NASA TM- 87647

NASA Technical Memorandum 87647 NASA-TM-87647 19860015266

A COMPUTER MODELING METHODOLOGY AND TOOL FOR ASSESSING DESIGN CONCEPTS FOR THE SPACE STATION DATA MANAGEMENT SYSTEM

William R. Jones

April 1986

NASA
National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665

ABSTRACT

A computer modeling tool is being developed to assess candidate designs for the Space Station Data Management System (DMS). The DMS is to be a complex distributed computer system including the processors, storage devices, local area networks, and software that will support all processing functions onboard the Space Station. The modeling tool will allow a candidate design for the DMS, or for other subsystems that use the DMS, to be evaluated in terms of performance, reliability, cost, power consumption, weight, and other trade parameters. The tool and its associated modeling methodology are intended for use by DMS and subsystem designers to perform tradeoff analyses between design concepts using varied architectures and technologies.

N86-24737 #

ACKNOWLEDGMENTS

The modeling methodology and tool described in this paper were defined and specified by David Bahrs, Edward Luczak, and Dr. Antonio Vallone of Computer Sciences Corporation, Silver Spring, Maryland. Lawrence F. Rowell, of NASA LaRC, provided support in the guidance of this effort and in the review of this paper.

TABLE OF CONTENTS

SECTION 1 - INTRODUCTION

The conceptual design process for the Earth-orbiting Space Station (SS) requires the systematic assessment of architecture and technology options for all station subsystems before selecting the most promising candidates for development. A key element of the Space Station is the Data Management System (DMS), which will provide the command, control, data processing, and coordination functions for all subsystems within the station. The DMS architecture, hardware, and software alternatives selected must be consistent with mission objectives and estimates of technology readiness, and must satisfy DMS system requirements. To aid in the selections, digital computer models can be used to represent candidate DMS designs and provide assessments of performance, reliability, and cost. Such models are applied during the design and development phases of the Space Station for the rapid evaluation of design and technology options and architectural configurations. In addition, such models improve station evolution by exposing future DMS technology needs and permitting cost and performance assessments of proposed DMS enhancements.

This paper describes a modeling methodology and an associated computer program tool that is being developed to enable software models of alternative Space Station DMS design concepts to be rapidly built and evaluated. The methodology is called the "SS DMS Assessment Methodology", and the tool is termed the "SS DMS Assessment Model." The DMS models built using the SS DMS Assessment Methodology and Model can be exercised to allow assessments of performance, reliability, cost, weight, power consumption, and other trade parameters. This paper describes the rationale, concepts and logic used in developing this computer-aided assessment modeling methodolody and tool. It also provides a high-level introduction to the modeling methodology and to the architecture and design of the modeling tool. The discussion should be of general interest to Space Station managers and engineers, and of special interest to DMS designers. New computer systems in other applications such as space platforms and ground support systems will likely have the same type of distributed architecture as the SS DMS. Designers of these systems should also find this paper of interest. Those who anticipate using this tool can find more detailed information in references 2, 3, and 4.

Section 2 of this paper briefly describes the SS and the DMS. Section 3 summarizes the requirements for the modeling tool and the modeling needs of its intended users. Section 4 describes the overall architecture of the modeling tool. Section 5 describes the modeling methodology and how the modeler uses the tool. Section 6 provides some details about the design of the tool, including the model elements supported, inputs, algorithms used, and types of output generated. Section 7 illustrates the use of the modeling methodology by showing several of the initial steps taken in modeling a candidate design for a component of the

Space Station Guidance and Control Subsystem. Section 8 provides some concluding remarks about the anticipated use of this modeling tool and methodology.

SECTION 2 - SPACE STATION DATA MANAGEMENT SYSTEM

This section provides a brief overview of the problem domain that has provided the motivation for the development of the modeling capability described in this paper. The following paragraphs describe the Space Station and its onboard DMS.

As currently conceived, the Space Station Program will consist of a base station, plus related equipment and platforms in co-orbiting and polar orbits. The base station, operating in a 28.5 degree orbit, is planned for continuous manned operation, but will also be capable of supporting periods of unmanned operation. Its initial configuration is planned to consist of two habitation modules, two laboratory modules, and one logistics (supply) module. An Orbital Maneuvering Vehicle (OMV) will aid in movement of supplies and equipment outside the base station environment. Future plans include the facilities for maintaining and operating Orbital Transfer Vehicles (OTVs) from the station. The laboratory modules will be used for scientific investigations, materials and pharmaceuticals experiments, space manufacturing, and other activities by station customers in pursuit of the commercialization of space. Some of the tasks identified require special conditions, such as microgravity and isolation from station contaminants, vibration, and shock. These requirements could dictate the need for separate platforms in the same or other orbits. Each of these station elements will be managed by its own Data Management System (DMS), which is used to support both the operation of the element and the management of its communications.

The physical arrangement of structural elements used as a reference by NASA at the time this work began is known as the "power tower" configuration, shown in figure 1. The physical arrangement of the station is important to consider during model development because of signal path lengths, system redundancy requirements, and reliability issues with which the modeling tool must deal in performing a complete DMS assessment. Detailed analyses of Space Station configuration requirements by NASA and its contractors have uncovered deficiencies in the power tower design and has led to the recent adoption of a new "dual keel" reference configuration, shown in figure 2. Because the SS physical configuration is reflected only indirectly in user-provided parameters, the modeling methodology and tool described in this paper will be fully applicable to the new dual-keel configuration.

The DMS consists of the set of standard onboard processors, storage units, local area networks, workstations, equipment interfaces, and software that collectively support the monitoring and control of all core and payload equipment and data functions onboard the SS. Other SS subsystems will use the support services provided by the DMS. In this paper, the term DMS is used often in a general sense to include hardware, system software, and subsystem application software that will use DMS services. Figure 3 shows the "Reference Configuration" (see ref. 1) for the DMS that has been established by NASA as a departure

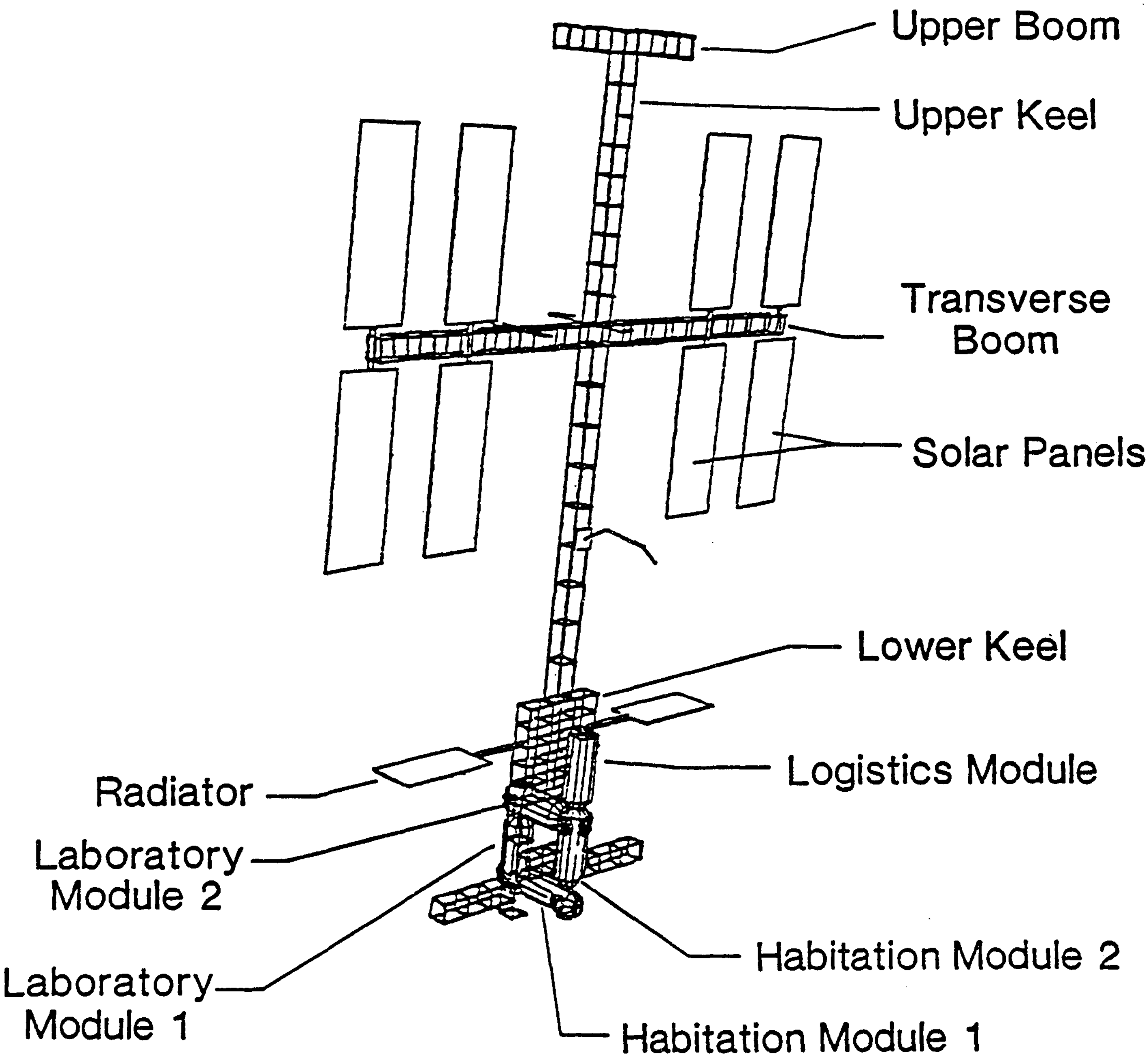

Figure 1. Representative Space Station Configuration.

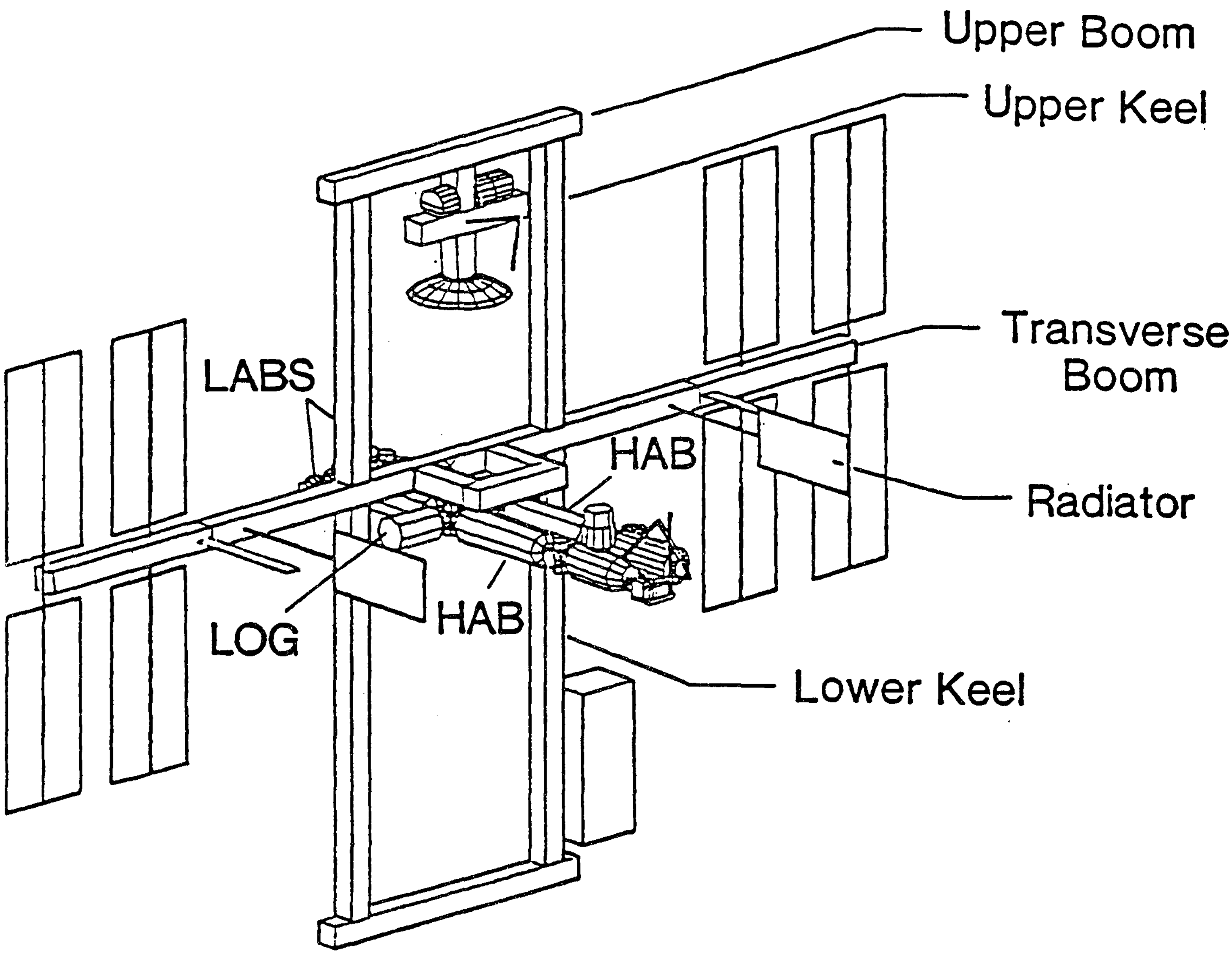

Figure 2. Dual Keel Space Station Configuration.

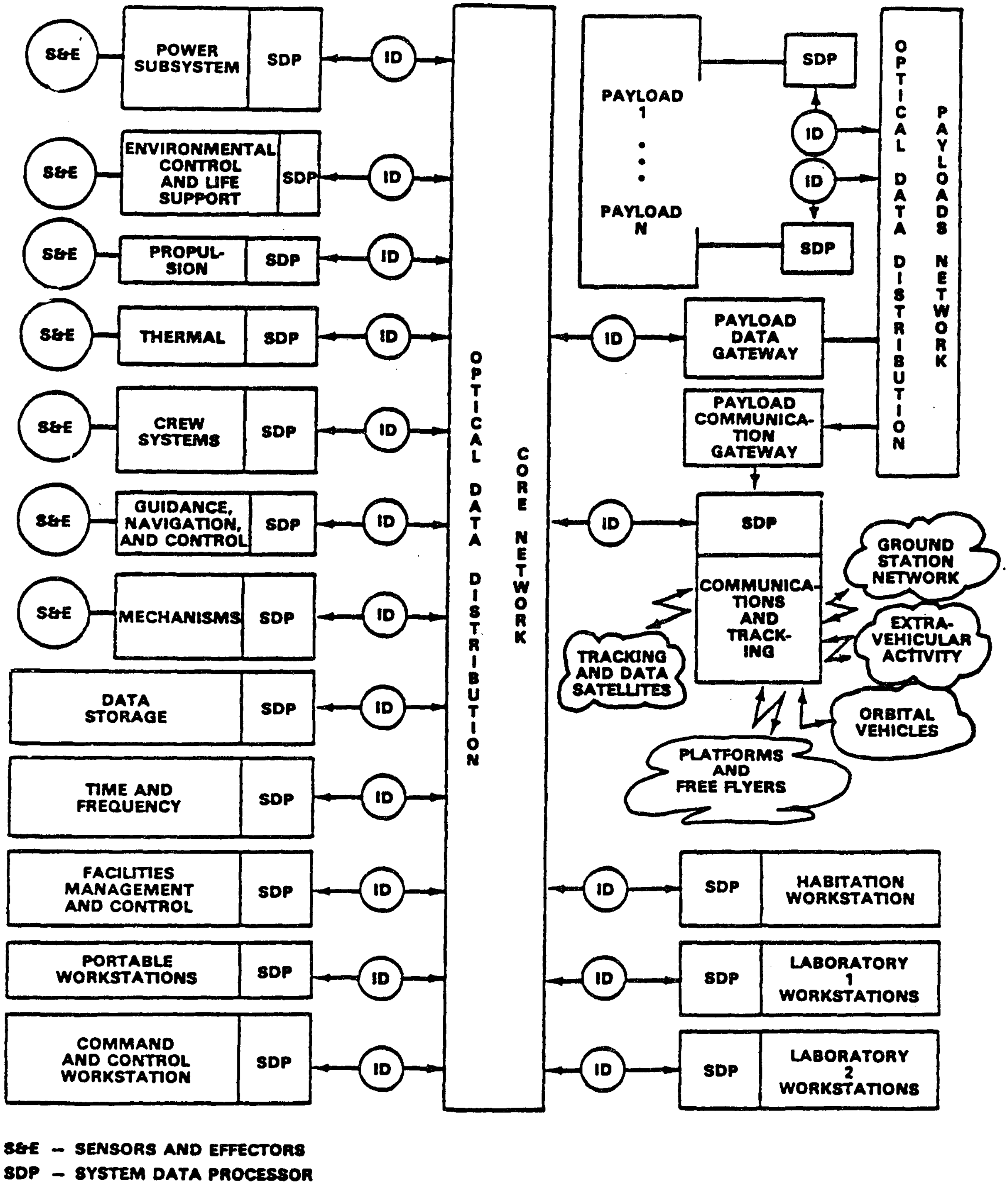

Figure 3. SS DMS Reference Configuration.

point for further definition and preliminary design. The figure shows the broad range of functions performed by the SS DMS.

Figure 4 shows a representative DMS core network layout, in which ring-based local area network components have been distributed throughout the SS modules and on the major structural members to connect processors, sensors, and effectors. Figure 5 shows a representative DMS system layout with a candidate design of the types, numbers, and distribution of equipment items that will comprise the DMS, including Subsystem Data Processors (SDP), Network Interface Units (NIU), local busses, and subsystem sensors and effectors. This figure also shows the planned relationships of the core network (supporting basic station control) and the payload network (supporting customer experiment control) and their physical locations within the station. From this representative physical layout of required equipment, model designers formulated the concepts for a computer-aided modeling tool to make engineering assessments about this complex data management system.

The Space Station DMS has the critical job of orchestrating the functioning of all onboard systems and of interfacing with the station crew. Its architecture must be flexible, adaptable, and highly reliable because the system must resist obsolesence over a continuing lifecycle. It must perform flawlessly with or without support from the crew and the ground. It must be capable of recognizing and reporting malfunctions and failures of all

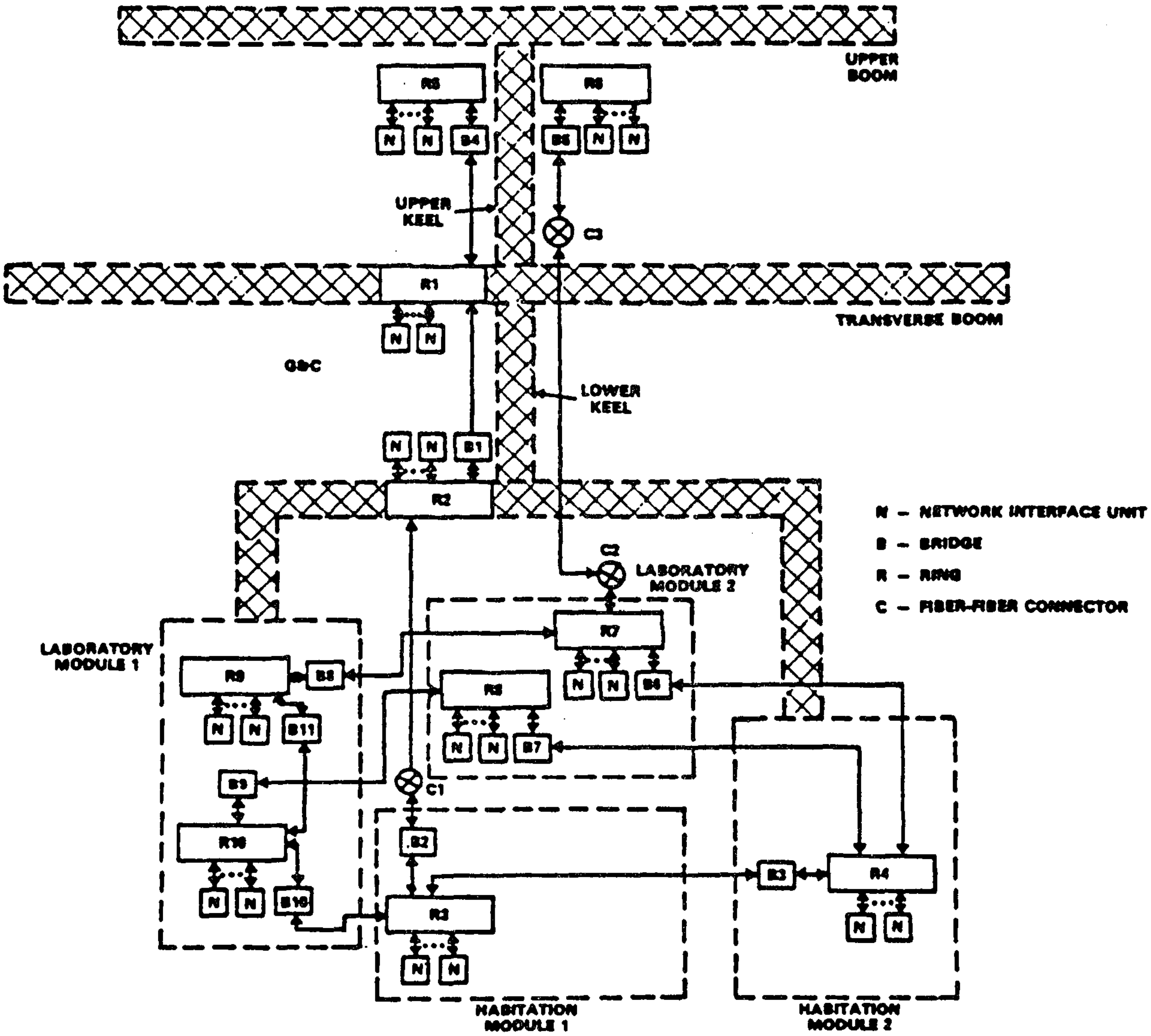

Figure 4. SS DMS Core Network Layout.

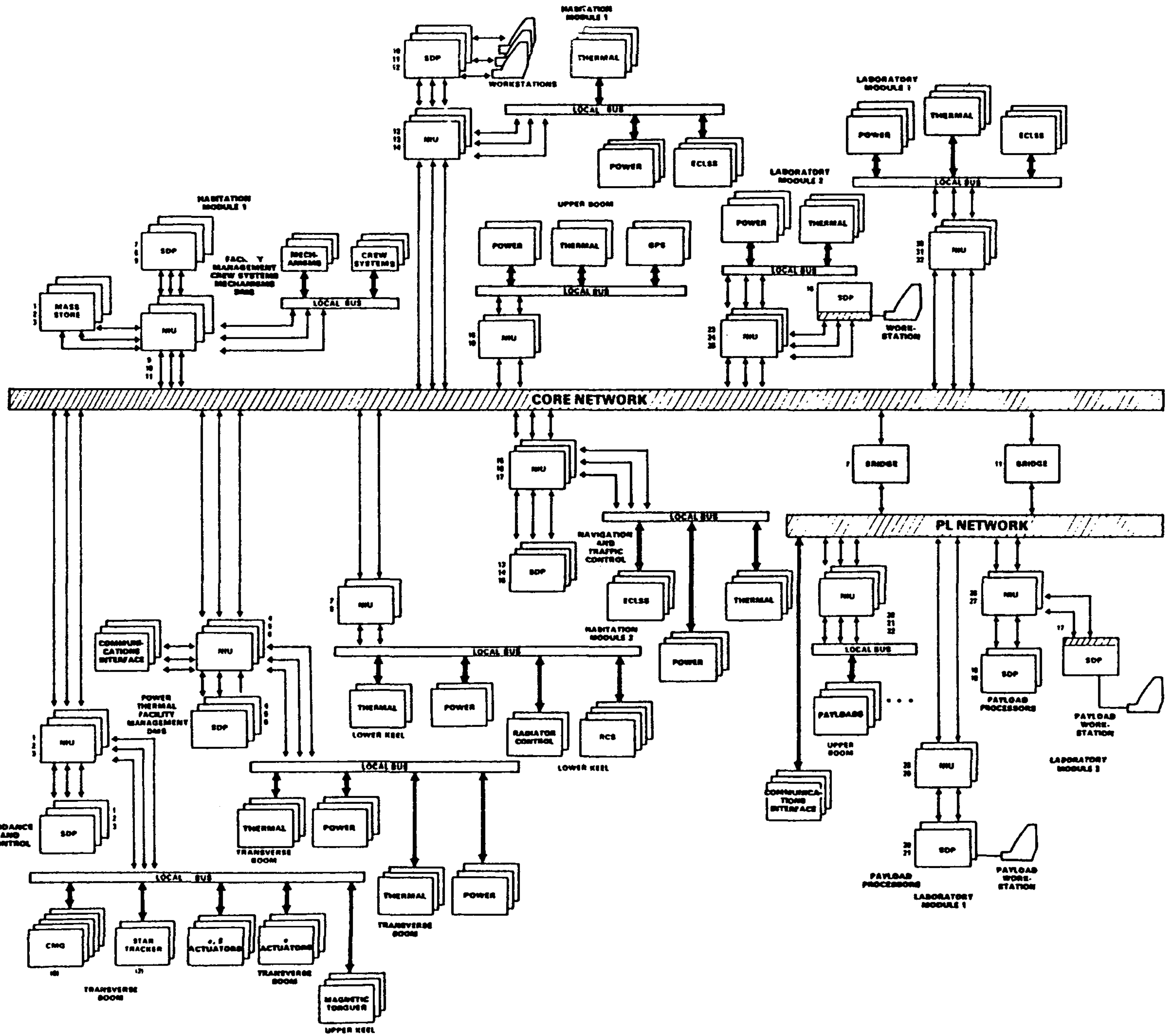

Figure 5. SS DMS System Layout.

station-critical subsystems. A wide variety of candidate architectural designs may be proposed for the distributed SS DMS. A key system engineering challenge is to assess whether a specific architectural design is capable of economically meeting overall DMS and SS mission requirements. This is the purpose of the computer-aided assessment tool and methodology described in the following sections.

SECTION 3 - MODEL REQUIREMENTS AND USER NEEDS

The Space Station DMS design issues are similar to those related to the design of any large, complex distributed computer system. These issues include the selection and distribution of equipment, the allocation of functions to each computer in the network, the choice of a networking scheme, and the selection of software operating systems, data storage formats, interface standards, and protocols. Because the SS DMS assessment methodology addresses all these areas, it may also find application in many other areas, such as space platforms and ground support data systems. The basic requirements for this computer-aided methodology are as follows:

- o Support of a database of SS DMS requirements, design options and technology options to facilitate the creation of DMS design models;
- o A set of computer tools to aid in the modeling and evaluation of performance, availability, cost, weight, power, and value of candidate DMS designs;
- o A methodology to support DMS design, selection, interactive evaluation, and integration of design activities of the Space Station contractors;
- o A means to evaluate the impact of new DMS technology on the Space Station.

NASA has organized the Space Station Program (SSP) using a three-tier management structure: Level A at NASA Headquarters, Level B at JSC, and Level C at the four NASA Centers which have been allocated responsibility for the Work Packages. Assessment needs for the SSP range from overall management and budget concerns (Level A), to major systems designs of the station (Level B), down to subsystem designs in the Work Packages (Level C). The assessment model must be capable of providing the following support to the three NASA levels of Space Station Program activity:

<u>SSP Level A:</u>

- o Cost, performance, and risk assessments of candidate SS DMS implementations for evaluation of programmatic options.

<u>SSP Level B:</u>

- o System level cost, performance, and reliability assessments for given requirements and subsystems design options.
- o Mechanism for maintaining consistency and configuration management of requirements and subsystem design model representations across the program.

<u>SSP Level C and Work Package Contractors:</u>

- o Modeling tool for evaluating subsystem candidate design trades at progressive layers of design detail in the presence of total system load.

Finally, the modeling tool must be portable. It must be readily available to a large group of potential users at many locations around the country. This requirement has led to the choice of the IBM PC XT personal computer to host the tool, because of its portability and prevalance at NASA Centers and contractor sites.

SECTION 4 - MODELING TOOL ARCHITECTURE

The overall architecture of the SS DMS assessment modeling tool (the "SSDMS Assessment Model") is shown in figure 6. The tool consists of an integrated set of databases and analysis algorithms that have been fashioned to support the construction of large, complex, distributed architecture models of DMS designs. The three databases shown at the left of the diagram are populated with current DMS system requirements, various software design options, and hardware technology options for components of the DMS. These databases can be interactively updated and extended as the SS DMS design evolves. These databases serve as libraries of requirements, design options, and technology options from which a modeling user can select items for inclusion in a specific candidate DMS model without having to re-enter all the detailed parameters associated with each item. For example, a specific type of processor can be included in the technology options database, with its associated set of performance, reliability, cost, weight, etc. parameters. A modeling user can include one or more instances of this processor type in a candidate DMS design model by merely referring to the processor type name when he defines the candidate model.

The requirements database consists of the functional and operational loading requirements levied on the DMS. Requirements are represented as end-to-end transactions. This database also provides a mechanism for function and data flow accountability. The design options database consists of sets of software designs for

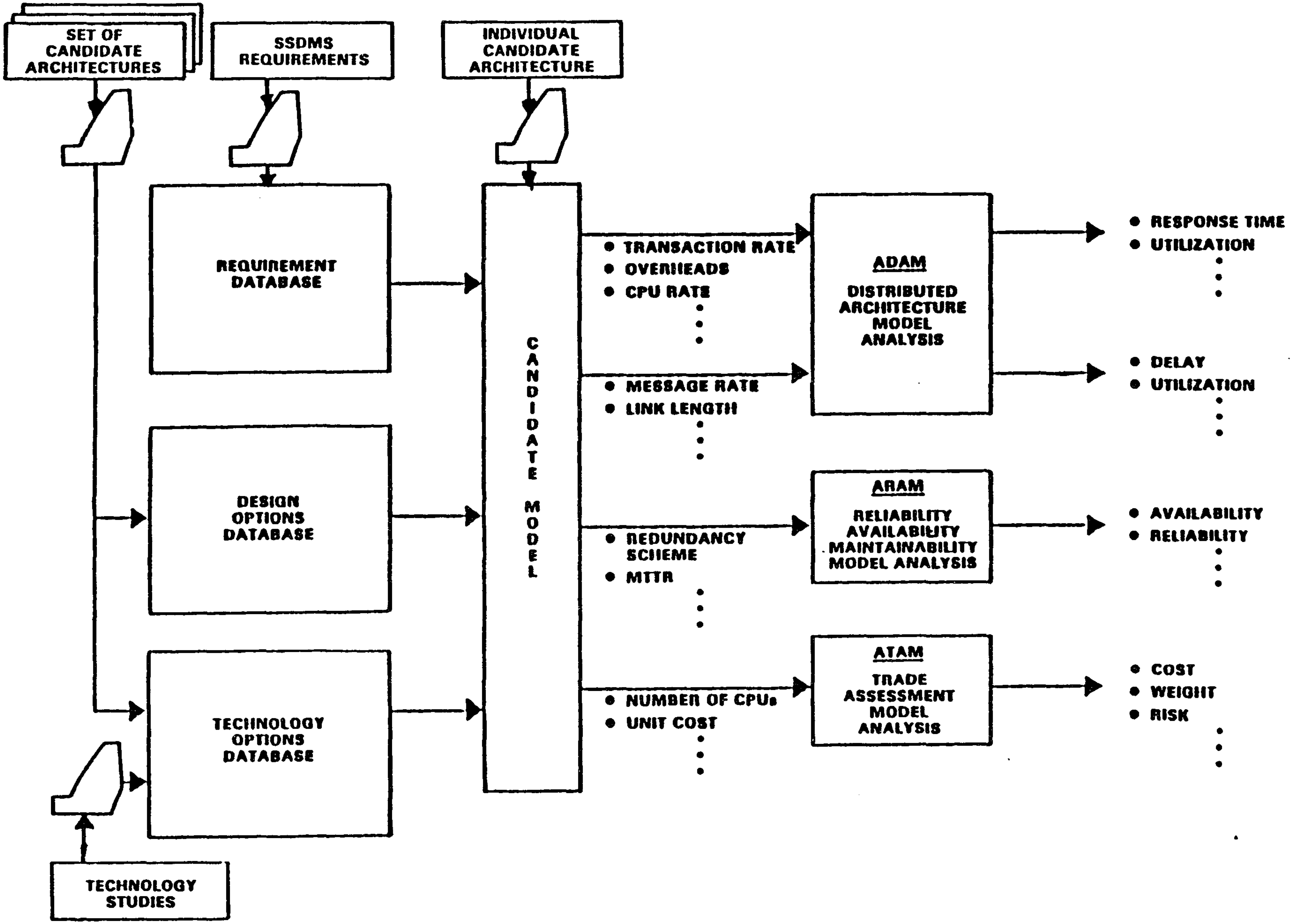

Figure 6. SS DMS Assessment Model Conceptual Architecture.

implementing candidate DMS architectures. The technology options database includes the information about specific hardware components which are candidates for inclusion in the DMS.

New information can easily be added as additional requirements, designs, and technologies mature for use. The databases provide the input for performing system performance, reliability, and cost analyses for specific candidate DMS architectures. As an aid to the modeling user, these databases will be populated with an initial version of a distributed architecture system used during the development of the tool. In addition, DMS designs resulting from two large NASA-sponsored DMS studies may also be loaded into the databases. The databases can accept other architectures or variations as appropriate.

The user creates the candidate model, located in the center of figure 6, by selecting appropriate requirements, design options, and technology options from the databases according to both the requirements and the characteristics of the candidate design being modeled. The candidate model will directly drive the analysis algorithms of the three model analysis programs provided (ADAM, ARAM, and ATAM). System performance characteristics, such as transaction response times and system component utilizations, are assessed by the Automated Distributed Architecture Model (ADAM) analysis program. The Automated Reliability/Availability/Maintainability Model (ARAM) analysis program evaluates the candidate architecture redundancy scheme, component mean time between failure (MTBF), component mean time to repair (MTTR), and repairman availability, and then predicts system availability and failure rates. The Automated Trade Assessment Model (ATAM) analysis program contains the algorithms needed for design, hardware, and technology trades involving system cost, weight, volume, power, and other parameters. The algorithms for these three model programs were developed to accomplish the objectives of the SS DMS assessment effort.

A specific candidate model is defined in a Design Model Definition file, which represents the selection and mapping of requirements, design options, and technology options from the databases onto components of the specific design. Minor modifications can be made to this file to model small variations in the design. This file, together with a baselined version of the databases, fully defines the DMS design being modeled.

The database architecture has been designed to allow a DMS design to be modeled at layered levels of detail. This will allow the tool to be useful at initial DMS design stages when only coarse design details are available, and to evolve with the design process to detailed DMS design stages when large amounts of design detail are available.

SECTION 5 - MODELING METHODOLOGY

This section describes the SS DMS Assessment Methodology used by the modeling user in conjunction with the modeling tool to develop and exercise models of DMS designs. The subsections below describe four aspects of this methodology: the modeling operations concept, the requirements data representation, the design data representation, and the technologies data representation.

5.1 OPERATIONS CONCEPT

Figure 7 shows the sequence of interactions that take place between the modeling user and the modeling tool in the process of populating the databases, constructing a specific DMS model, and exercising that model. The user initially populates each of the three databases with the model elements that represent overall SS DMS requirements, the results of technology studies, and a set of candidate architectures of interest. If the databases already contain much of this information from previous modeling efforts, the user need only augment the databases with those model elements that are unique to the individual candidate architectures or subsystem requirements he wishes to model. The user then selects the specific technology and design options from the databases to define the model for the candidate architecture of interest. Technology options are selected to define the hardware configuration of the candidate architecture model, and design options are selected to define the software design for the functions of the candidate architecture. The resultant DMS model can then be executed as directed by user-entered run-time parameters.

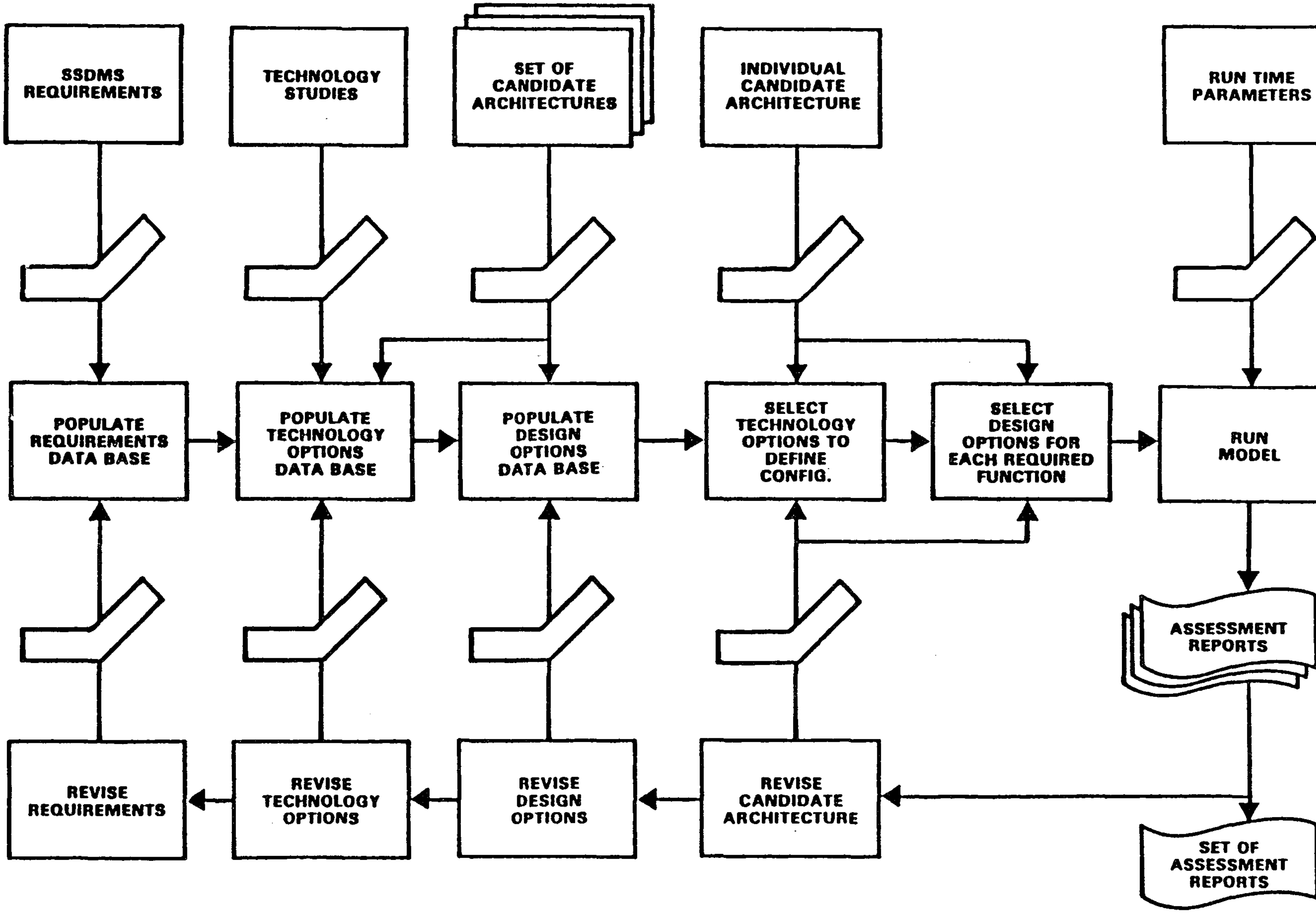

Figure 7. SS DMS Assessment Model Operations Concept.

The user can choose to have a component utilization and response time analysis performed by ADAM, an availability and failure rate analysis performed by ARAM, or a trade parameter analysis performed by ATAM. Assessment reports produced by these analysis programs are reviewed by the user, who may wish to revise the candidate architectures to optimize one or more of the assessment parameters. The user's revisions may involve augmenting the design and technology options databases if new types of software and hardware components are to be modeled. The requirements database may be also revised if alternative functional or workload requirements sets for the DMS are to be modeled.

5.2 REQUIREMENTS REPRESENTATION

The SS DMS Assessment Methodology uses a structured analysis technique for representing information in the requirements database. The approach is derived from T. DeMarco's structured analysis technique (see ref. 5), which is commonly used to define system requirements specifications. The technique uses data flow diagrams (DFDs) to logically represent the functions that a system must perform, the data interfaces between these functions, and external entities. One of the strengths of the technique is its graphical format, in which functions are represented by labeled, numbered bubbles and data flows are represented by labeled arrows. (See fig. 9 for an example of DFD.) The technique is hierarchical, allowing a function in a given DFD to be decomposed into its component subfunctions and data flows and represented in a lower level DFD.

A transaction is a related set of functions and data flows on a DFD that represent a sequence of activity within the system that is initiated by a stimulus at the boundary of the DFD. The stimulus may physically be the receipt of sensor data or of an input from outside the system, or the expiration of a timer interval. It may also be the receipt of a message generated by another component of the DMS design being modeled. A transaction may generate one or more response messages during its activation. A simple transaction is a single thread of functions and data flows extracted from a DFD. A transaction represents a processing requirement on the system; the frequency of occurrence of the transaction quantifies the required workload.

A transaction component is represented by a single data flow arrow on a DFD, together with both the portion of the source function bubble that is associated with generating that data flow, and the portion of the destination function bubble that is associated with receiving that data flow. A transaction can be represented by a set of transaction components, where all the processing activity associated with a transaction is reflected in the summation of the activities of the transaction components. The frequency of occurrence of a transaction component is a function of the frequencies of all the transactions to which the transaction component belongs.

To represent the requirements for the DMS, the user first develops a set of DFDs to specify the required functions and the data flows. The user then identifies the complete set of transactions that represent the required processing activity in the system and specifies their frequency of occurrence. The modeling tool computes the frequencies for each of the transaction components. The set of transactions, transaction components, stimulus and response messages, and frequencies define the DMS requirements being modeled.

5.3 DESIGN REPRESENTATION

The SS DMS Assessment Methodology uses a structured design technique for representing the software design information in the design options database. The approach uses software module structure charts to specify the hierarchical relationship of modules in a task. A task is a set of software modules that collectively performs an identifiable function, and all execute on the same processor. (See Section 7 for an example task structure chart.) The hierarchical structure of the modules in a structure chart specifies the module-calling relationships. A module is the lowest level software design unit, and is characterized by a number of executable instructions, a memory occupancy size, a development cost (delivered lines of code), and other parameters. Modules execute instructions, occupy memory, make data storage and peripheral device accesses, call for operating system resources, and transmit and receive messages.

A module path is a set of modules in a given task that is executed when a transaction component is activated. Each transaction component in the requirements database is associated with a set of module paths, to represent the software design that implements the requirement. When the modules in each referenced module path are executed, they impose a load on the processors and other hardware resources accessed by those modules.

To represent the software design for a candidate DMS architecture, the user first develops a set of structure charts that will implement the DMS functional requirements. Each structure chart defines a task that will be allocated to a specific processor. Instruction counts and other loading parameters are identified for each module. These design data are placed in the design options database. To create the specific candidate architecture model, the user selects these elements from the database and maps them to the requirements. The mapping is performed by specifying one or more module paths for each transaction component in the selected requirements set. The set of task, module, file, module path, and mapping definitions specify the DMS software design being modeled.

5.4 TECHNOLOGIES REPRESENTATION

The SS DMS Assessment Methodology uses a set of straightforward techniques for representing hardware configuration information in the technology database. The hardware elements that can be defined are processors, controllers, devices, and network elements. Network connectivity can be defined in terms of routing linkages. Component capacities and characteristics such as instruction execution rate, MTBF, MTTR, development cost, unit cost, maintenance cost, power consumption, weight, and volume are specified for each type of hardware element to be used, and are stored in the technology options database.

To represent the hardware configuration for a candidate DMS architecture, the user selects the number and types of hardware components desired from the database and specifies the hardware connectivity between the components. For example, for a mass storage device, the user specifies the controller and processor that control that device; for a processor, the user specifies its associated paging device and network interface element. To perform a reliability analysis, the user also specifies the redundancy configuration of the hardware, in terms of nested serial and parallel groups of components, and the criteria for component group and system availability. The user then maps the tasks and files of the candidate architecture software design to specific processors and storage devices in the hardware configuration. Each network message is mapped to a network routing linkage.

The set of hardware element definitions, the reliability configuration, and the mapping of tasks, files, and messages to hardware elements comprise the DMS hardware configuration being modeled.

SECTION 6 - DESIGN OF THE MODELING TOOL

This section describes the major characteristics of the SS DMS Assessment Model tool. In the four subsections below, some general design information is presented, followed by discussions of each of the three model analysis programs: ADAM, ARAM, and ATAM. The model analysis program subsections include discussions of their inputs, algorithms, and outputs.

6.1 GENERAL DESIGN

The SS DMS Assessment Model has been designed to run on an IBM PC XT or IBM PC AT personal computer having 640KB of main memory. It is written primarily in Microsoft Fortran and runs under the PC-DOS operating system. The Microrim RBASE 5000 database management system is used to manage many of the data files used by the tool and to generate many of the output reports.

6.2 ADAM

The Automated Distributed Architecture Model (ADAM) analysis program implements a set of analytic queuing algorithms that computes the utilization of hardware resources and response times for functional transactions. The types of model elements it supports include processors, controllers, devices, network routing linkages, transactions, transaction components, tasks, modules, module paths, files, messages, and network protocols. Almost 150 different parameters are used to describe the characteristics of these model elements. Default value mechanisms are provided to allow model elements to be used without the need for always specifying all their parameter values uniquely. A wide variety of local area network architectures can be modeled using a rich set of network protocol and routing linkage parameters. The service time distribution type can be specified independently for each hardware device. Some of the model parameters associated with a processor are the instruction execution rate, checkpoint issue and receive overheads, page size, paging device and paging overhead, network element, and network protocol. Some of the model parameters associated with a software module are main and loop instruction counts, language efficiency multiplier, processor priority, module size, fault rate, files accessed, and messages transmitted and received. Output reports include absolute loads and percent utilizations for each hardware component at each priority level. Contributions to these loads by each transaction component, task, and module path are also reported. End-to-end response times are reported for each transaction, as well as the contributions to these totals by each transaction component, task, and module path.

6.3 ARAM

The Automated Reliability, Availability, and Maintainability Model (ARAM) analysis program uses an event simulation approach to predict hardware system availability. A simulation approach was adopted because the equations in an analytical approach rapidly become intractible as the configurations and repair disciplines increase in complexity. The program uses a random number generator to simulate hardware component failures at each component MTBF rate. When a component fails, it is marked "down" until a repairman is available to fix the unit. The simulated repair time is randomly chosen according to the component's MTTR rate. Whenever a hardware component is simulated to have failed, the system redundancy configuration is consulted by the program to determine if sufficient components are still operating to meet the criteria for system availability. If a failed component causes a system availability criterion to be no longer satisfied, the entire system is marked "down" until a simulated component repair puts the system back in service. The program maintains statistics on component failure rates and system availability throughout the simulation.

The ARAM inputs include the system redundancy diagram, component MTBF and MTTR parameters, definitions of system failure and availability, the number of repairmen available, and the delay time before a repairman begins repair work. The redundancy diagram is represented in terms of nested serial and parallel groups of components. The failure and availability criteria are defined as the number of members of each parallel group that must be available for the entire group to be available. All members of a serial group must be available for the serial group to be available.

The ARAM can generate both summary and detailed reports for the simulation. Summary reports present the computed availability for each hardware component and group, including the entire system. The number of times each component or group cycled between available and failure states is also reported. Detailed reports trace the entire event simulation timeline indicating the time of each component failure and repair, intervals when components were queued awaiting a repairman, and intervals during which component groups or the entire system were down.

6.4 ATAM

The Automated Trade Assessment Model (ATAM) analysis program performs simple algebraic computations on trade parameter values to provide aggregate values for an entire system design. The ATAM inputs include: a list of all hardware and software components in the candidate architecture; development, unit, and maintenance costs for each component; weight, volume, and power consumption parameters; and development risk estimates. The ATAM reports include summations of trade parameters, such as cost across all components, and weighted assessments incorporating several trade parameters into an aggregate figure of merit.

SECTION 7 - MODEL OF A CANDIDATE GUIDANCE CONTROL SYSTEM FOR THE SPACE STATION

To demonstrate the utility of the SS DMS Assessment Methodology and to generate a realistic, detailed model for use in testing the SS DMS Assessment Model tool, a detailed hardware and software design for the SS DMS Guidance Control (G&C) system was developed and modeled (ref. 2). As shown in figure 3, the G&C system is one of the major DMS applications onboard the SS. The G&C system was chosen for this exercise because its high degree of complexity provided a good test of the methodology and because local expertise was available in this area to develop the candidate design. This example traces several of the initial steps taken in the modeling methodology to represent the design and obtain a performance analysis of the design. This example does not discuss the reliability and trade parameter assessment features of the modeling tool.

Figure 8 shows the reference G&C system hardware architecture (see ref. 1), which includes several Subsystem Data Procesors and related G&C sensors, actuators, and dedicated electronics. Although not shown here, performance, reliability, and trade parameter values for the hardware components were identified for each of the hardware components.

The SS G&C system provides four primary functions: orbit maintenance, thruster control, attitude control, and momentum management. Functional data flow diagrams (DFDs) were drawn for each of these areas. The composite DFD for the attitude control

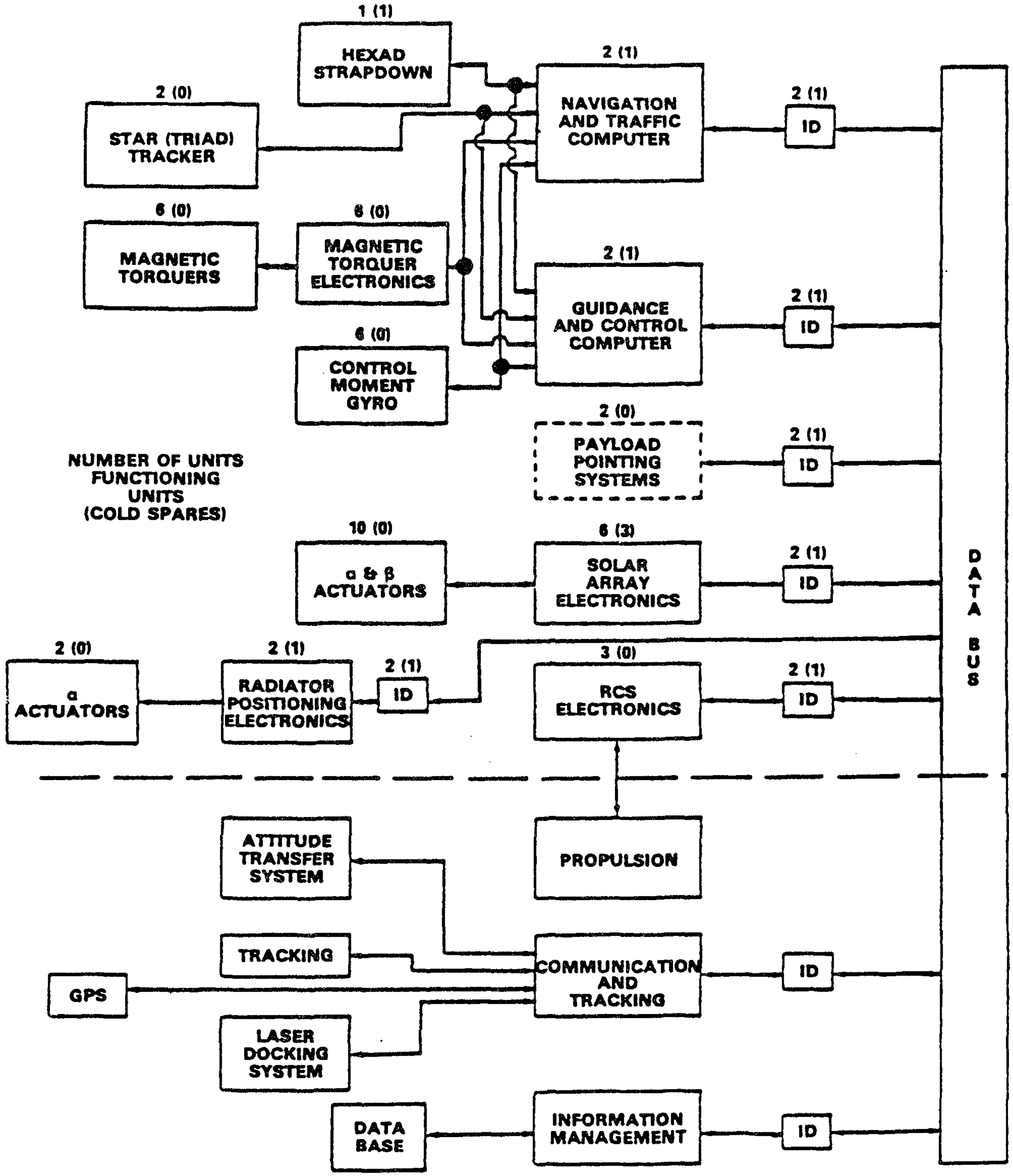

Figure 8. SS DMS GN&C System Architecture.

function is shown in figure 9. In this diagram, the numbered bubbles represent G&C software functions, and the unnumbered bubbles represent sensors and other DMS application subsystems. System transactions were identified as data flow threads through each of the DFDs. One of the simpler transactions in the attitude control function is shown in figure 10. This "Maintain Attitude" transaction represents a functional processing flow that is initiated by the receipt of the data item "gyro readings" from an external sensor. It causes four processing bubble functions to be performed and results in the transmission of a message to another subsystem. The five component data flows in this transaction map into five transaction components that impose a processing work load on the G&C system. Although not shown here, a parameterized transaction frequency was identified for each of the transactions. From this high-level workload definition, the modeling tool computes the frequencies for each transaction component.

Figure 11 shows the candidate software design developed for the attitude control function. In this design, the attitude control function executes entirely on one processor; therefore, one task was defined that consists of the 15 modules structured hierarchically as shown. Each software module is assigned a 9-character module identifier and a descriptive name. The numbers above each module box in the figure represent the estimated number of lines of code in the module. Although not

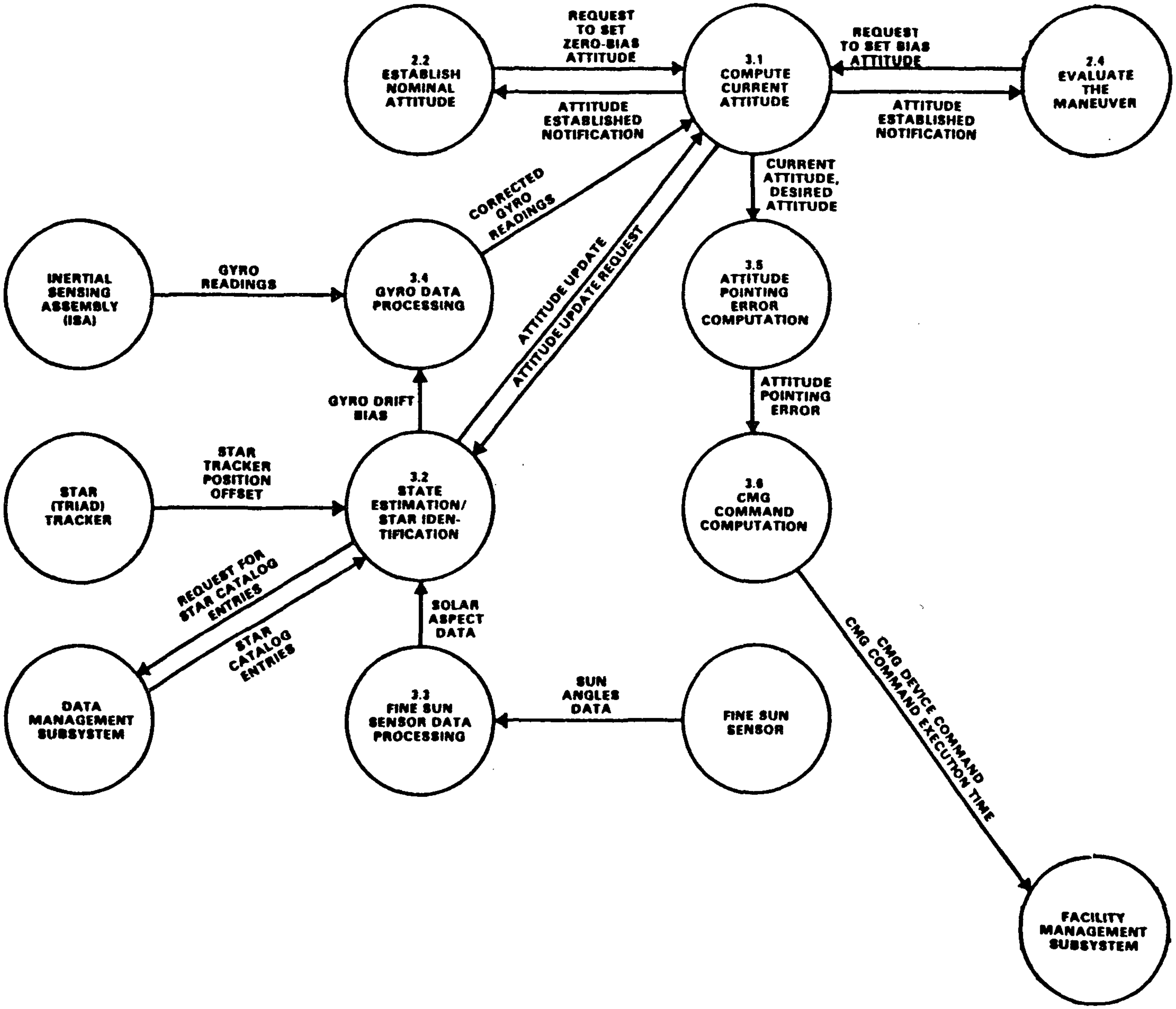

Figure 9. Attitude Control Composite Data Flow.

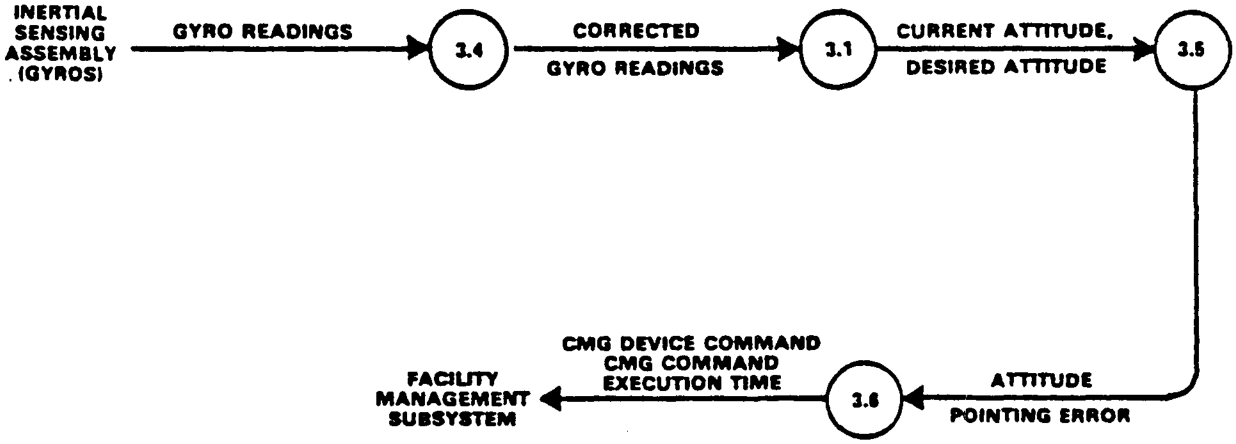

Figure 10. Maintain Attitude Transaction.

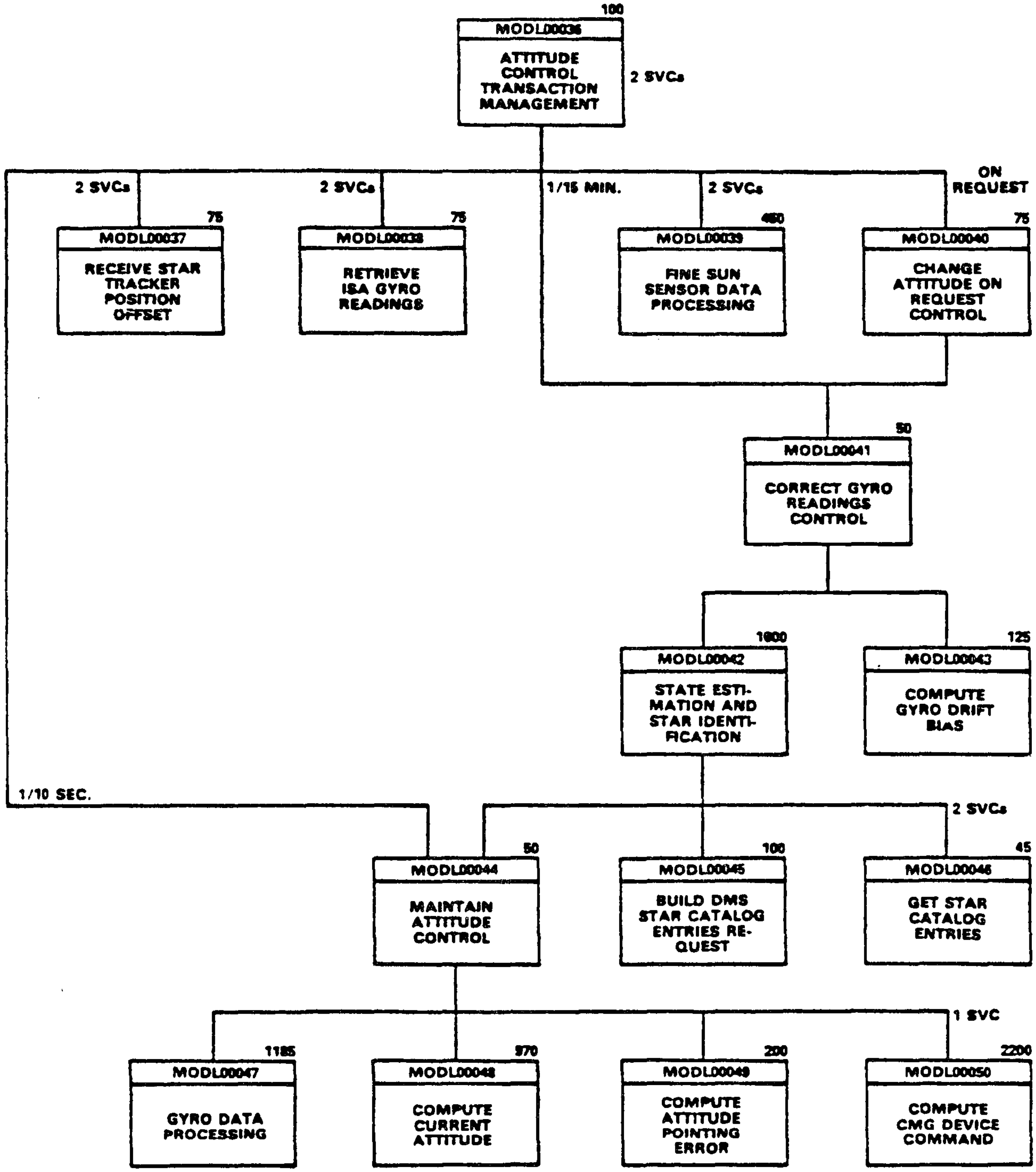

Figure 11. Attitude Control Task Structure Chart.

shown here, additional parameters were identified for each module, including the number of instructions executed per invocation.

Figure 12 shows the software module execution paths associated with each transaction component for the three transactions that span the attitude control task. The "Maintain Attitude" transaction shown in figure 10 is identified here as Transaction ID #08. Its five transaction components are identified as ID #31-35. Each of the transaction components is associated with the sequence of software modules (fig. 11) that will execute when that transaction component is activated. For example, an activation of transaction component ID #31 causes software modules #36 and #38 (MODL00036 and MODL00038 on Figure 11) to be executed.

Whenever the "Maintain Attitude" transaction is activated, each of its transaction components is activated, causing each of their respective module paths to be executed. The processor on which the attitude control task is mapped will experience the processing load of these modules, as represented in their number of application instructions executed, operating system calls, paging overheads, file accesses, and message transmissions. The modeling tool will sum the processing loads imposed by all transactions in the model to obtain the aggregate load on the attitude control processor, storage devices, controllers, etc. By dividing the resource loads by the hardware capacities, the tool will compute the percent utilization of each of the resources.

TRANSACTION		TRANSACTION COMPONENT		MODULE PATH	
ID	DESCRIPTION	ID	DATA FLOW LINK	ID	SEQUENCE
06	CHANGE ATTITUDE ON REQUEST	16	2.2 ⟶ 3.1	27	36, 40
		29	2.4 ⟶ 3.1	27	36, 40
		36	3.1 ⟶ 3.2	28	41,43, 42
		37	3.2 ⟶ DMS	29	45, 46
		38	DMS ⟶ 3.2	30	46
		39	STR ⟶ 3.2	31	36, 37
		40	FSS ⟶ 3.3	32	36, 39
		41	3.3 ⟶ 3.2	33	39
		42	3.2 ⟶ 3.1	33	39
		43	3.2 ⟶ 3.4	33	39
		31	ISA ⟶ 3.4	34	36, 38
		32	3.4 ⟶ 3.1	35	36, 44, 47
		33	3.1 ⟶ 3.5	36	48
		34	3.5 ⟶ 3.6	37	49
		35	3.6 ⟶ FAC	38	50
		17	3.1 ⟶ 2.2	39	36
		30	3.1 ⟶ 2.4	39	36
07	CORRECT GYRO READINGS	36	3.1 ⟶ 3.2	28	41, 43, 42
		37	3.2 ⟶ DMS	29	45, 46
		38	DMS ⟶ 3.2	30	46
		39	STR ⟶ 3.2	31	36, 37
		40	FSS ⟶ 3.3	32	36, 39
		41	3.3 ⟶ 3.2	33	39
		42	3.2 ⟶ 3.1	33	39
		43	3.2 ⟶ 3.4	33	39
		31	ISA ⟶ 3.4	34	36, 38
		32	3.4 ⟶ 3.1	35	36, 44, 47
		33	3.1 ⟶ 3.5	36	48
		34	3.5 ⟶ 3.6	37	49
		35	3.6 ⟶ FAC	38	50
08	MAINTAIN ATTITUDE	31	ISA ⟶ 3.4	34	36, 38
		32	3.4 ⟶ 3.1	35	36, 44, 47
		33	3.1 ⟶ 3.5	36	48
		34	3.5 ⟶ 3.6	37	49
		35	3.6 ⟶ FAC	38	50

NOTE: DMS – DATA MANAGEMENT SUBSYSTEM
FAC – FACILITY MANAGEMENT SUBSYSTEM
FSS – FINE SUN SENSOR
ISA – INERTIAL SENSING ASSEMBLY
STR – STAR TRACKER

Figure 12. Attitude Control Task Module Paths .

By applying the appropriate analytical queuing equations, the mean service and wait times are computed for each hardware component.

These response times are appropriately summed by the tool to provide end-to-end response times for each transaction, including the "Maintain Attitude" transaction.

As the example above has attempted to illustrate, this modeling methodology and tool provide a powerful and flexible modeling environment that allow the complex interactions between the hardware, software, and workload elements of a DMS design and other SS subsystem designs to be clearly and easily represented.

SECTION 8 - CONCLUDING REMARKS

A computer-assisted modeling tool and modeling methodology for use in assessing design concepts for the Space Station Data Management System have been described. Development of the tool and documentation of the methodology are nearing completion. The utility of the methodology is being demonstrated by representing a candidate design for the Space Station Guidance and Control System. Initial databases have been populated with parameters that represent a candidate Data Management System (DMS) design produced by a major NASA DMS Study Contractor.

This modeling tool and methodology will be available for the DMS design teams to support the Space Station Program Phase B definition and preliminary design effort. As the databases are populated with DMS design data, other subsystem designers will likely begin to use the tool to assist in performing Space Station subsystem design tradeoffs.

REFERENCES

1. NASA, JSC-19989, Space Station Reference Configuration Description, August 1984.

2. Computer Sciences Corporation, CSC/TM-85/6050, Computer-Aided Assessment Model Development for Space Station Data Management System Concepts, Space Station Representative Design (Revision 1), J. P. Craig, June 1985.

3. Computer Sciences Corporation, CSC/TM-85/6123, Automated Distributed Architecture Modeling (ADAM) Users' Guide, Mei-Zong Wu, December, 1985.

4. Computer Sciences Corporation, CSC/TM-85/6053, Automated Reliability/Availability/Maintainability (ARAM) Model Users' Guide, J. P. Craig, October, 1985.

5. T. DeMarco, Structured Analysis and System Specification, Yourdon Press, 1978.

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TM-87647		

4. Title and Subtitle

A Computer Modeling Methodology and Tool for Assessing Design Concepts for the Space Station Data Management System

5. Report Date

April 1986

6. Performing Organization Code

483-36-23-03

7. Author(s)

William R. Jones

8. Performing Organization Report No.

10. Work Unit No.

9. Performing Organization Name and Address

NASA Langley Research Center
Hampton, VA 23665-5225

11. Contract or Grant No.

13. Type of Report and Period Covered

Technical Memorandum

12. Sponsoring Agency Name and Address

National Aeronautics and Space Administration
Washington, DC 20546-0001

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

A computer modeling tool is being developed to assess candidate designs for the Space Station Data Management System (DMS). The DMS is to be a complex distributed computer system including the processor, storage devices, local area networks, and software that will support all processing functions onboard the Space Station. The modeling tool will allow a candidate design for the DMS, or for other subsystems that use the DMS, to be evaluated in terms of performance, reliability, cost, power consumption, weight, and other trade parameters. The tool and its associated modeling methodology are intended for use by DMS and subsystem designers to perform tradeoff analyses between design concepts using varied architectures and technologies.

17. Key Words (Suggested by Author(s))

Space Station
Data Management Systems
Computer Modeling
Reliability
Computer Architecture Analysis
Technology Assessments

18. Distribution Statement

Unclassified - Unlimited

Subject Category - 18

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
unclassified	unclassified	46	A03

N-305 For sale by the National Technical Information Service, Springfield, Virginia 22161