comprehensive functional and environmental

screens to expose production problems and move the hardware further along the reliability curve. Considerable effort, in time, special test equipment, adapters and software are required to support this effort, with attendant costs, yet subsequent failures are not precluded.

With the limited production quantities of the Space Station, the costs (particularly non-recurring) of these tests will be significant. A viable option would be to defer the module (card level) testing until the next level (subassembly or assembly). Additional inspections plus utilization of high pedigree (rescreened?) components will minimize any added risk. A coherent, hierarchial built-in-test (BIT) scheme is anticipated for the SSDS hardware/software products. In such a scheme, higher level assemblies will monitor responses of the lower level modules to the normal or background diagnostic stimuli. Anomalous data will be captured as required and the fault incidents will be flagged to the next higher assembly level. The BIT capabilities will be verified early in the integration sequence using module simulation in order to gain the required confidence in their operation. The BIT will then become a verification tool at all subsequent levels of integration/verification providing adequate checkout of the modules.

The advantage of this option would be the elimination of special module fixturing and software development costs plus the individual test time; the disadvantage would be the potential of failures at the subassembly and assembly level which would net out to the disassembly and reassembly time required for the module repair/replacement time.

Selective Environmental Testing

As indicated previously, the typical "high rel" test sequence performs thermal environment (temperature-altitude and/or temperature-cycling) testing at several levels of hardware build-up. The justification is the accelerated exposure of component, material or workmanship faults that would normally occur during operational life. A viable option would be to perform thermal environmental testing only on the qualification unit to verify the hardware material and fabrication processing tolerance to thermal environments. The random failures of other production units would be acceptable within the framework of the Station fault tolerance requirements.

The advantages of this approach would be reduced testing, documentation and manpower; the disadvantage would be the reduced opportunity to expose hardware faults prior to utilization.

Modified Qualification Testing

Hardware qualification testing is generally performed on the first production units of an assembly (black box) as representative samples of the components, materials and fabrication processes. This testing demonstrates capability envelopes to establish a high confidence that, with a typical distribution of component and workmanship quality, subsequent production units will meet acceptance and mission requirements. These "qual" units are normally dispositioned as "non-flight" since they have expended 100% of the demonstrated environmental fatigue life and they have been subjected to the exceptional handling required for the qualification sequence, mechanical, thermal environments, EMC, etc. With the relatively small production runs on some of the SSP equipment, however, non-flight hardware may represent a large percentage of the production run costs.

The modified qualification test option would retain the flight status of this protoflight hardware by subjecting it to less than the traditional qualification levels/durations such that confidence in "the fleet" is established yet the risk of failure during subsequent operations is low. For example, the +6 dB qualification levels could be performed, but only for acceptance test durations. Also, the fact that the Space Station is not subjected to the usual external environments of ascent, re-entry and landing, mitigates the normal qualification levels. The various launch packages must, of course, be boosted to the assembly orbit however the hardware packages will generally not be operational during the ascent and can therefore be specially packaged/handled to significantly reduce the launch environments. The risk may be reduced further by applying "over-design" factors and also by performing additional confidence activities (open box inspections, etc.) following the qualification sequence. In some cases, the "over-design" factor may be sufficient such that qualification can be by analysis rather than testing.

The advantage to this approach is the reduction in production hardware cost. The disadvantages are (1) normal full confidence in production hardware could not be achieved by demonstration, (2) capability to assess production unit fatigue life would be limited, and (3) no production workhorse unit would be available for use in troubleshooting, special tests or mockups.

The issue of protoflighting/non-flight units must be carefully evaluated for each equipment type with appropriate weighting applied for:

the Local Area Networks (LAN) could greatly facilitate this distributed effort. Final integration of the onboard SSDS is therefore dependent of the integration activity of the overall Space Station.

o the degree of commonality achieved (the number of unique hardware configurations,

and,

o the hard requirements for non-flight elements.

NOTE: Software testing is covered in the Software Development Plan.

3.5.3.2 Integration Options

3.5.3.2.1 Description

It is anticipated that the hardware and software products for the onboard SSDS (excluding application software for payloads) will be the responsibility of NASA or a single support contractor (or contracting team). These products will be procured (built or bought), tested and integrated into the individual assemblies (black boxes) of the onboard SSDS. In this development effort, the contractor will utilize the SSP developed test beds and other simulation and test facilities in order to perform the interface compatibility and performance verification of the individual assemblies.

In addressing the integration options for the onboard SSDS, it must be acknowledged that this is not a stand-along system; it will be installed into elements (modules and structures) of the overall Station that will be developed in geographically separate locations. The nature of the designs of the Normal Operation System (NOS), the Data Base Management System (DBMS), and the Local Area Networks (LAN) could greatly facititate this distributed effort Final integration of the onboard SSDS is therefore dependent of the integration activity of the overall Space Station.

The Station integration testing will be the single most expensive operation of the test and evaluation sequence because of the complexities of set-up, hardware and software problems on both the SSDS and supporting test equipment. Difficulties will arise from sheer magnitude of the system, the total coordination required, and the need to simultaneously satisfy all the facility and test equipment requirements. The basic options therefore relate to the degree of integration and the location of the integration/verification effort.

The options associated with the integraiton effort are:

- o full integration at a central facility
- o a segmented, serial integration effort
- o on orbit integration

3.5.3.2.2 Options Characterization

Full Integration at a Central Facility

This options addresses the traditional approach of performing the overall integration and verification at a central facility. The approach would have all SSP elements, payloads, modules, station structural, and assemblies shipped to the central facility for full IOC station integration performed within the limits of the 1g environment. This test would also utilize special TDRS system configurations/operations to provide a communication interface between the assembled station and TDRS.

This central facility could be located at a contractor's plant or at any one of several NASA installations. The major advantages and disadvantages of

full integration at a central facility are basically the same whether at a contractors or at a NASA facility. Of major importance is the geographical location of the facility, considering material transportation costs, personnel resources required for TDY support, and facilities availability.

This approach provides the primary advantages of:

- o a comprehensive verification of the station in its assembled configuration
- o a common point of coordination for problems involving multiple contractors
- o a high degree of pre-launch confidence in the IOC station
- o an opportunity for the crew to experience the integrated capabilities of the actual IOC system
- o a tool to support on-orbit operations, modifications, troubleshooting, etc.

The disadvantages are:

- o high planning/schedule cost
- o the test is difficult to perform, inefficient to manage and therefore costly
- o new facilities will be required
- o expensive special handling/cradling equipment will be required
- o large support contracts for personnel and equipment required
- o transient on-site NASA and support contractor staffing will be required
- o test failures could result in excessive delays
- o all elements may not be available at the same time

Segmented, Serial Integration

As noted in the previous option, large integration efforts tend to be inefficient and therefore expensive due to the requirements to simultaneously manload and provide facility/equipment support for all involved elements. A viable option would be to perform a series of smaller integration efforts. There appears to be sufficient segmentation and available physical interfaces within the Station such that the integration could be performed in a more piece-meal fashion; i.e., interconnection and checkout of only one to two

modules with its onboard SSDS equipment at a time. Systems simulation will be available in sufficient fidelity to support these smaller integration packages through substitution of missing elements/interfaces. The requirement to intermate and verify the interfaces of each station element would be satisfied by a staggered sequence of the segmented integration.

This option has inherently more schedule flexibility since only a fraction of the integration effort is affected if one SS element fails. The acquisition phase may in fact be designed to take advantage of the Station "build-up" sequence; i.e., elements for the final launch package may not be ready for integration when the early packages are being assembled on orbit.

Another consideration for the segmented integration is the required interface compatibility and performance support of institutional resources, TDRSS and the Data Distribution Network, etc. These resources will have a limited availability for these tests and must be scheduled in advance. These constraints would be more easily accommodated within the segmented format proposed by this option.

Full assembly of the SSP elements (modules and structure) will not be performed in this approach, however through the use of CAD techniques and designed in mechanical flexibilities, the risks levels should be acceptable.

The advantages of this option would be:

- o greater flexibility and less schedule impact from detected incompatibilities
- o less handling and fixturing equipment requirements
- o small work force requirements
- o shorter on-site durations for support personnel

The disadvantages are:

- o possibly more simulation software required
- o higher risk for on-orbit success

On-Orbit Integration

For this option, outfitting/integration of the Station modules would be followed only by a module checkout during ground testing. This checkout, supported by the Space Station system simulation would verify only the general module capabilities. Performance testing, will have been demonstrated at the subsystem level utilizing SS simulation and test facilities. Module/structure interfaces will be mated to insure physical compatibility followed by brief operation tests to verify functionality of power, thermal, fluid, network and logical interfaces. Following module testing, the Station elements will be assembled into their respective launch packages for pre-launch integration and checkout. Full station integration/verification will occur on orbit.

The advantages of this approach are:

- o reduction of integration facilities, fixtures, etc.
- o reduction of integation manpower, documentation, schedule
- o provides a more serial operation that is compatible with the buildup sequence

The disadvantages of this approach are:

- o higher risk during on-orbit integration
- o first integrated end-to-end performance checks on orbit
- o most severe penalty for integration detected incompatibilities or deficiencies (includes wait period until next STS flight)
- o no crew exposure to actual system operation on ground

3.5.3.3 System Verification Options (IOC)

3.5.3.3.1 Description

In the traditional system, hardware and software verification is a distributed effort, performed as early as possible during system development such that there is a high confidence when the final prelaunch integration/ verification is performed. Consistent with this conservatism is the

independent verification/validation of critical system software. The incentive to substantially reduce acquisition phase costs of the Space Station leads to options to:

- o minimize system level verification
- o eliminate independent software verification/validation

3.5.3.3.2 Options Characterization

Minimized System Level Verification

It is anticipated, if not given, that the development environment for the Space Station will be rich in simulation and development tools including the end-to-end models and test bed capabilities such that each contractor can develop and comprehensively verify hardware and software at the subsystem level. Detailed verifications of hardware signal characteristics (signal levels, frequencies, etc.), BIT and fault-tolerance operation would be performed as predefined in an overall system test and verification plan. When completed, each verification item would be mapped onto appropriate data bases within the TMIS to provide current visibility on the system verification status. Within this option, only residual performance verification would be performed during the final integration tests. This residual testing would be limited to overall functional capabilities with their timing requirements to validate the accumulation of tolerances allocated to the individual subsystems. Detailed subsystem hardware functional and fault tolerance/ redundancy testing other than power up and BIT diagnostics would not be performed. Neither would "programmed failure" tests be performed to demonstrate system operability/recoverability to hardware/software discrepancies.

The advantage of this option, would be the obvious scope reduction of the integration testing, while the disadvantage would be the lack of repeated detailed testing to insure that there have been no performance shifts as a result of transportation damage, or the integrated environment. Also, there would be no opportunity for contractors/crew to experience system level responses to induced failures.

Elimination of Independent Software Verification/Validation

Within the typical large space project, outside agencies have been commissioned to review software specifications and perform an independent verification/validation of the generated code. This activity is intended to provide an objective review and test of the software without the designer "biases". This option proposes to eliminate the independent review. Instead, the normal sequence would be followed wherein software designers would fully verify individual modules and integrated programs then pass the programs on for system validation. The software would support the system integration process in either case, however the risk of redesign, recoding operations and residual (post integration) errors may be higher. The corrective action for such errors could be significant if PROM components must be reworked in addition to the recording, and recompiling activity.

The advantage of this option is strictly cost; the disadvantage is the increased risk of design/coding errors detected at the system level.

3.5.3.4 Integration Options (Growth)

3.5.3.4.1 Description

The growth phase presents an additional complexity and potential risk since the actual station hardware will not be accessible. The available options for the growth phase relate to the degree of risk mitigation activities performed prior to the final on-orbit integration. The growth hardware/software products will be contractor supplied to NASA with full design/performance and interface compatibility verification completed. The options range from a comprehensive pseudo-integration in a high fidelity system mock-up/simulator and/or use of the TDRSS link to the Space Station to a very limited checkout that only demonstrates interface compatibility utilizing crude mockups, fixtures and available simulation facilities. These options may depend somewhat on the magnitude, complexity, criticality of the integration effort. A small set of non-critical assemblies and/or software packages, for example, could be installed, integrated and checked out serially on the station, with little concern for impact to system operation. A more critical hardware/software package, or critical payload, may dictate additional ground "integration" testing.

These options are:

- o intensive ground integration
- o minimal ground integration

3.5.3.4.2 Options Characterization

Intensive Ground "Integration"

In this option, the hardware and software products will be fully checked out using a high fidelity mockup/simulator, a replication of the station built essentially from production hardware, or use of the TDRSS link to the Space Station. This approach, of course, relies on the availability of these facilities/tools (see section 3.5.3.6).

The testing sequence would install the hardware in the mockup, perform its power up, and BITE diagnostics, load its applicable software (as requied) then perform appropriate set of system diagnostics to verify the physical attributes and performance of the growth products. The advantages would be a high confidence in the effectiveness and compatibility of the products and a high capability for evaluation team (ground and orbital crew) to gain operational experience. The disadvantages are primarily the cost/schedule impacts of developing and maintaining the high fidelity simulation and the additional effort to perform the extensive ground checkout.

Minimal Ground Pre-Integration

In this option, the products are merely (re-)checked for interface compatibility utilizing a minimal set of test fixtures, crude mockups, and the resources of the Software Support Environment to validate the software. The advantages of this approach are the basic economics of the sparse checkout equipment requirements and the reduction of the ground activities. The disadvantages are the higher risks of successful integration.

3.5.3.5 Integration Options (Man-tended)

3.5.3.5.1 Description

The man-tended option is assumed to be a 3 to 5 year interval that delays the continuous manned presence in the IOC Space Station. The buildup of the man-tended reference configuration could use the same initial launch sequence as that proposed for the manned Space Station, with two exceptions: in Flight 3 the man-tended system would have laboratory module delivered rather than a habitat module, and it would have one airlock rather than two. Flight 4 would deliver the externally mounted payloads and supporting structure for payload mounting. The man-tended configuration would then be operational.

The man-tended option will have many similarities, in both hardware and software, to the manned IOC reference configuration. However, further definition is likely to identify some required differences, such as an expanded use of Automation and Robotics (A&R) for both Payload support and Space Station housekeeping. Also, the fact that crew members will not be continuously present means that less time will be available for the checkout of those non-critical functions that may have been intentially deferred from the pre-launch checkout.

The need for intensified ground controlled functions will have an effect on the SSIS, primarily in the amount of manpower required at the ground station.

The man-tended option definition must also include the planning for the transition to a permanently manned configuration.

3.5.3.5.2 Options Characterization

The integration options for the man-tended configuration are not dissimilar from those for the manned reference configuration discussed in Section 3.5.3.2. The on-orbit integration option discussed previously may have

an added disadvantage of a longer elapsed time to complete the on-orbit integration depending upon the parameters of NSTS crew size, length of stay time, and the frequency of revisits.

The man-tended version presents an additional option concerning Payload accommodations, in that the crew support capability will be seriously curtailed.

- o Forego the accommodation of those payloads requiring crew intensive support
- o Increase the use of expert system technology and A&R to service those payloads, with the attendant increase in hardware and software complexity and integration resources.

3.5.3.6 Facilities Options

Large projects associated with flight hardware generally beget large facilities for the purposes of H/W development, S/W development, System Simulation, Systems Integration, and Training that are distributed across the sites of the prime contractors, the project control center and the launch centers. Although these facilities and associated equipment are comprehensively planned to support the project in an efficient and cohesive manner, final configurations generally surpass expected costs and functions are often duplicated (not replicated) or overlapped in several locations. Such facilities have served their purposes well but at high cost.

A set of Space Station facilities must serve the same functions, however, with the projected minimal funding, more cost effective approaches must be employed.

3.5.3.6.1 Software Support Environment

A software support environment (SSE) system is a given that must provide the normal development and analysis tools, with supporting data base and configuration management functions, etc. In addition, the system software

simulation will be housed by the SSE. Clearly, there will be simultaneous demands by a large number of users and the facilities options must address whether this system will be a centralized facility with local and telecommunications access or whether a distributed or hybrid system would be more serviceable and cost effective. These issues have already been comprehensively discussed within the "Software Devel-option" options category and will not be reiterated here.

3.5.3.6.2 Hardware Development

3.5.3.6.2.1 Description

Hardware development of the Station subsystems will be a concurrent effort at many separate contractor locations. The development must be supported by a number of tools and capabilities to allow contractors to perform early verification of interface and performance compatibility with the overall system. These tools could be distributed (individualized) in the form of fixtures and adapters supported remotely by the SSE or a centralized, relatively high fidelity system (or hybrid) could be provided. The options are therefore:

- o Distributed capabilities
- o Centralized capabilities

3.5.3.6.2.2 Options Characterization

Distributed Capabilities

As the preliminary designs are developed, contractors will need to evaluate their products in terms of compatibility and performance and in some cases will need to develop access and routing options for their equipment. Interface evaluations including mating, signal level, timing and protocol compatibility could be supported by interface fixtures with adapters remotely driven from the systems simulation capability of the SSE. This capability

assumes the availability of a remote work station(s) or MPAC's at the contractor site to schedule and control the SSE configuration. This full capability work station/MPAC would also be used to: interface the TMIS, access the SSP data bases to update program documentation and status, etc. The fit checks, routing and access evaluations would be performed on mockups that provide dimensional accuracy only for access, supports and mounting surfaces.

The advantages of this distributed approach would be the relative independence of the contractor (except perhaps the scheduling conflicts for the system simulator) and the immediate accessibility of the SS support capabilities without travel or transportation requirements. The disadvantages would be the quantities and control of fixtures, adapters and mockups to be used by the contractors. Each item must be under configuration control, thus system changes could be reflected into a continuous flow of rework and modifications at the contractor or NASA site.

Centralized Capabilities

For this approach a centralized hardware development facility would be provided that would not have the fidelity/capability of the SAIL for example but will be sufficient to allow installation of hardware products for ground integration/evaluation. Relatively high fidelity mockup(s) of the Space Station would be provided that are semi-operational in that appropriate fixtures would be pre-integrated to allow remote support by the SSE system simulator. The mockups will be segmented such that several independent activities could be simultaneously supported. This facility, most likely located at the NASA SE&I center, would be used to evaluate all intercontractor interfaces in addition to allowing each contractor to perform the normal fit checks. This capability will be supplemented by the NASA test beds available to test and evaluate DMS concepts, fault tolerance concepts, etc.

The mockups would also support preliminary mission definition and some crew training.

The advantages of this approach would be the tighter configuration management of the mockups and a higher visibility of potential incompatibilities between contractors. The disadvantages would be the travel and transportation requirements for each contractor.

3.5.3.6.3 Test Beds

Test beds represent those NASA facilities that are or will be used to explore or validate design concepts. They will be available to contractors in support of any advanced development, however, it is not clear that they will play a role in the downstream development either during acquisition or the growth phases. To do so would detract from their role in advanced development and would require them to fall under the purvue of external configuration management and quality control functions. Their value may be somewhat limited to a specific design production when the facilities planned for the Space Station should accommodate any assembly or subsystem integration/verification.

3.5.3.6.4 Integration Facilities

Integration Facilities are discussed in Section 3.5.3.2.

3.5.3.6.5 Training

Flight crew training will involve the following areas:

- Station operation and maintenance
- payload operational support
- Station habitation

A large part of this training for each of these areas will involve utilization of the fixed and portable MPAC's which will be available in quantities in all facilities. For training purposes, these MPAC's will be driven by the SSE to simulate the tasks of station and payload support in both normal and "failure" operational modes.

Another segment of training will be devoted to maintenance, repairs, and removal/replacement of equipment from structure and modules. This training effort could be accomplished via the detailed mockups of the Hardware Development Facilities in conjunction with equipment supplied by contractors. Separate training facilities for the DMS oriented functions could be provided; however, it is anticipated that the commonality between space development and test capabilities and the actual system capabilities will minimize the need for such special facilities.

Training for payload operations, zero-g environments, etc., that are not associated with the DMS are not applicable to this paper.

Tables 1 through 6 provide summaries of the options discussed above along with a tabulation of their advantages/disadvantages.

Table 1. TEST OPTIONS

OPTION	DEFERRED MODULE TESTING	SELECTIVE ENVIRONMENTAL TESTING	MODIFIED QUALIFICATION TESTING
ADVANTAGES	1. ELIMINATION OF SPECIAL MODULE FIXTURES. 2. REDUCTION OF S/W DEVELOPMENT COSTS. 3. REDUCTION OF TEST TIME.	1. REDUCED: - TEST TIME - DOCUMENTATION - MANPOWER	1. REDUCTION IN PRODUCTION HARDWARE COST.
DISADVANTAGES	1. POTENTIAL OF FAILURE AT SUB AND ASSEMBLY LEVEL. 2. RE-ASSEMBLY AND RE-TEST TIME IN EVENT OF FAILURE	1. REDUCED OPPORTUNITY TO EXPOSE HARDWARE FAULTS.	1. NORMAL FULL CONFIDENCE IN PRODUCTION HARDWARE NOT DEMONSTRATED. 2. LIMITED ABILITY TO ASSESS PRODUCTION UNIT FATIGUE LIFE. 3. NO PRODUCTION UNIT WORKHORSE (HANGER QUEEN) AVAILABLE.

Table 2. INTEGRATION OPTIONS

OPTION	FULL INTEGRATION AT A CENTRAL FACILITY (CONTRACTOR OR NASA)	SEGMENTED SERIAL INTEGRATION	ON-ORBIT INTEGRATION
ADVANTAGES	1. COMPLETE VERIFICATION OF SS IN ASSEMBLED MODE. 2. COMMON POINT OF COORDINATION FOR MULTIPLE CONTRACTORS. 3. HIGH DEGREE OF PRE-LAUNCH CONFIDENCE. 4. CREW EXPERIENCE ON THE INTEGRATED SS. IF AT NASA-KSC: 5. EXISTING FACILITIES PARTIALLY UTILIZED. 6. NASA STAFFING PARTIALLY IN PLACE. 7. REDUCED TRANSPORTATION REQUIREMENTS TO LAUNCH SITE.	1. GREATER FLEXIBILITY, LESS IMPACT OF DETECTED PROBLEMS. 2. LESS HANDLING EQUIPMENT REQUIRED. 3. SMALLER WORK FORCE REQUIRED. 4. SHORT ON-SITE DURATION FOR SUPPORT PERSONNEL.	1. REDUCTION OF INTEGRATION FIXTURE AND FACILITIES 2. REDUCTION OF INTEGRATION MANPOWER, DOCUMENTATION, SCHEDULE. 3. A MORE SERIAL OPERATION - COMPATIBLE WITH BUILD-UP SEQUENCE.
DISADVANTAGES	1. DIFFICULT TO PERFORM, INEFFICIENT TO MANAGE, COSTLY. 2. ALL ELEMENTS MAY NOT BE AVAILABLE AT THE SAME TIME. 3. SPECIAL HANDLING EQUIPMENT REQUIRE. 4. LARGE SUPPORT CONTRACTS REQUIRED. 5. FAILURES RESULT IN EXCESSIVE DELAYS.	1. POSSIBLY MORE SIMULATION SOFTWARE REQUIRED. 2. HIGHER RISK OF ON-ORBIT SUCCESS.	1. FIRST END-TO-END PERFORMANCE CHECKS ON-ORBIT. 2. HIGHER RISK DURING ON-ORBIT INTEGRATION. 3. MOST SEVERE PENALTY FOR DETECTED PROBLEMS. 4. NO CREW EXPOSURE TO SYSTEM ON THE GROUND.

Table 3. SYSTEM VERIFICATION OPTION

OPTION	MINIMIZE SYSTEM LEVEL VERIFICATION	ELIMINATE INDEPENDENT SOFTWARE VERIFICATION
ADVANTAGE	1. OVERALL REDUCTION IN SCOPE OF INTEGRATION TESTING.	1. COST SAVINGS.
DISADVANTAGE	1. LACK OF REPEATED DETAILED TESTING TO UNCOVER PERFORMANCE SHIFTS. 2. REDUCED OPPORTUNITY FOR CREW TO EXPERIENCE SYSTEM LEVEL RESPONSE TO INDUCED FAILURES.	1. INCREASED RISK OF DESIGN/CODING ERRORS DETECTED AT THE SYSTEM LEVEL.

Table 4. INTEGRATION OPTIONS [GROWTH]

OPTION	INTENSIVE GROUND INTEGRATION	MINIMAL GROUND INTEGRATION
ADVANTAGES	1. HIGH CONFIDENCE IN THE EFFECTIVENESS AND COMPATIBILITY OF THE PRODUCTS. 2. OPERATIONAL EXPERIENCE FOR THE GROUND AND ORBITAL CREW.	1. RELATIVE LOW COST OF THE GROUND CHECKOUT EQUIPMENT. 2. REDUCED GROUND CHECKOUT TIME.
DISADVANTAGES	1. COST AND SCHEDULE IMPACT OF DEVELOPING AND MAINTAINING A HIGH FIDELITY SIMULATION. 2. EXTENSIVE GROUND CHECKOUT TIME.	1. HIGHER RISK OF SUCCESSFUL INTEGRATION.

Table 5. INTEGRATION OPTIONS [MAN-TENDED]

OPTION	LIMIT CREW INTENSIVE PAYLOAD ACCOMMODATIONS	INCREASED USE OF EXPERT SYSTEMS TECHNOLOGY, AUTOMATION, AND ROBOTICS FOR PAYLOAD ACCOMMODATIONS
ADVANTAGES	1. LESS COMPLEX SYSTEM FOR CHECKOUT AND VERIFICATION. 2. LOWER COST SYSTEM.	1. MOST PAYLOADS CAN BE ACCOMMODATED IN THE MAN-TENDED MODE.
DISADVANTAGES	1. CERTAIN PAYLOADS CANNOT BE ACCOMMODATED UNTIL SS IS FULLY MANNED.	1. MORE COMPLEX SYSTEM FOR CHECKOUT AND VERIFICATION. 2. HIGHER COST SYSTEM.

Table 6. FACILITIES OPTIONS

OPTION	DISTRIBUTED CAPABILITIES	CENTRALIZED CAPABILITIES
ADVANTAGES	1. INDEPENDENT CONTRACTOR OPERATION. 2. IMMEDIATE ACCESS TO ELEMENT SUPPORT CAPABILITIES. 3. MINIMIZED TRAVEL AND TRANSPORTATION REQUIREMENTS.	1. EASIER CONFIGURATION CONTROL MANAGEMENT PROBLEM. 2. HIGHER VISIBILITY OF POTENTIAL INCOMPATIBILITIES BETWEEN CONTRACTORS AND BETWEEN ASSEMBLIES.
DISADVANTAGES	1. LARGE CONFIGURATION CONTROL EFFORT FOR FIXTURES, ADAPTERS, MOCK UPS, ETC.	1. LARGE TRAVEL AND TRANSPORTATION REQUIREMENTS FOR EACH CONTRACTOR.