

**SYSTEM DESIGN OF A
SATELLITE RADIO FREQUENCY INTERFERENCE ANALYSIS SYSTEM**

by

Douglas Hall Descalzi


Project Report submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE
in
SYSTEMS ENGINEERING

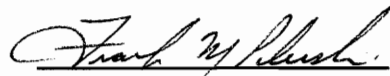
APPROVED:



T. Pratt, Chairman



B. S. Blanchard



F. M. Pilewski

May 1996

Blacksburg, VA

Keywords: satellite, communication, interference, systems, software

c.2

LD
5655
V851
1996
D473
c.2

SYSTEM DESIGN OF A
SATELLITE RADIO FREQUENCY INTERFERENCE ANALYSIS SYSTEM

by

Douglas Hall Descalzi

Committee Chairman: T. Pratt

Systems Engineering

(ABSTRACT)

The conceptual and preliminary design of the Dynamic Link Tool (DLTool) System are presented. The DLTool System performs moderate fidelity, satellite radio frequency (RF) interference and communications analysis. The primary customer of the DLTool System is Program 607 of Lockheed Martin Corporation (LMC) in Valley Forge, Pennsylvania. The system will dramatically improve the existing satellite RF interference analysis capabilities of Program 607, which are currently inadequate. Most importantly, the DLTool System will reduce the required analysis time for satellite RF interference problems from approximately 6 hours to 1 hour.

The DLTool System is intended to provide the capability to quickly predict or anticipate potential periods of RF interference. The design of the DLTool System includes an analysis of the needs of the customer, a feasibility study, the definition and allocation of operational and maintenance requirements, and the functional analysis of the system. The system is designed for a workstation-based local area network with simultaneous users, and includes a graphical user interface for input and output. The core component of the

DLTool System is customized C++ code that performs the computational analysis of user-defined satellite-ground station scenarios. The primary users of the system are communication engineers who will use the DLTool System to study RF interference issues for their customer.

ACKNOWLEDGMENTS

I would like to acknowledge each member of my review committee for their support and help in completing this project. First, I would like to thank a co-worker and friend, Frank Pilewski. He offered support, guidance, and a sense of humor when one was needed most. Frank took time out of his busy day to answer all of my questions, discuss any problems I was having, and review my project many times. Second, I would like to acknowledge Dr. Timothy Pratt. Although my project was not in his specific technical area of interest, he willingly and enthusiastically chaired my review committee. I appreciated his flexibility and understanding. Finally, I would like to thank Dean Blanchard and the Systems Program at Virginia Tech. Dean Blanchard accepted my many program changes throughout my degree and showed true interest in my education. Without these individuals, this project would have been far more difficult.

Finally, I would like to acknowledge my employers over the past 3 years: Lockheed Martin Corporation, the Corporation for Studies and Analysis, and Sycamore Associates, Incorporated. Their enthusiasm, support, generosity, and encouragement allowed me this opportunity. I truly appreciate their interest in my education and thank all of those responsible.

TABLE OF CONTENTS

	Page
INTRODUCTION	1
Background	1
Objective	5
Scope	5
CONCEPTUAL SYSTEM DESIGN	7
Definition of Need	7
Feasibility Study	10
Definition of System Operational Requirements	15
Maintenance Concept	22
PRELIMINARY SYSTEM DESIGN	25
System Functional Analysis	25
Allocation of Requirements	53
Trade-off and Optimization	56
Synthesis and Definition	60
CONCLUSIONS	62
REFERENCES	63
APPENDIX A: LIFE-CYCLE COST ANALYSIS	64
APPENDIX B: SYSTEM RELIABILITY ANALYSIS	69
APPENDIX C: ACRONYM LIST	72

TABLE OF FIGURES

	Page
Figure 1: Satellite Radio Frequency Interference Problem	4
Figure 2: Top Level Functional Flow	28
Figure 3: Operate System (FFD)	29
Figure 4: Build/Edit Scenario (FFD)	30
Figure 5: Load Scenario (FFD)	31
Figure 6: Create New Scenario (FFD)	32
Figure 7: Print Scenario (FFD)	33
Figure 8: View/Edit Scenario (FFD)	34
Figure 9: Select Scenario Path (FFD)	35
Figure 10: Compose Scenario (FFD)	36
Figure 11: Select Objects/Data to Print (FFD)	37
Figure 12: Select Objects/Data to Edit/View (FFD)	38
Figure 13: Perform Analysis (FFD)	39
Figure 14: Set Run Parameters (FFD)	40
Figure 15: View Output (FFD)	41
Figure 16: Select Case(s) (FFD)	42
Figure 17: Select Output (FFD)	43
Figure 18: Select Output Format (FFD)	44
Figure 19: Maintain System (FFD)	45

Figure 20: Perform Corrective Maintenance (FFD)	46
Figure 21: Call System Software Support (FFD)	47
Figure 22: Call Hardware Vendor (FFD)	48
Figure 23: Perform Preventive Maintenance (FFD)	49
Figure 24: Monitor the System (FFD)	50
Figure 25: System Utilization (FFD)	51
Figure 26: System Backups (FFD)	52
Figure 27: Work Breakdown and Allocation of Requirements	55
Figure 28: DLTool System Lifetime Schedule	65
Figure 29: DLTool System Labor Man-hours by Year	65
Figure 30: DLTool System Lifetime Cash Flow	66
Figure 31: Break-even Analysis For Satellite RF Interference Analysis Systems	67
Figure 32: DLTool Component Reliabilities for $t=1,000$ Hours	70

TABLE OF TABLES

Table 1: Summary of DLTool Functional Flows

27

INTRODUCTION

Background

Program 607 of Lockheed Martin Corporation (LMC) is a major systems integration effort for a classified United States government customer. Lockheed Martin employs approximately 50 people full-time on Program 607, primarily in Valley Forge, Pennsylvania. These individuals perform various analysis, monitoring, small software development, and other ad hoc activities. The mission of Program 607 is to provide system integration and support to their customer.

Most of the work performed by Program 607 involves orbit estimation and determination and satellite communications analysis. Over the many years that Lockheed Martin has worked Program 607, the company has built a solid reputation as a reliable, accurate, and timely resource for orbit and communication analysis and system integration. However, with the explosion of new technology, Lockheed Martin has not completely migrated or converted to new, more modern and powerful platforms and applications for their integration and analysis efforts. Due to the variety of platforms currently used at Lockheed Martin, there are tools written for Sun SPARCStations and tools written on old IBM mainframes. This lack of conformity has created many analysis tools and processes that are either too cumbersome or too slow by current standards.

Currently, Program 607 uses a variety of applications to perform satellite communication and interference analysis. Some of these applications run on old IBM mainframes and compete with hundreds of other jobs for computer resource. Other

applications and/or programs run on completely different platforms, such as Sun workstations. These various analysis tools are often run in sequence to analyze high priority and sensitive government situations. Thus, there are interface issues and difficulties associated with the satellite communications analysis cycle. In a typical situation, an orbit propagator on an IBM mainframe may run to compute ephemeris for several satellites. Next, a Matlab program will take the ephemeris and perform communications analysis on a Sun workstation. Finally, data from the Matlab program will be brought back to the IBM mainframe to use in a SAS application for more robust statistical analysis and processing.

Program 607 at Lockheed Martin Corporation is beginning to do much more satellite interference analysis than in previous years. However, the current process to perform satellite radio frequency interference analysis for Program 607 is inadequate. Demand for interference analysis is increasing with the growth of satellite communication in the 1990's. Every satellite launched requires a portion of the limited frequency range to communicate back to earth. Thus, the capability to anticipate or predict periods of potential radio frequency interference in a quick manner has become critical. The Dynamic Link Tool is intended to provide this critical capability for Program 607 of Lockheed Martin Corporation.

A graphical depiction of the satellite radio frequency interference problem is shown in Figure 1. Figure 1 shows a primary satellite, which is attempting to communicate with the earth-based ground station, and an interfering satellite. Based on the geometry of the situation, the power of the satellites, and the signal and antenna parameters of the satellites

and the ground station, the interfering satellite may disrupt communication between the primary satellite and the ground station. There is a definite need to be able to predict these periods of disrupted communication.

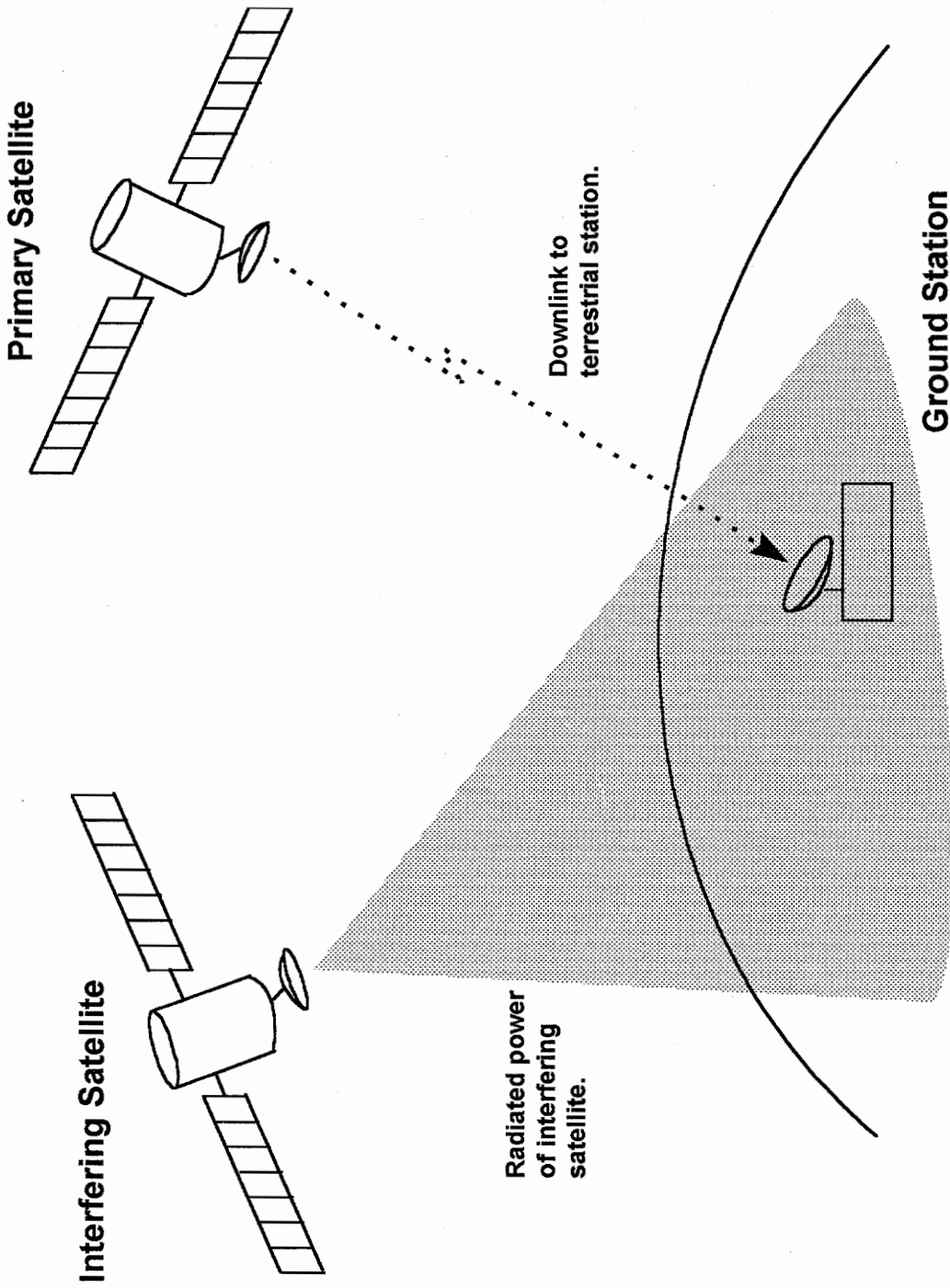


Figure 1: Satellite Radio Frequency Interference Problem

With the growth of satellite use over the last decade and multiple organizations interested in similar orbits and frequencies for their satellites, signal interference has become a very relevant issue. The current system in place to perform satellite communication and interference analysis in Program 607 is insufficient for the need.

Objective

The objective of this project is to engineer the design of a satellite communications and interference analysis system that will satisfy the needs of Program 607. The software tool that will become the primary component of the system shall be called the Dynamic Link Tool (DLTool). The systems engineering process will be utilized to determine the Conceptual System Design and the Preliminary System Design for the Dynamic Link Tool System. In systems engineering fashion, using a top down approach, the need for the DLTool system will be established, a feasibility study will be performed, the operational and maintenance requirements will be identified, and finally the design of the DLTool System will be presented.

Scope

The scope of this project is limited to the preliminary design of the DLTool System and those activities that are required to complete the design. Because the DLTool System

is primarily software, operational and maintenance requirements must be defined to direct the developers of the system and to provide familiarity with the necessary functionality.

The need for the DLTool System is immediate due to the current lack of capability. Work should begin as soon as possible to create the DLTool System such that it might support analyses by May 1997.

CONCEPTUAL SYSTEM DESIGN

Conceptual system design is the first step in the systems engineering process. The primary purpose of the conceptual design is to lay the foundation for the more advanced and detailed design activities later in the development process. In order to lay this foundation, several activities take place in conceptual system design:

1. Definition of Need
2. Feasibility Study
3. Definition of System Operational Requirements
4. Definition of Maintenance Concept [1]

When these activities are complete and reviewed, preliminary design of the system may begin.

Definition of Need

The definition of need is the first step in the system engineering process and the definition of system requirements. [1] Identification of the actual need is important to ensure that any solution alternative fulfills the need. Without defining the need, a system may be developed or procured that does not satisfy the consumer or user.

The need for a versatile, moderate fidelity, and quick satellite communications analysis capability in Program 607 of Lockheed Martin Corporation is real. Several employees were contacted to discuss and clarify the needs of the program. The need is

based on the existing capabilities available at Lockheed Martin, which do not suit quick turn-around analysis and employ out-of-date technology. Based on Lockheed Martin employee estimates, simple radio frequency (RF) interference analysis can take several man-hours (5-7) to set up and complete. Often, steps in the analysis need to be repeated or worked-around, due to the unreliable nature of the current system. To adequately support the analysis required, RF interference analysis needs to be complete in one hour. To support this turn around, the reliability of the analysis system needs to be no less than 99.99% per 1 hour or 91% per 1,000 hours (see Appendix B for system reliability calculations). Reliability describes the probability that a system will perform without failure over a specified time and thus is a critical measure of system performance. With a faster and more reliable system, Program 607 will be able to more quickly and accurately analyze their customers' radio frequency interference problems.

The basic need of Program 607 at Lockheed Martin is a tool or application capable of running satellite radio interference scenarios for the purpose of predicting periods of RF interference. The high level quantitative needs are summarized by the following list:

- Turn-around time of 1 hour for standard analysis procedures, including setup of scenario, computation of output, and high level analysis of results.
- Flexibility to perform moderate fidelity analysis.
- Overall system reliability of 99.99% per 1 hour and 91% per 1,000 hours (see Appendix B for system reliability calculations).
- Overall system availability of 99.2%.

- Support up to 30 analysts/users simultaneously and 100 simulated scenarios per day.
- Flexible and portable to several computing platforms, primarily Sun/Unix.
- Output in tabular and graphical format.
- Training time on system of 3 hours or less.
- Installation time of system on new hardware/network of 6 hours or less.
- Complete User's Manual.
- Complete maintenance concept.
- 10 year life cycle.
- Menu-driven graphical user interface (GUI).
- Store frequently used scenarios for future analysis and reference.
- Fully commented source code.

The current system in place for satellite radio frequency interference analysis only satisfies two of the above needs.

Due to the sensitivity and importance of the analysis that Program 607's client requires, as well as the antiquated means by which RF interference problems are analyzed currently, a more modern system is needed to study satellite RF interference problems.

The following represents a list of basic components required for a new system to perform satellite RF analysis:

- A stand-alone software program or application that performs the desired computational analysis.

- A graphical user interface for the software that provides panels for data input and for specialized tabular and graphical output.
- A database that stores and archives frequently used data for use in the software.
- Computer hardware to run the software program or application.

The above components constitute the framework of a satellite communications analysis system.

The cost of the system will be offset by the time savings produced by the improvements in the analysis process. Although the cost of developing the DLTool System is considerable, it is anticipated that the system will recoup the initial investment and start paying dividends by the third year of the life cycle (see Appendix A). In addition, the new system will provide new capabilities and functionality that is currently unavailable. Because there is no existing COTS product to fulfill Program 607's need, the possibility exists to market the tool outside Lockheed Martin and/or use the tool to win new business.

Feasibility Study

Feasibility studies are performed after the need for the system has been established to determine the technical approach for system development. [1] In this phase of the systems engineering process, different technological alternatives are evaluated with the goal of integrating technology into the system requirements definition process.

Many factors were considered when performing the feasibility study for the DLTool System. However, when discussing the true need with employees of Program 607 at Lockheed Martin, speed and data sharing seemed most important. Thus, a system was needed that provided results quickly and that included the capability of sharing data. Through the feasibility analysis, a system was recommended that includes the following components:

- Networked computer workstations.
- Custom-built code to perform computation analysis.
- Database containing stored scenarios/data.
- Menu-driven, graphical user interface.

The feasibility study is broken into functional areas: hardware configuration, software, data storage, and user-interface. These areas of the DLTool system are discussed in greater detail in the following sections.

Hardware Configuration

The hardware needs for the DLTool System are significant. The system will perform millions of computations every time it is used and thus computational power is important. In addition, the DLTool system requires networking capabilities to share data and resources. The network must be secure and controlled, and thus userid/password protection is needed. There are two primary hardware platform options that meet these high-level, basic needs: a personal computer (PC)-based local area (LAN) network and a Unix-based workstation network.

There are several advantages to building a PC LAN to host the DLTool System. A PC LAN is easy to use and provides considerable flexibility to users. There are thousands of commercial-off-the-shelf (COTS) products built for personal computers that could be useful for the development, use, and maintenance of the DLTool System. Also, networking PCs is not difficult; PC LANs are used effectively in millions of businesses worldwide. However, there are also several disadvantages to using a PC LAN for the DLTool system. Personal computers are not as fast as typical engineering workstations nor do they provide the robust data sharing capabilities. Typically, PC operating systems such as Windows 3.11, Windows NT, Windows 95 and IBM OS/2 are tailored to the home computer user or to perform less computationally intensive tasks. In addition, many of the operating systems for PCs are only now becoming multitasking and very few offer any distributed processing capability.

The other primary option for the hardware configuration for the DLTool System is an engineering workstation network. Although not quite as easy to use, most workstations run industrial operating systems such as Unix or VMS that are tailored toward more intensive processing and data sharing. Like PC LANs, workstations are easily connected to provide networking capabilities. Workstations networked together also provide some distributed processing capabilities as well as more computational power than personal computers. [4] Additionally, Program 607 currently uses a workstation based network and is thus familiar with its operation and maintenance. For these reasons, a hardware configuration utilizing computer workstations is recommended for the DLTool System.

Software

Several technical approaches exist for software that might support the requirements of the DLTool System. The first approach to consider is integration of existing COTS products to perform the required computational analysis. A COTS approach minimizes development time and thus can often minimize cost. For common problems or analytical techniques that exist in industry such as statistical analysis or orbit propagation, using COTS products is often superior to custom code that could be developed for the needs of the system primarily due to cost considerations. However, integration of COTS products rarely produces a tool that is ideally suited for the mission for which the tool was intended. Therefore, the customer or user must settle for a final product that may not suit his or her needs. In addition, integration of COTS products puts a strong dependence on the COTS vendors for support, maintenance, upgrades, reliability, and other effectiveness factors.

The other alternative for software is custom built code, which may be structured or object oriented. Custom built code has the advantages that are the disadvantages of integrating COTS products, and vice versa. Developing custom code is, in most cases, more expensive and time consuming than integrating COTS products. However, custom built code may be designed and developed to suit the actual needs of the user. Upgrades, maintenance, reliability, availability, support, and other factors are also under the control of the developer by building custom code, rather than external software vendors. Due to

the special needs of the DLTool System as well as the advantages of better control, custom built code is recommended for the DLTool System.

Data Storage

The DLTool system will require a data storage or repository capability to support run-time processes as well as post run-time analysis. The data produced from a single use of the system could reach 100 Megabytes. There are two primary options to control and manage data: data stored in flat files or in a database.

Flat files are common and simple. They make design of the input and output functions of the system easy and quite flexible. However, storing data in flat files is best suited for small amounts of data or when only one user will be accessing the data. A database application defines data structures and formats and more efficiently handles data for multiple users. Database programs are fast, reliable, and allow many query capabilities not available with other data storing options. Databases provide data or record locking capabilities, data security, and other functions as well. Although integrating a database program into the system represents more work and development time, the benefits overwhelm the disadvantages. Thus, utilizing a database package for data storage is recommended highly for the DLTool System.

User Interface

The user interface is critical to the success of the DLTool System. The interface must be simple and intuitive, yet provide the flexibility for the functionality that must be

included in the system. [2] There are a few general alternatives for user interface development. A text-based interface could be developed for the DLTool System, however this does not fill the Program 607 need. Text-based interfaces are often more tedious to use and provide less functionality than graphics or window-based interfaces. In addition, Text-based interfaces require the user to have a greater knowledge of the system to be effective and thus require greater training time. Graphical user interfaces allow quicker training and ease in understanding. Graphical user interfaces also provide better means of viewing output with color plots, diagrams, animation, etc. GUI has also become the standard type of interface for modern software development and there are various COTS packages that can be used to support custom interface development. [2] For these reasons, a GUI is recommended for the DLTool System.

Definition of System Operational Requirements

The system operational requirements flow from the need and feasibility study. Operational requirements define and establish the technical parameters for system design. The definition of the system operational requirements includes the following:

1. Mission definition.
2. Performance and physical parameters.
3. Use requirements.
4. Operational deployment or distribution.
5. Operational life cycle.

6. Effectiveness factors.
7. Environment. [1]

Each of these components is discussed as applied to the DLTool System in the following sections.

Mission Definition

The primary mission of the DLTool System is to predict periods of potential satellite radio frequency interference. The secondary missions of the DLTool System are ad hoc satellite communications analysis and ad hoc satellite orbit analysis. The system will have a moderate level of fidelity and satisfy the high level requirements defined in the Definition of Need. The average use of the DLTool System is expected to be 100 simulations per week.

The DLTool System will consist primarily of custom-built software and will be installed on customer hardware. The DLTool user will have the following capabilities:

For the simulated scenario:

- Define the simulation start and end dates and times. (Simulation duration is limited only by system hardware.)
- Define type of orbit propagation.
- Define an unlimited number of satellites and ground stations to be included in the scenario.
- Save scenario input data to database.

- Recall scenario input data from database.

For each satellite in the scenario:

- Define the satellite orbit in Keplerian elements or 3-card element set.
- Define satellite epoch date and time.
- Define satellite stabilization mechanism.
- Define the antenna pattern(s) for the satellite.
- Define satellite signal parameters.
- Save satellite input data to database.
- Recall satellite input data from database.

For each ground station in the scenario:

- Define the location and altitude of the ground station.
- Define the antenna pattern(s) for the ground station.
- Define geometrical constraints of ground station.
- Define climate region of ground station.
- Save ground station input data to database.
- Recall ground station input data to database.

Output formats:

- Tabular output of data (where applicable).
- 2-D plotting of communication and geometric data (where applicable).
- 3-D visualization/animation for satellite orbits.
- View all output on screen or in hardcopy.

Performance and Physical Parameters

The following parameters describe the operating requirements of the DLTool

System:

- System shall have at least 30 capable workstations.
- Workstations shall have 32 Megabytes of RAM.
- Workstations shall have a minimum clock speed of 90 MHz.
- Workstations shall have at least one local 2 Gigabyte SCSI drive.
- Workstations shall have color displays of at least 19 inches.
- Workstations shall be connected to support 10 Megabyte/sec network traffic.
- Workstations shall support GUI panels.
- Workstations shall have keyboard and mouse for usability.
- Installed software and required libraries shall not exceed 200 Megabytes.
- Software shall support 30 simultaneous users.
- Database shall have capacity to store a minimum of 2000 scenarios.
- Database shall support a maximum retrieval time of 2.5 seconds.
- Database shall provide interface for SQL compliant routines.
- Interface menus shall not exceed a depth of 3 levels.
- Interface shall support simultaneous analysis of multiple scenarios.

Use Requirements

The DLTool System is designed for use 24 hours a day, 7 days per week, except for one hour per week scheduled for preventive maintenance. The maintenance may be scheduled during off peak hours. Higher levels of use are expected Monday through Friday, due to the standard work week.

Operational Deployment

The hardware for the DLTool System is currently in place. The software, including the GUI code, will be delivered to Program 607 in Valley Forge, PA by DLTool software developers. The network administrators will load the software onto the Program 607 SPARCStation LAN and setup the required libraries with supervision from DLTool development personnel. Installation and site-test will last approximately 1.5 days, at which time the DLTool System will become operational. All upgrades to the DLTool software will follow this procedure as well.

Operational Life Cycle

The DLTool System is designed to operate for approximately 10 years. Throughout this lifetime, periodic upgrades in computer hardware are expected as well as enhancements to the DLTool software. The network administrators at Program 607 will add to their duties the maintenance of the DLTool system. Maintenance is not expected to exceed 1-2 hours per week once the system is installed. The primary role of the network administrators is to provide configuration and data management for the

operational system. Once testing of the software is complete, maintenance of the code will require less than 1 hour per week. Users of DLTool will be communication engineers who are familiar with the technical aspects of satellite communication analysis.

Later development phases of the DLTool System are possible since changes in satellite communication technology are likely. However, the DLTool System is designed to minimize work required for maintenance or upgrades. Thus, additional upgrade phases would require relatively low levels of effort and would not impact the current operational code. Cost figures were developed that allow for minor upgrades and are included in the life cycle cost (see Appendix A). Major overhauls of the software will require a rework of the costs of the system.

Effectiveness Factors

The DLTool System is designed for moderate fidelity analysis and quick turn-around time. These factors necessitate a design that does not require extensive computer knowledge. Network administrators will monitor the system using network tools that will show resource utilization, speed, processing time, etc. The effectiveness factors for the DLTool System are:

- System operational availability of 99.2%.
- System reliability shall be no less than 99.99% for 1 hour and 91.0% for 1,000 hours (see calculations in Appendix B).
- The Mean Preventive Maintenance (Mpt) time shall be one hour per week.

- The Mean Time Between Failures (MTBF) shall be no less than 10,600 hours.
- The total cost of the system shall not exceed \$450,000.

Environment

All of the hardware and COTS components of the DLTool System already exist at Program 607's facility in Valley Forge. Version 1 of the DLTool software will be installed exclusively on the Program 607 LAN. The software will be delivered on 8mm tape which will include source code, executable and any special development libraries not resident on the Program 607 LAN. A backup of the tape will also be delivered for archival and backup purposes.

The ideal room conditions for the DLTool System are 60 to 78 degrees Fahrenheit and humidity between 40 and 55 percent; however, LMC network administrators and facilities personnel will control the environment. This environment optimizes human and machine performance.

Storage and server facilities should contain hazard and backup systems to reduce risk of hardware, software, and data loss. Rooms where servers are located should be protected by sensors that monitor temperature and detect smoke. Servers should also be connected to alternate power supplies in the case of a surge or temporary loss of electric power. Tape backup systems should perform daily backups of data to avoid loss of work due to data corruption or other unexpected problem.

Maintenance Concept

The system maintenance concept evolves from the system operational requirements and is intended to identify the basic levels, responsibilities, and policies of maintenance support. The maintenance concept includes logistical support, effectiveness factors, and maintenance environment requirements. The goal of the maintenance concept is to provide a baseline for supportability requirements and logistical support. [1]

The DLTool System maintenance concept contains three levels: organizational maintenance, intermediate maintenance, and supplier maintenance. Each level is described in greater detail in the following sections.

Organizational Maintenance

Organizational maintenance is performed at the operating site of the system. For the DLTool System, the computer network administrators will perform the organizational maintenance. Most of organizational maintenance is preventive with very little down time. The skill level required for organizational maintenance is relatively low, however some computer networking knowledge is necessary. The following list contains the primary tasks for organizational maintenance:

- Monitor network use, volatility, and traffic levels.
- Monitor resource utilization and allocated space on network drives.
- Perform backups of system configuration including database.
- Log all system anomalies for report to intermediate level.

- Control access to data and software.

Technical DLTool System manuals will also support operational maintenance tasks.

Intermediate Maintenance

Intermediate maintenance tasks are typically more complex and often require greater skills than organizational maintenance tasks. Primary suppliers of intermediate maintenance are DLTool System developers for the software and the hardware vendors for the workstations and network. Intermediate maintenance is still performed on-site at the customer location. A maintenance and service contract will be written with developers and vendors with the following requirements:

- Hardware and database vendors will respond to problems within 18 hours of call on weekdays, within 30 hours of call on weekends.
- DLTool System developers will respond to software problems within 18 hours of call on weekdays, within 30 hours of call on weekends.
- All software upgrades and/or fixes shall be unit and system tested by the software developers before becoming operational on the Program 607 LAN.
- DLTool software developers will provide updated documentation for every delivery or change in the operational baseline.
- All hardware upgrades shall be tested at the vendor site prior to being introduced to the Program 607 LAN; Program 607 network administrators will add the tested hardware to the system.

- The mean corrective maintenance time (Mct) of the DLTool System is 4 hours.
- The maximum maintenance downtime of the DLTool System is 24 hours on weekdays and 36 hours on weekends.
- The mean maintenance downtime of the DLTool System is 1.35 hours.

Supplier Maintenance

Supplier maintenance tasks require the highest skills and largest amount of time.

Supplier maintenance is usually performed at a vendor location and includes the use of specialized equipment and highly trained workers. Supplier maintenance tasks include the following:

- Replacing system components at the customer location to perform repair at the vendor location.
- Disposal of faulty components at the vendor location or using the faulty components for spare parts after repair.

PRELIMINARY SYSTEM DESIGN

Preliminary system design is the process by which detailed quantitative and qualitative requirements are derived from the conceptual design of the system.

Developing the preliminary system design is an iterative and continuous process that requires several distinct steps:

1. System Functional Analysis
2. Allocation of Requirements
3. Trade-off and Optimization
4. Synthesis and Definition [1]

Each of these steps will be discussed for the DLTool System in the following sections.

System Functional Analysis

The objective of system functional analysis is to translate system operational and support requirements into specific qualitative and quantitative design requirements. [1]

The functional analysis for the DLTool System will be performed using functional flow diagrams (FFD). The primary purpose of functional flow diagrams is to structure system requirements into functional terms. The diagrams are built to identify what needs to be accomplished, rather than how something should be done.

The following set of figures illustrate the operational and functional flows for the DLTool System. Figure 2 represents the top level functional flow and contains the basic

steps in the traditional system lifecycle. The final two boxes in Figure 2, “Operate system” and “Maintain system”, are addressed in detail in the functional analysis of the DLTool System. Figures 3-18 show the high level operational functional flow while Figures 19-26 show the maintenance functional flow. Note that all diagrams in the functional analysis of the DLTool system are related and may be traced back to Figure 2; in this sense, the functional flows that follow constitute one large diagram. Also note that the functional flows presented are not intended to show the technical aspects of RF interference analysis, rather they are intended to show how the user would operate and maintain the DLTool system at a high level.

Table 1 summarizes the functional flows for the DLTool System. Table 1 contains figure number, figure title, functional class, parent figure number, and flow level for each functional flow diagram. The class defines whether the flow is operational or maintenance, and the flow level designates how many flow diagrams exist above the current flow.

Table 1: Summary of DLTool Functional Flows

Figure No.	Figure Title	Functional Class	Parent Figure No.	Flow Level
2	Top Level Functional Flow	Top Level	N/A	1
3	Operate System	Operational	2	2
4	Build/Edit Scenario	Operational	3	3
5	Load Scenario	Operational	4	4
6	Create New Scenario	Operational	4	4
7	Print Scenario	Operational	4	4
8	View/Edit Scenario	Operational	4	4
9	Select Scenario Path	Operational	5	5
10	Compose Scenario	Operational	6	5
11	Select Objects/Data to Print	Operational	7	5
12	Select Objects/Data to Edit/View	Operational	8	5
13	Perform Analysis	Operational	3	3
14	Set Run Parameters	Operational	13	4
15	View Output	Operational	13	4
16	Select Case(s)	Operational	15	5
17	Select Output	Operational	15	5
18	Select Output Format	Operational	15	5
19	Maintain System	Maintenance	2	2
20	Perform Corrective Maintenance	Maintenance	19	3
21	Call System Software Support	Maintenance	20	4
22	Call Hardware Vendor	Maintenance	20	4
23	Perform Preventive Maintenance	Maintenance	19	3
24	Monitor the System	Maintenance	23	4
25	System Utilization	Maintenance	24	5
26	System Backups	Maintenance	23	4

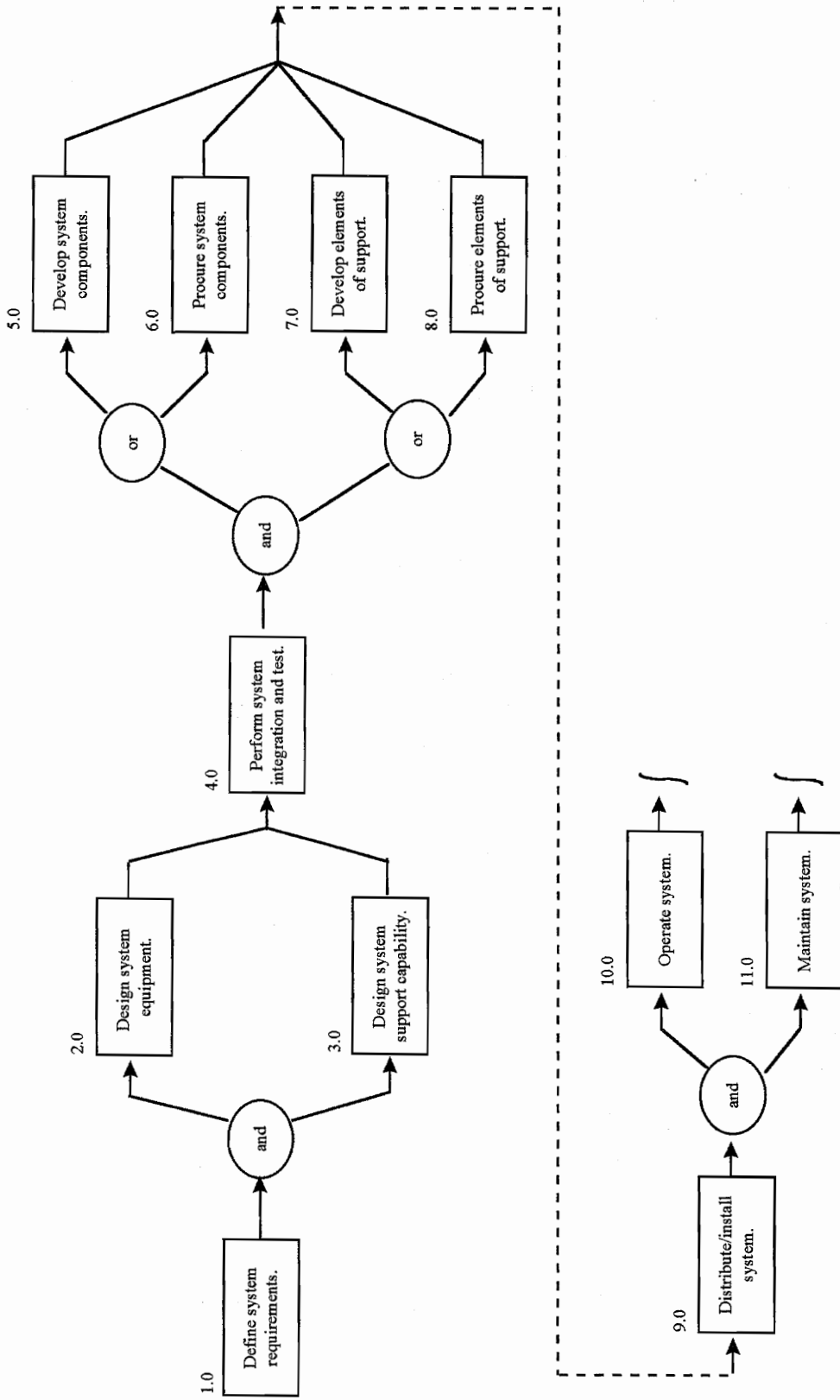


Figure 2: Top Level Functional Flow

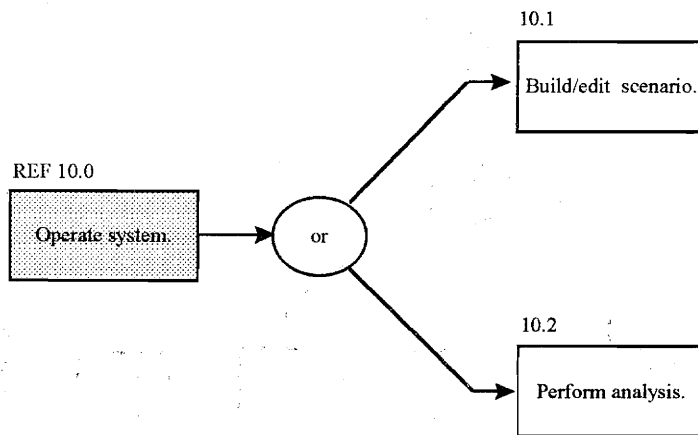


Figure 3: Operate System (FFD)

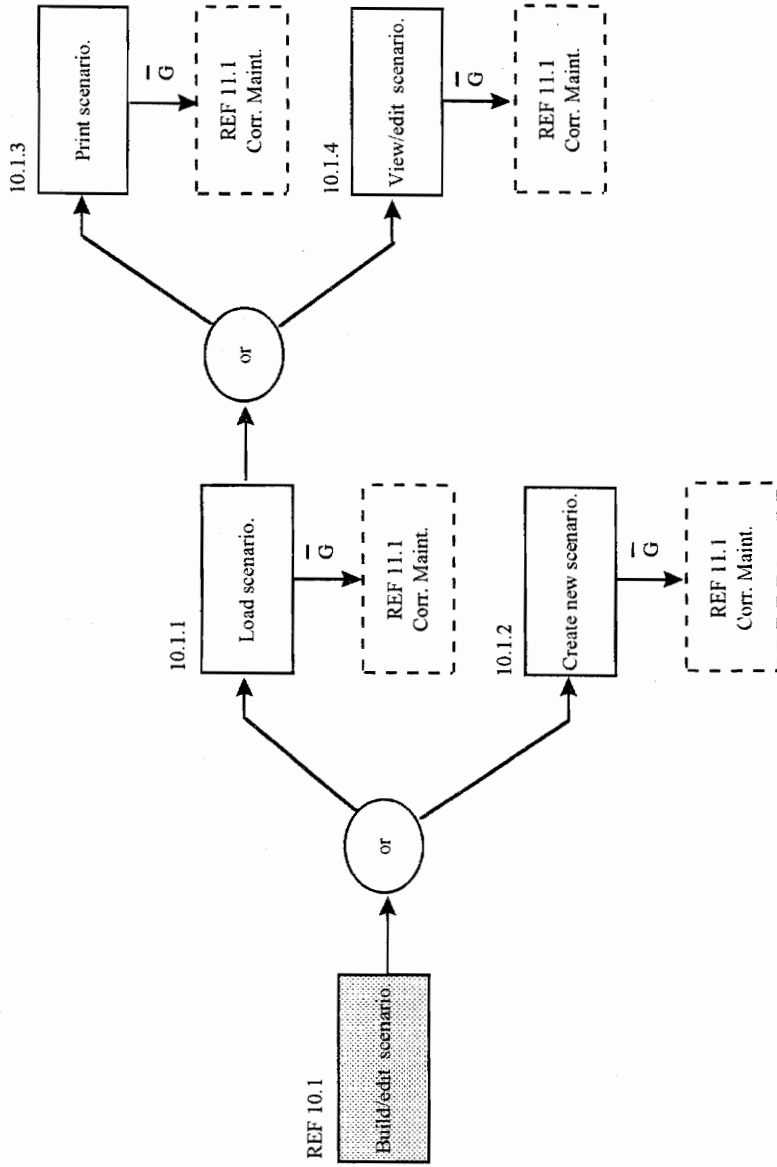


Figure 4: Build/Edit Scenario (FFD)

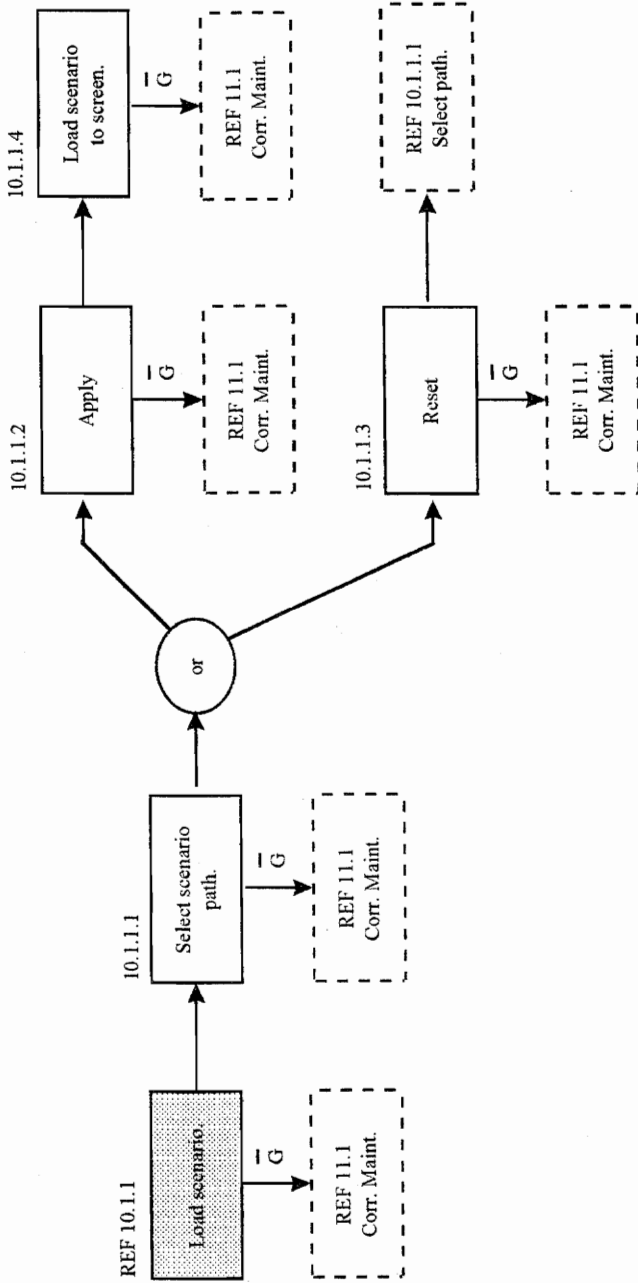


Figure 5: Load Scenario (FFD)

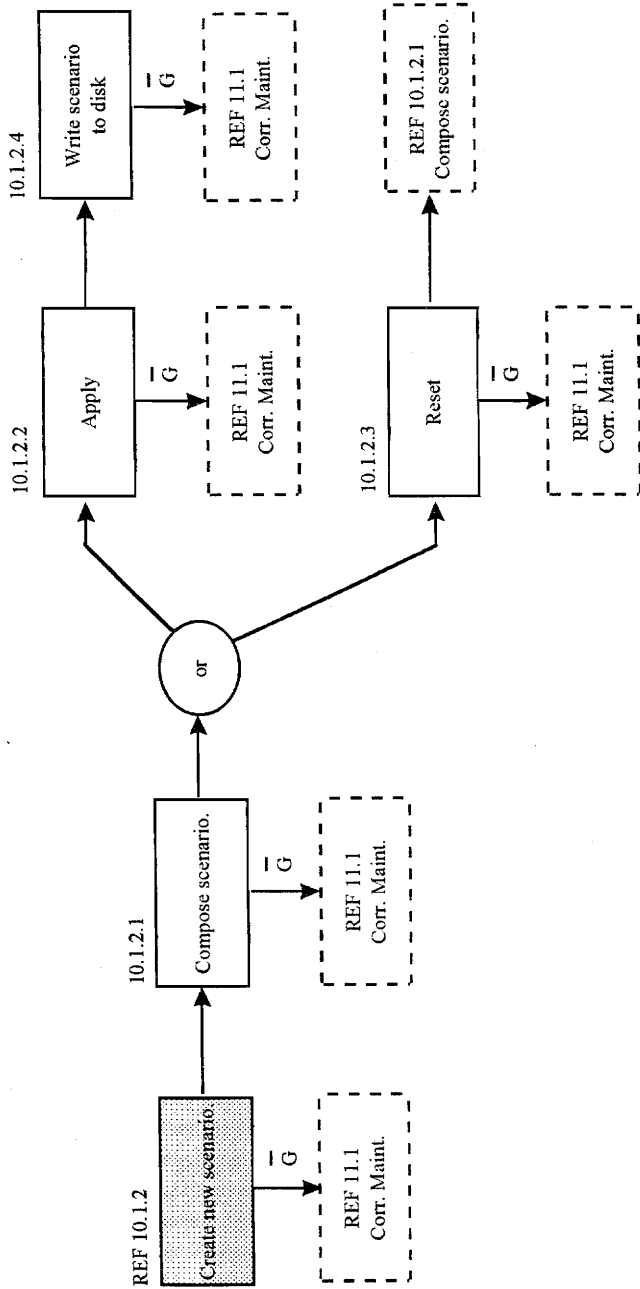


Figure 6: Create New Scenario (FFD)

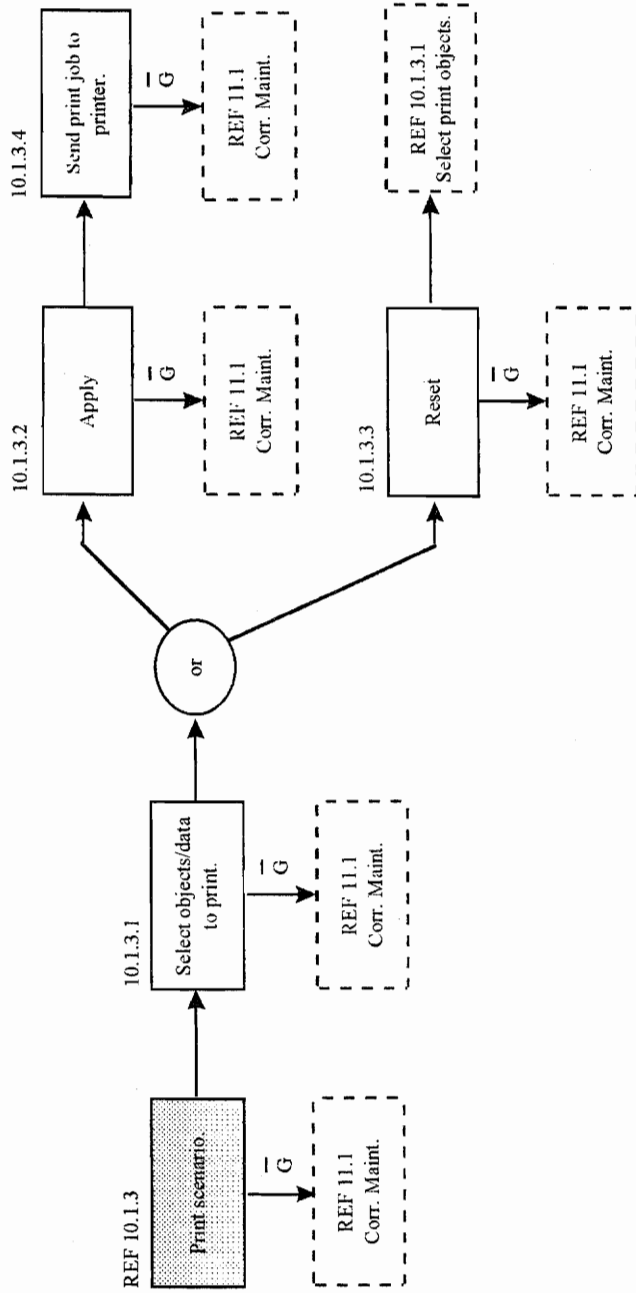


Figure 7: Print Scenario (FFD)

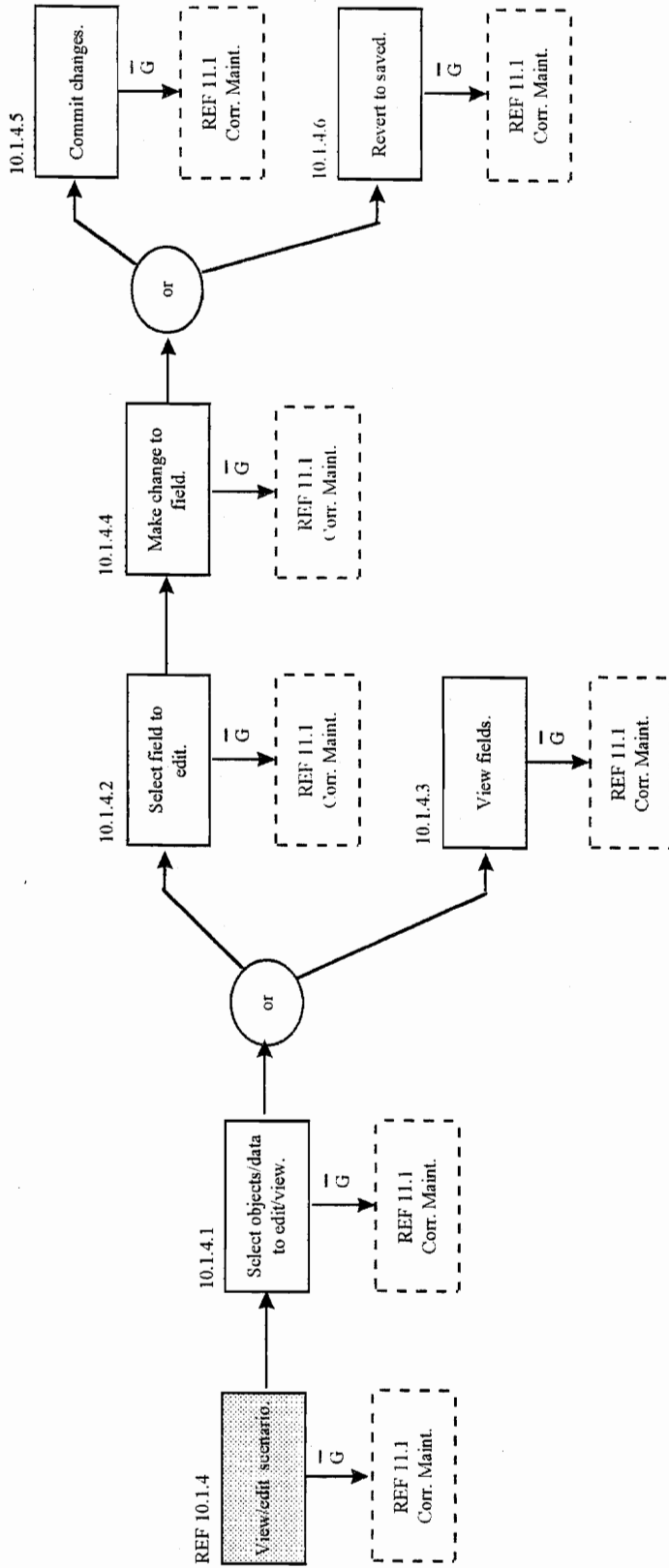


Figure 8: View/Edit Scenario (FFD)

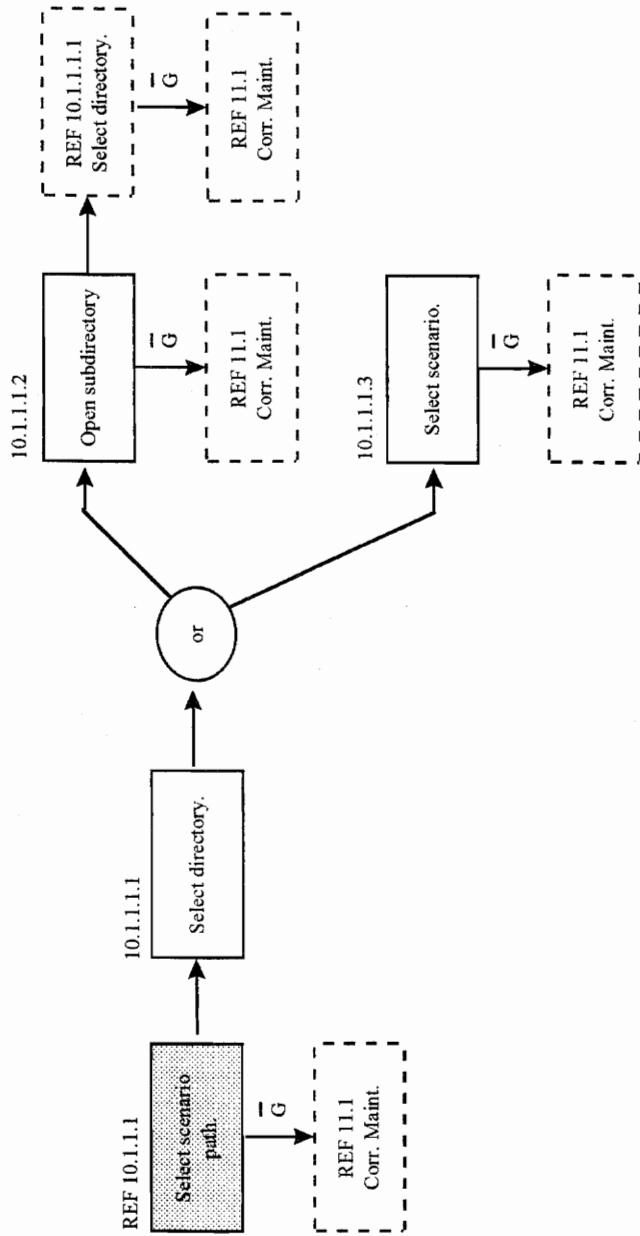


Figure 9: Select Scenario Path (FFD)

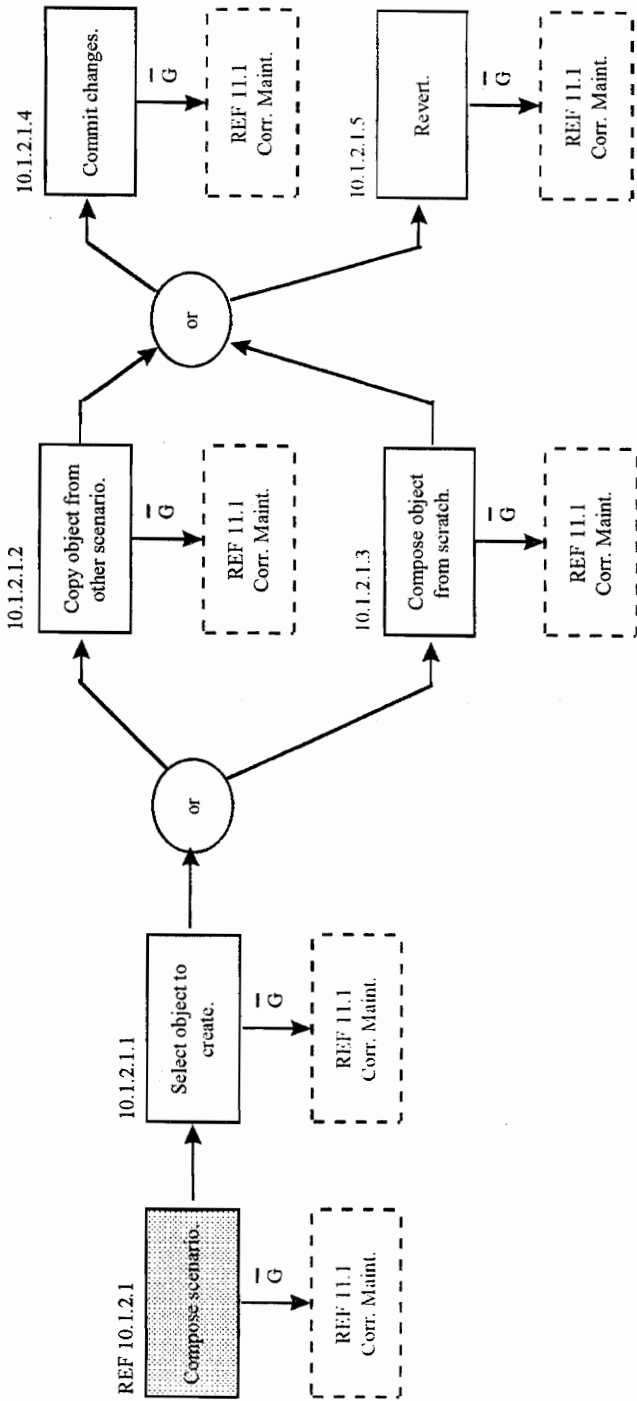


Figure 10: Compose Scenario (FFD)

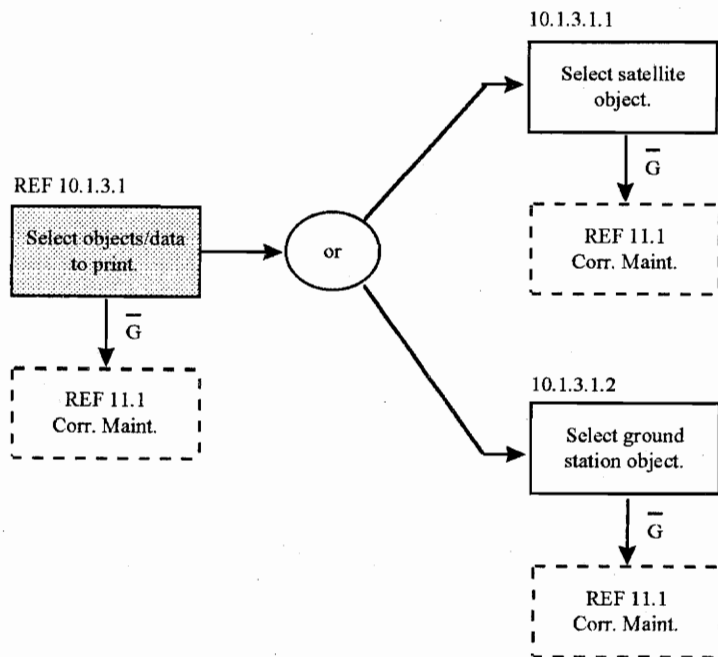


Figure 11: Select Objects/Data to Print (FFD)

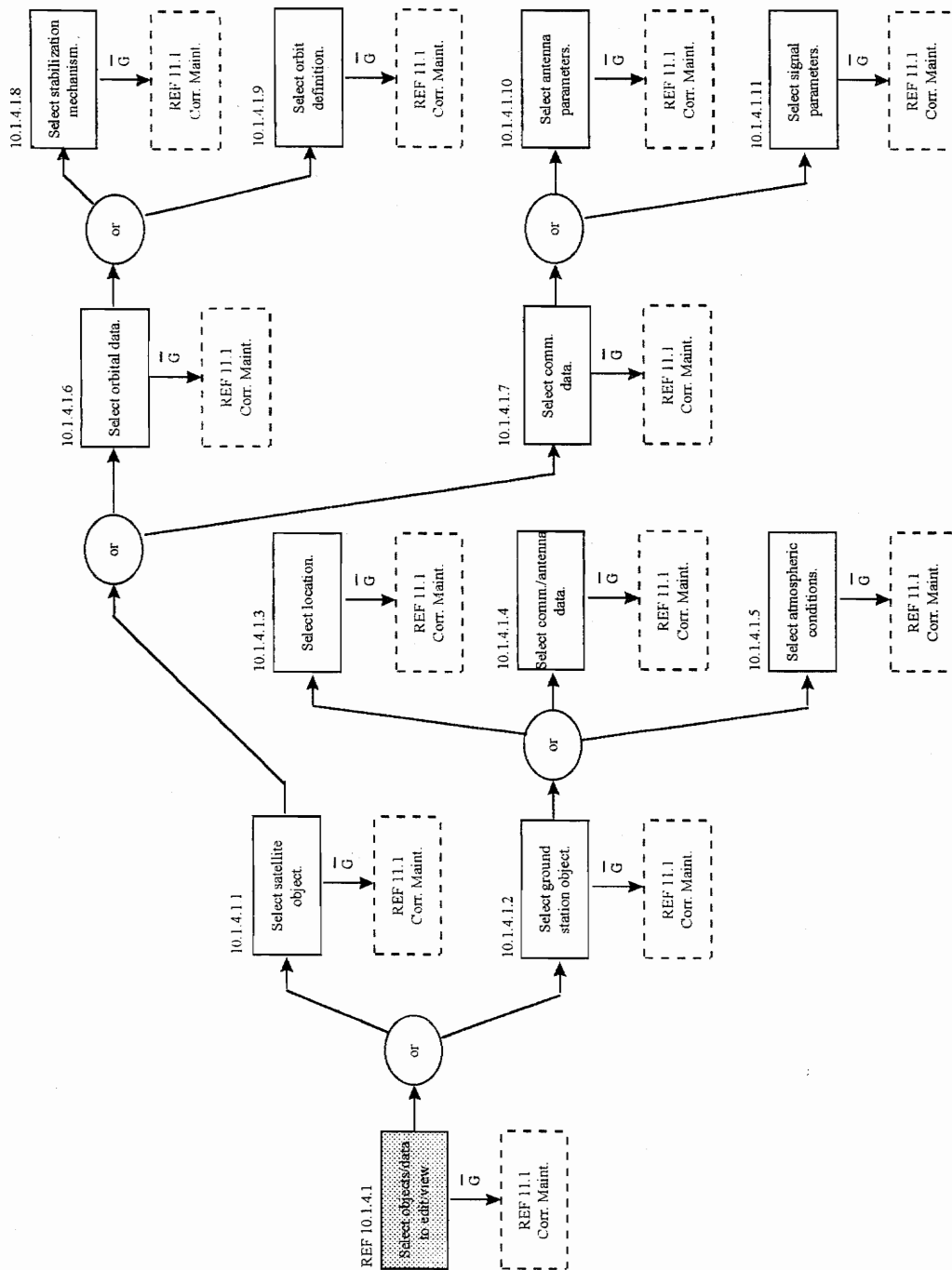


Figure 12: Select Objects/Data to Edit/View (FFD)

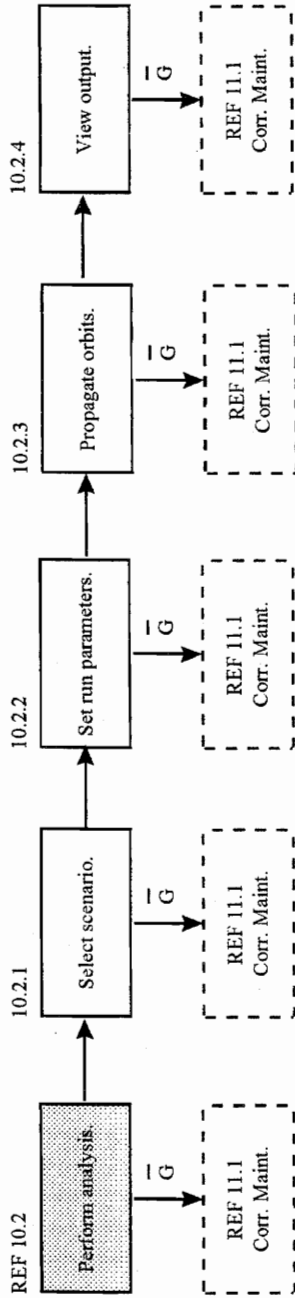


Figure 13: Perform Analysis (FFD)

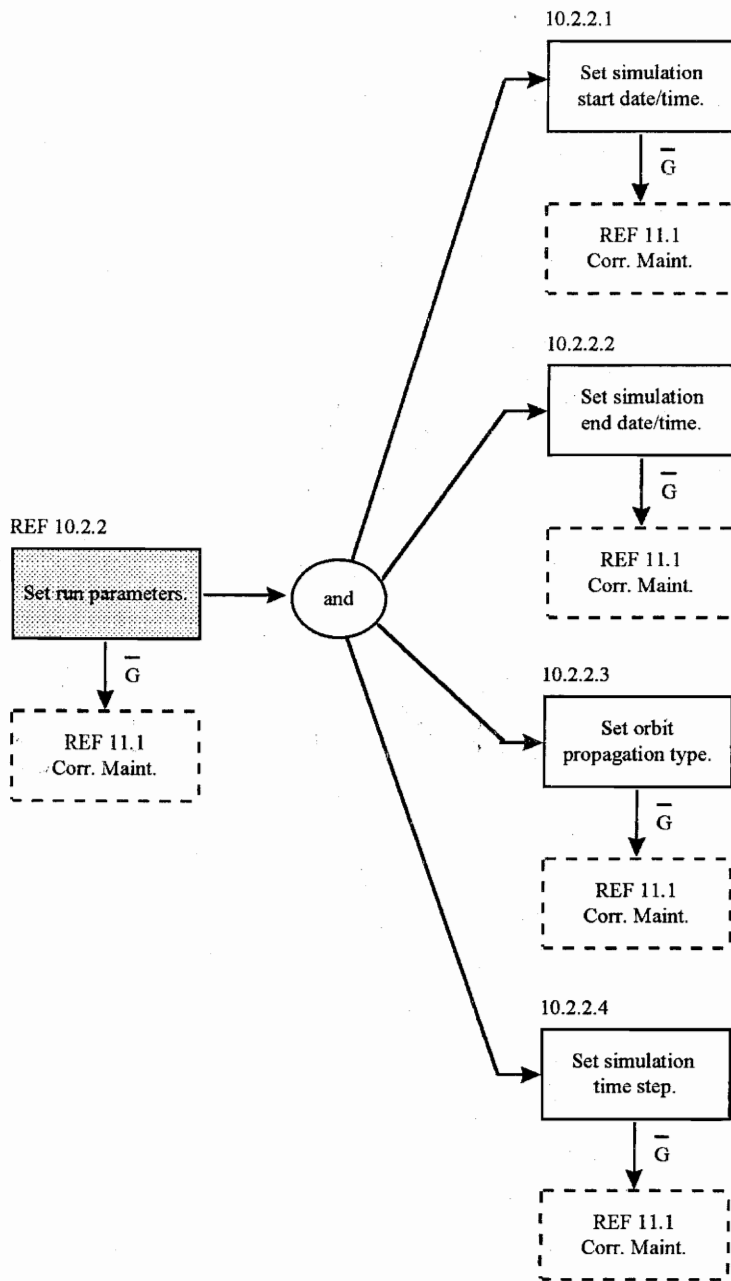


Figure 14: Set Run Parameters (FFD)

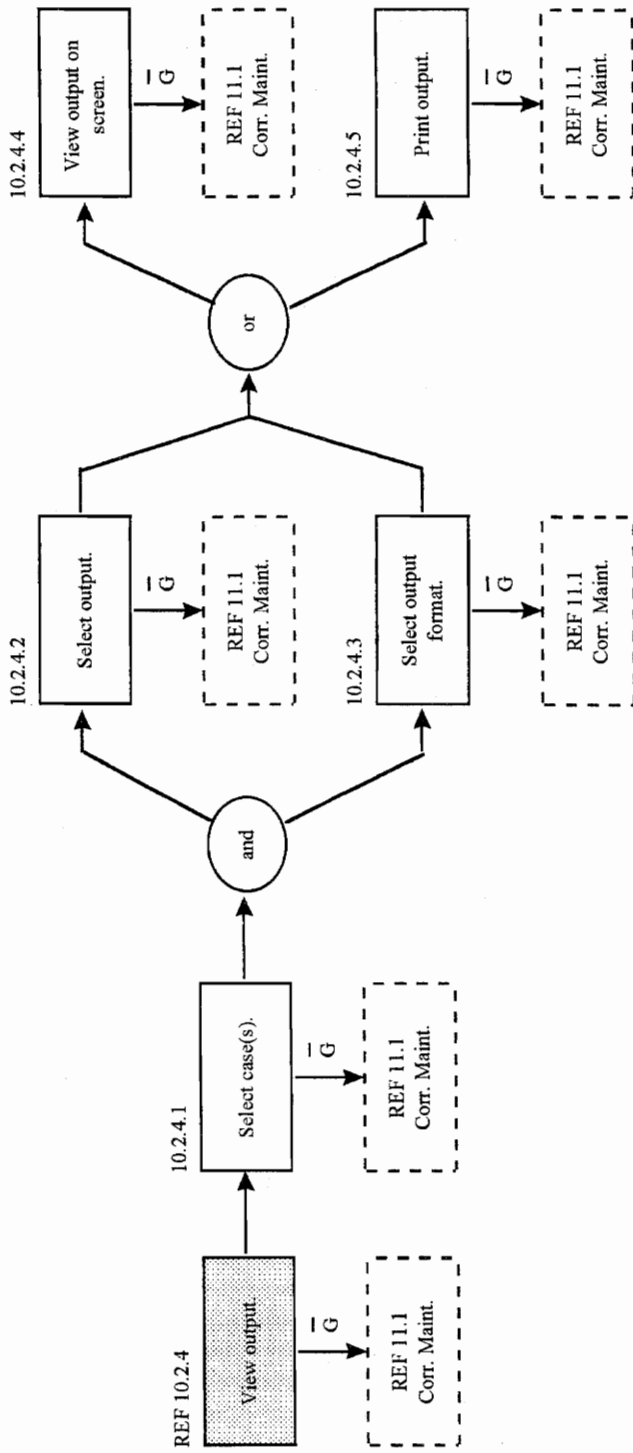


Figure 15: View Output (FFD)

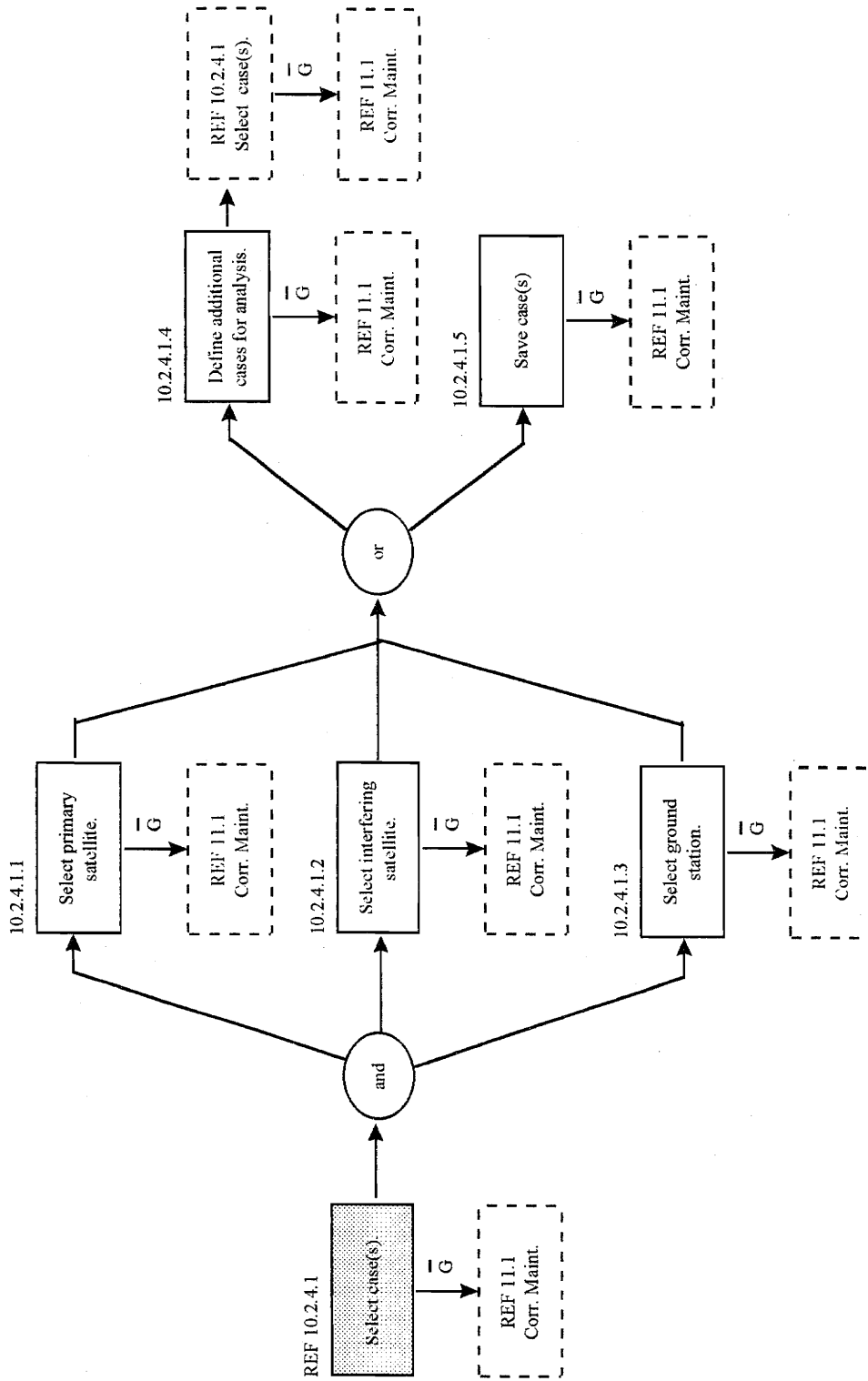


Figure 16: Select Case(s) (FFD)

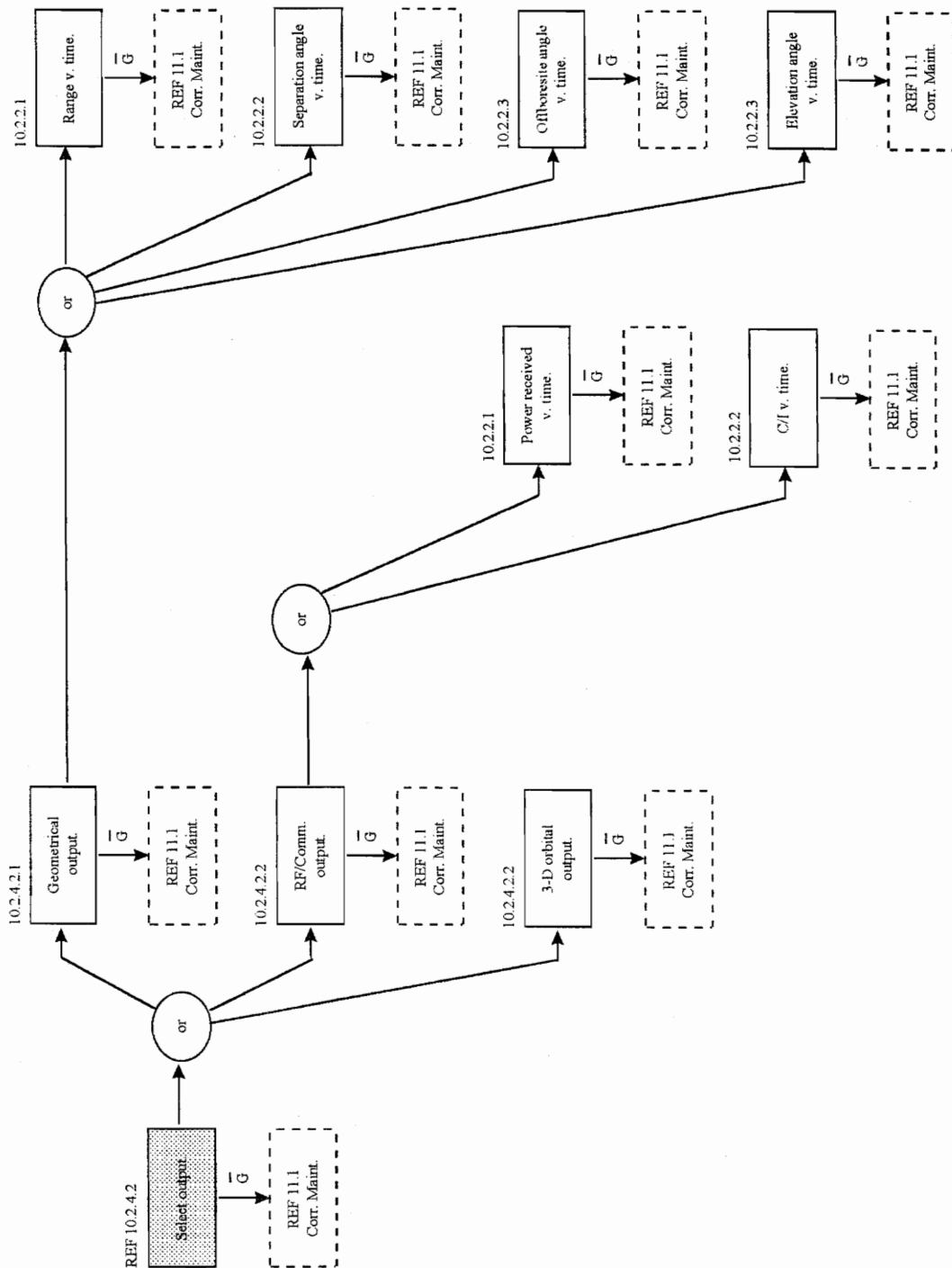


Figure 17: Select Output (FFD)

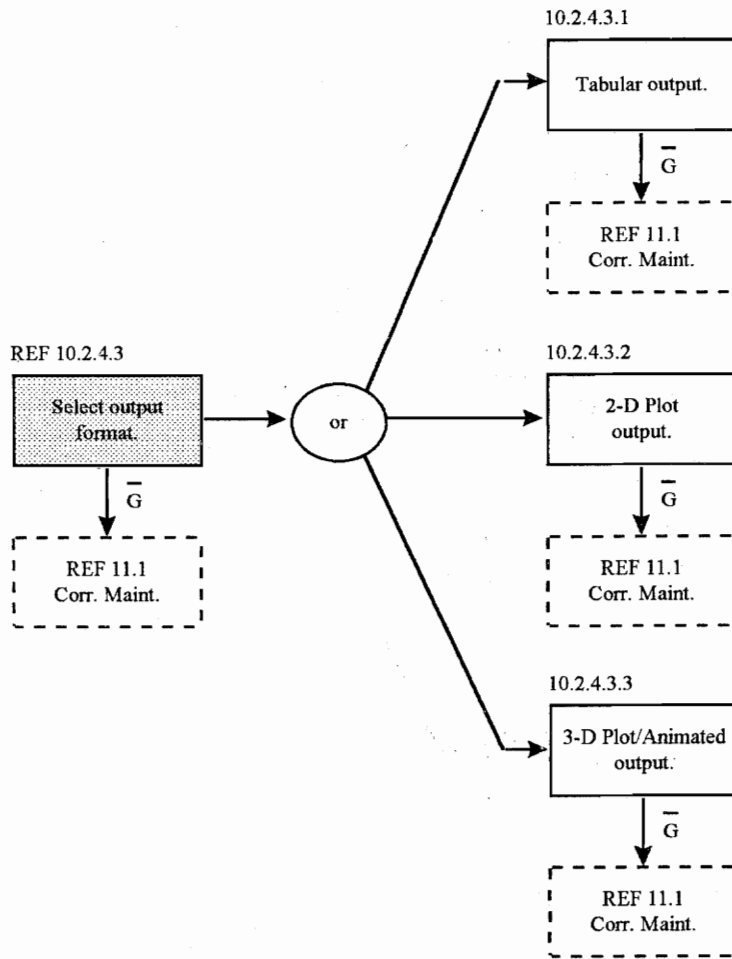


Figure 18: Select Output Format (FFD)

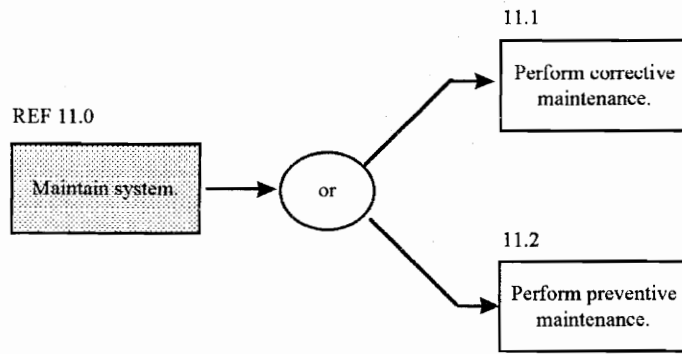


Figure 19: Maintain System (FFD)

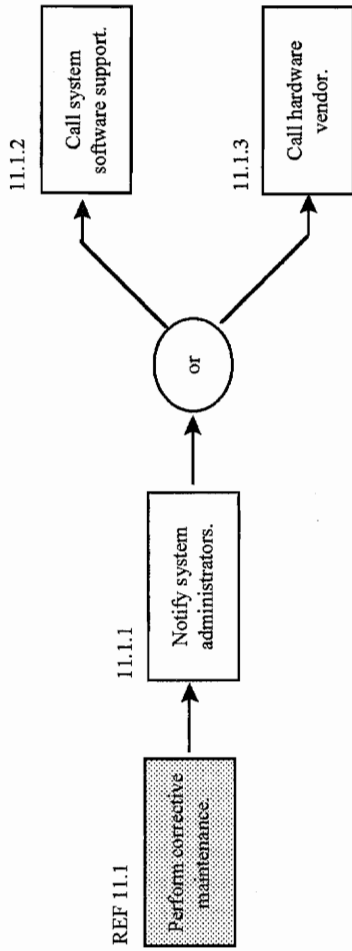


Figure 20: Perform Corrective Maintenance (FFD)

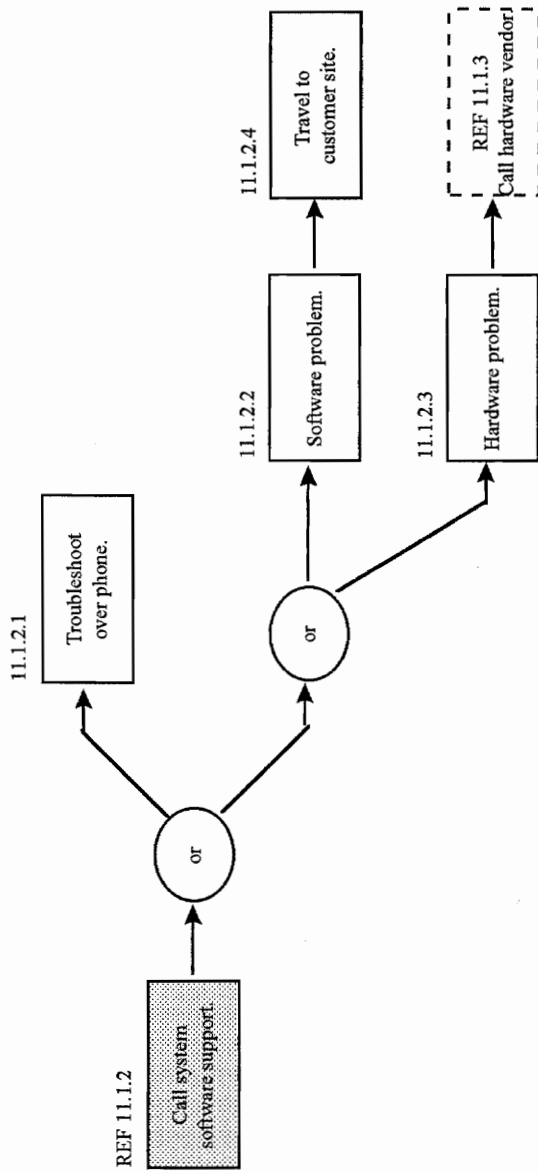


Figure 21: Call System Software Support (FFD)

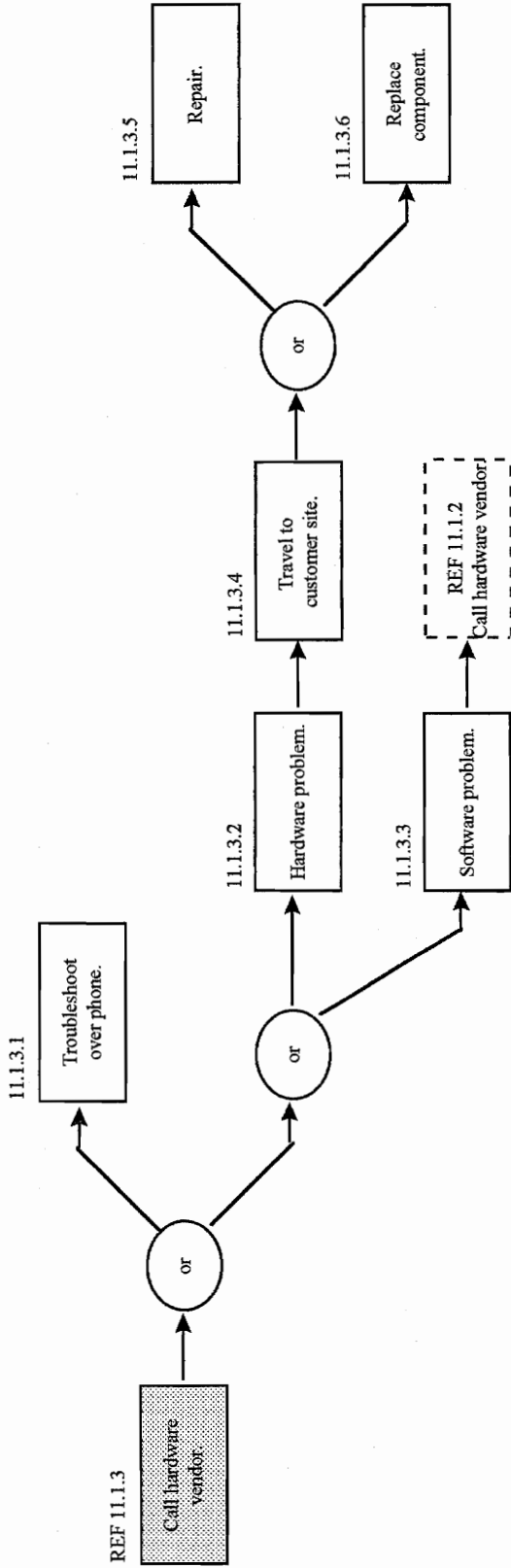


Figure 22: Call Hardware Vendor (FFD)

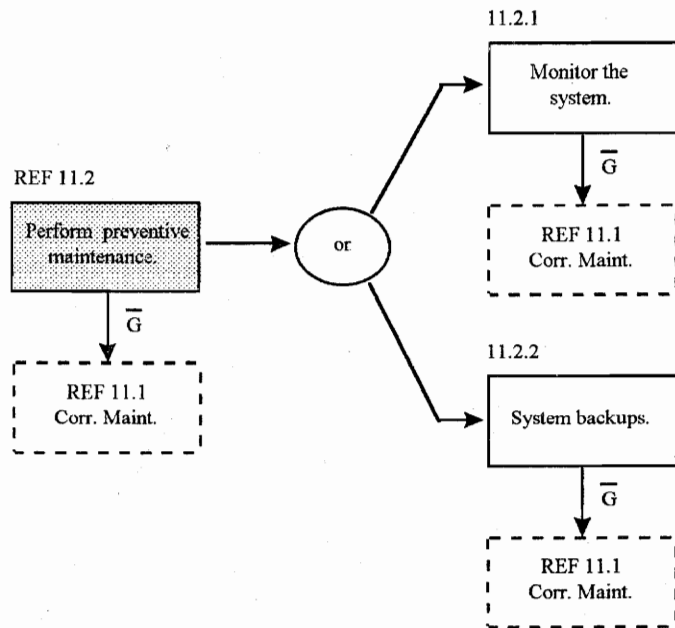


Figure 23: Perform Preventive Maintenance (FFD)

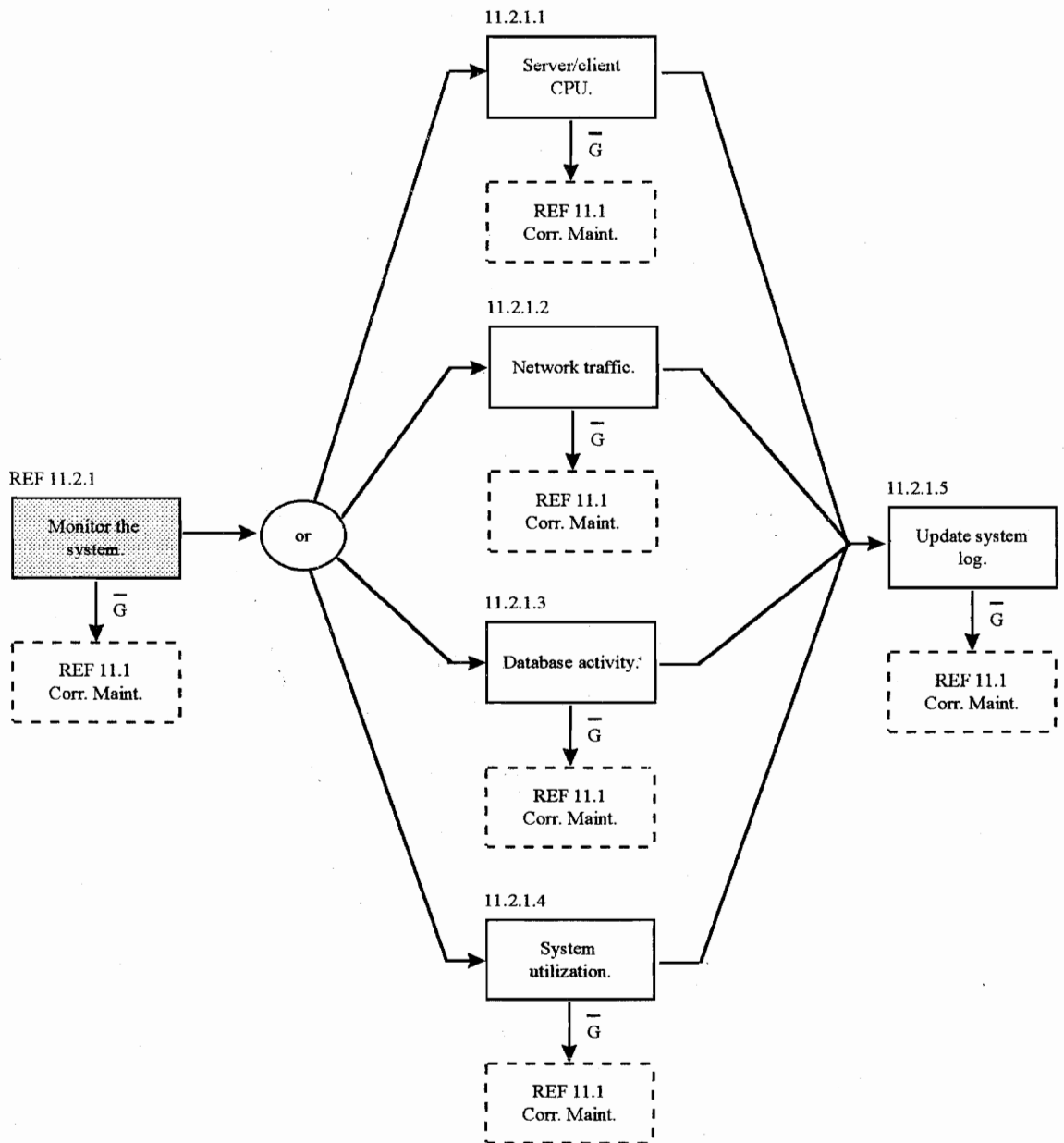


Figure 24: Monitor the System (FFD)

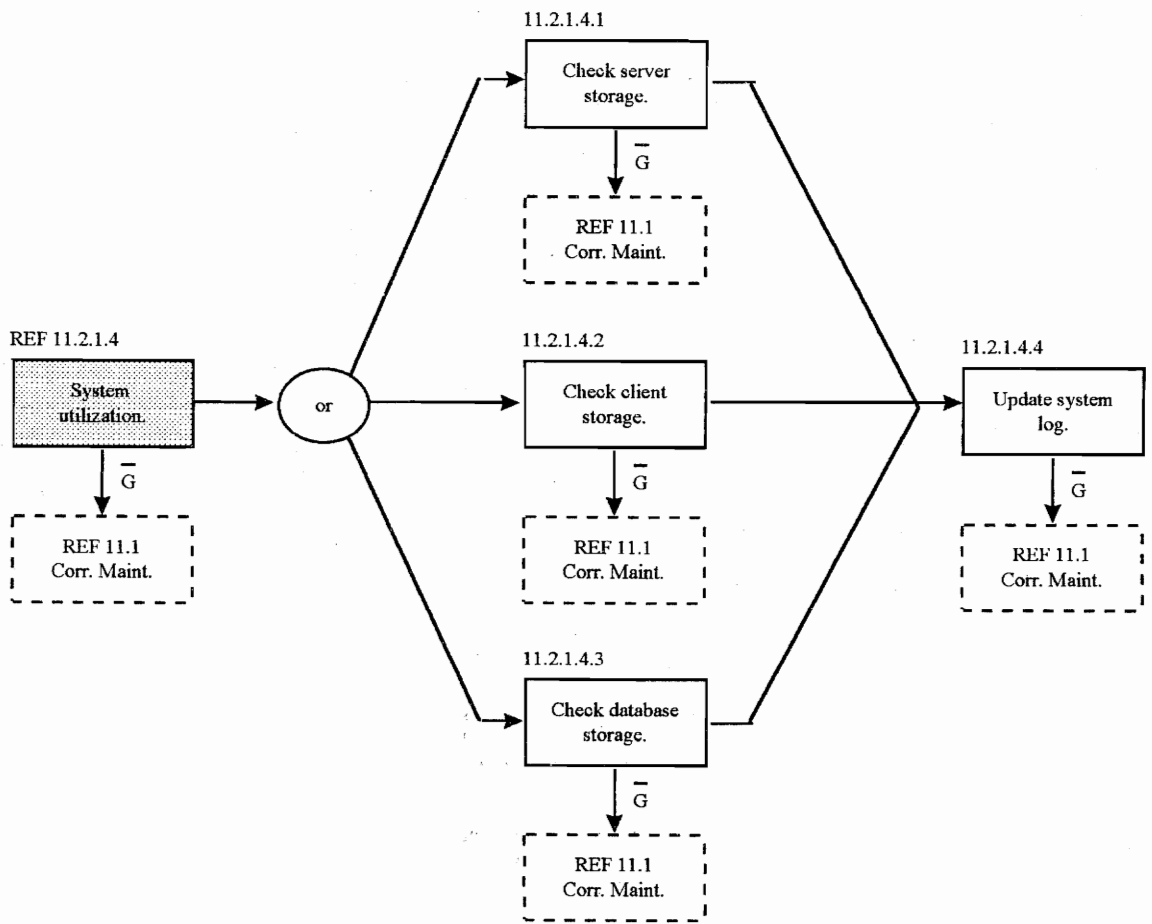


Figure 25: System Utilization (FFD)

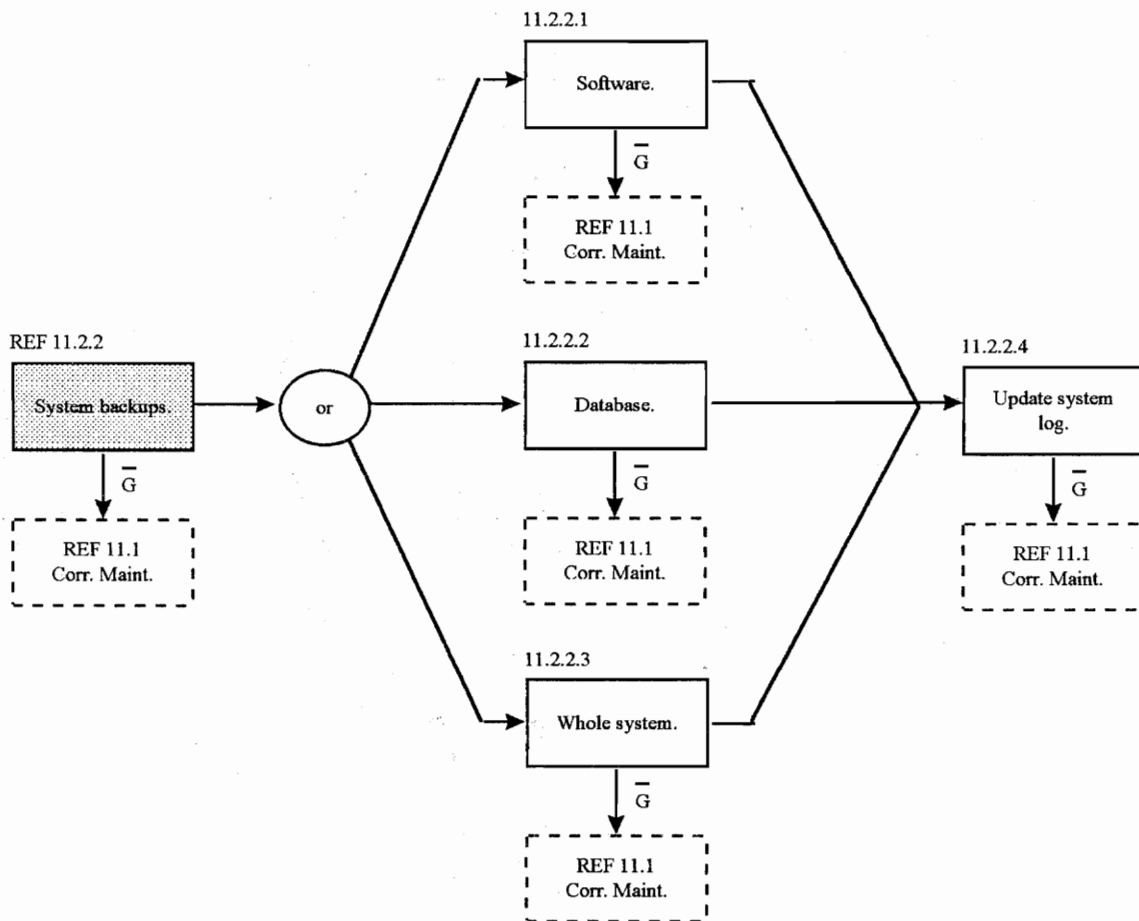


Figure 26: System Backups (FFD)

Allocation of Requirements

Once the functional operational and maintenance flows are defined for the system, a packaging scheme must be developed and requirements must be allocated to the various system components. Requirements are allocated to subsystems to ensure the system level requirements are met.

In order to allocate requirements for the DLTool System, the system hierarchy, or packaging scheme, must first be established and key performance or effectiveness factors must be identified. Components of the DLTool System were identified by packaging functions into logical groupings. In completing the packaging scheme, two levels of subcomponents for the DLTool System were established. The primary components are hardware, software, and database support. Each of these components has subcomponents as well, as shown in Figure 27.

The key factors for allocation of requirements are operational availability (A_0), MTBF, and cost. Figure 27 shows the hierarchy of the DLTool System and the breakout of key performance and effective factors into the major system components. Key performance factors of the subcomponents combine to support the top level, overall system requirements.

The costs of the DLTool System in the figure are primarily the labor for software development, integration, test, and installation. Thus, the costs associated with the database and hardware components of the DLTool System are for maintenance only and

therefore are very small. Also note that maintenance costs, especially for hardware, are shared across the multiple systems that the hardware supports.

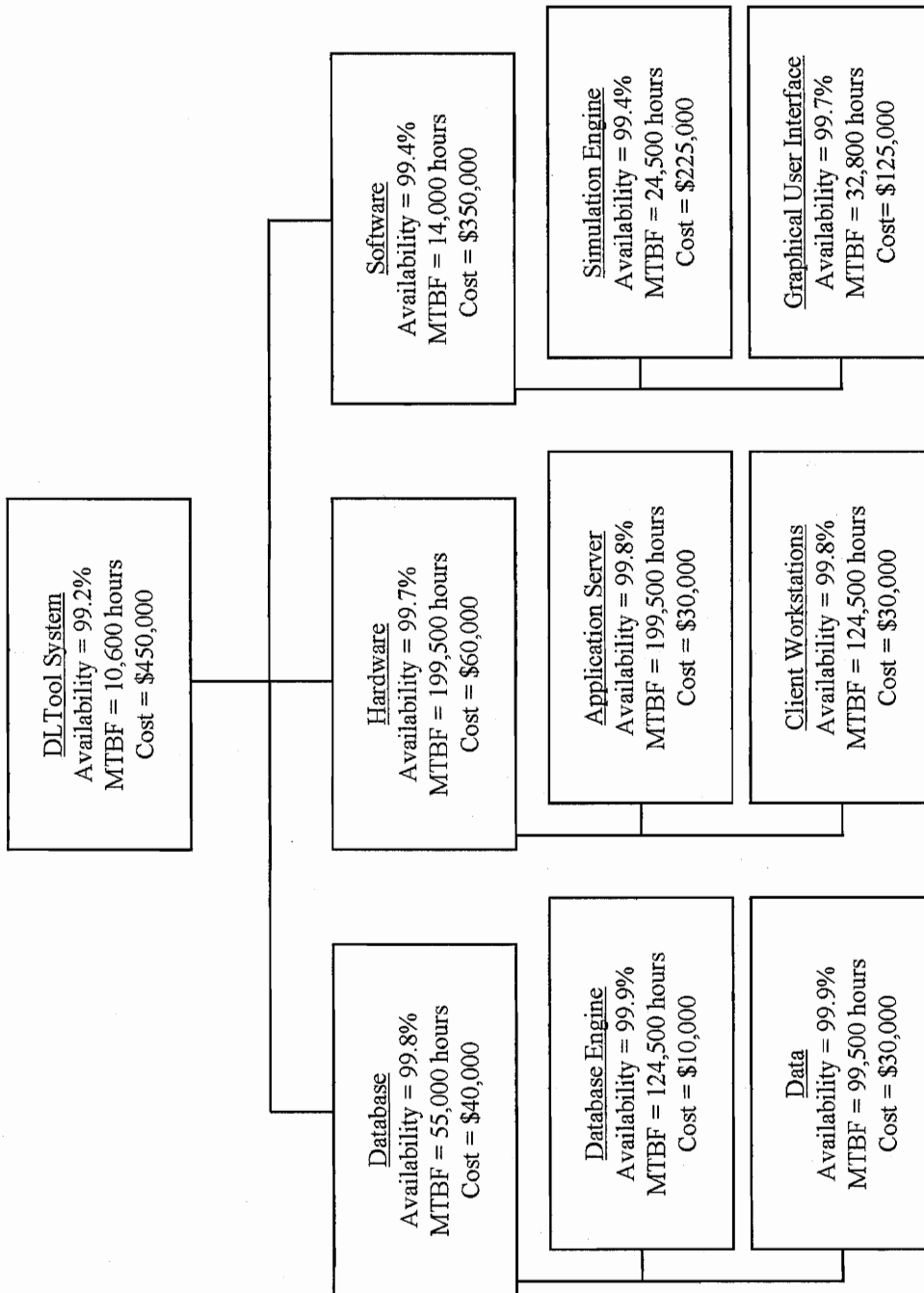


Figure 27: Work Breakdown and Allocation of Requirements

Trade-off and Optimization

In the trade-off and optimization phase of the systems engineering process, design alternatives are identified and evaluated. Configurations are considered within the context of the requirements previously described for the system. [1] Within the DLTool System, four technical components were studied to determine the optimum configuration:

- Computer workstations
- Software language
- Relational database support
- Graphical user interface package

Decisions were based on cost, performance factors, and technical support considerations.

Upon conclusion of the trade-off and optimization analysis, the following design recommendations were made:

- Workstation network of Sun/Unix SPARCStations
- Custom built, object oriented C++ software
- ORACLE database support
- Motif compliant GUI built with U/IMX development package.

Each of these components is discussed in detail in the following sections.

Computer Workstations

Various vendors such as IBM, DEC, and Sun produce computer workstations that would satisfy the requirements of the DLTool System. Performance of these machines is comparable and thus cost and support become the primary determinants. Program 607 at Lockheed Martin Corporation already uses a network of Sun SPARCStations. The actual models range from the slower LX to the top of the line SPARCStation 20. Each of these workstations has the minimum RAM, processor clock speed, and disk space specified by the required physical parameters. In addition, Program 607 already has a maintenance contract with Sun Microsystems for hardware support. Since each workstation costs approximately \$15,000 to \$20,000, using the existing Sun SPARCStations for the DLTool System is a very cost effective solution that does not sacrifice performance. Due to the hardware in place in Program 607, a Unix-based Sun SPARCStation network is recommended.

Software Language

To construct the DLTool System, custom built software will be required. Although COTS products do exist that may be integrated to perform some of the functions required by Program 607, a system composed of only integrated COTS products does not fill the needs of the program.

The primary options for software language are C, C++, FORTRAN 77, or SimScript (or other simulation language). Each of these languages provides advantages. SimScript, and other simulation languages, contain internal functions that support many

standard simulation functions such as queuing and statistics. However they are not as portable, nor as well known. Thus, flexibility and maintenance become a problem.

FORTRAN is a very well established scientific language. It is very fast, but relatively dated. It does not provide some of the advanced features of modern languages such as dynamic memory allocation and support to graphical user interfaces. C and C++ are well established, like FORTRAN, yet more modern and flexible. They provide advanced features and are portable across platforms. The C family of languages are considered relatively fast, and development packages exist to facilitate the coding work. Many software development programs use C or C++.

The difference between C and C++ is the object oriented (OO) nature of C++. Object oriented software flows from object oriented design. Thus, OO is an entire methodology that has become very popular in software development. Typically, OO software is very modular which promotes reuse and ease in maintenance. Lockheed Martin has several large development programs in Valley Forge that require C++ programmers. Thus, there is a skill base available to support development of software in C++. These advantages make C++ the recommended development language for the DLTool System.

Relational Database Support

In the feasibility study for the DLTool System, a database program was recommended to manage and control the data used by the system. Although a database program could be developed to support the DLTool System, there are numerous COTS

database applications available to integrate into the system. Due to the complex nature and high cost of building a custom database as well as the strength of the existing COTS products, integration of an existing database engine into the DLTool System is highly recommended.

Two popular providers of COTS database software are ORACLE and Sybase. Other smaller database applications are available such as Access or FoxPro, but they do not suit the size and complexity of the need in the DLTool System. ORACLE and Sybase are both relational database engines of comparable speed. They are both Sequential Query Language (SQL) compliant as well. However, Lockheed Martin has already purchased ORACLE licenses and support for other purposes and has some ORACLE experience as well. Therefore, ORACLE is the most cost effective alternative for database support and is recommended for the data storage function of the DLTool System.

Graphical User Interface Package

The most common graphical windowing environment for Unix-based workstations is OpenWindows/Motif. Additionally, this is the windowing environment already used by Program 607 on their Sun SPARCStations. Motif will allow development of a graphical, menu-driven interface that provides panels for data input and graphical output. Motif provides the functionality required and is cost effective. Thus, a Motif GUI is recommended. Several COTS products are available to support GUI development for Motif compliant clients. Lockheed Martin currently holds licenses for a GUI development package called U/IMX. U/IMX helps developers build interface panels to accept input

and display output. To facilitate building the GUI, U/IMX is recommended to support the development of the DLTool System.

Synthesis and Definition

The synthesis and definition phase of the design process involved developing typical satellite radio frequency interference scenarios and analyzing the steps required to perform the desired analysis using the DLTool System. Program 607 employees participated in this activity to help identify the strengths and weaknesses of the system design as well as any concerns before going into detailed design of the DLTool System.

Much of the review revolved around the functional analysis performed previously and how these flows would be implemented in the graphical user interface. Since earlier tools used for satellite radio frequency interference at Program 607 were primarily text based, the functionality and user-friendliness of the GUI were significant interests of the potential users. Panel design for the system was discussed and outlined to insure the user's desires were captured.

By stepping through typical scenarios, the modularity and maintainability of the DLTool software also seemed important. Users would like the capability to have pieces of the software easily upgraded or enhanced. One example was the possibility of including multiple orbit propagation tools in the system that provide different levels of fidelity such that the user could define the accuracy of the output.

The synthesis and definition effort emphasized the importance of maintaining user participation throughout requirements definition and design phases of the system. Early and continuous involvement from system users will reduce the number of changes or adjustments later in the systems engineering process and save costs.

CONCLUSIONS

The conceptual and preliminary design of the Dynamic Link Tool System presented outlines the fundamental requirements and design for a system that will support the satellite radio frequency interference analysis needs of Program 607 of Lockheed Martin Corporation for 10 years. Through the systems engineering process, user's needs and available technology were combined to form system requirements and a preliminary design. Throughout the process, potential users of the DLTool System were consulted to insure needs were being met.

Continued progress in the systems engineering process will take the DLTool System through detailed design and development, construction, support, and phase-out activities. By following this path, the system is sure to complete its mission of providing moderate fidelity analysis for its users at an acceptable cost.

Potential future enhancements to the DLTool system include on-line help, tutorials, and a training manual. Each of these improvements would support the use of the DLTool System and possibly reduce support costs associated with helping inexperienced users of the system.

REFERENCES

- 1) Blanchard, Benjamin S. and Wolter J. Fabrycky, Systems Engineering and Analysis, Englewood Cliffs, NJ, Prentice-Hall, 1990.
- 2) Cushman, William H. and Daniel J. Rosenberg, Human Factors in Product Design, Elsevier Science Publishing Company, Inc., New York, 1991.
- 2) Pratt, Timothy and Charles W. Bostian, Satellite Communications, John Wiley and Sons, New York, 1986.
- 4) Stallings, William, Local and Metropolitan Area Networks, Fourth Edition, MacMillan Publishing Company, New York, 1993.
- 5) White, John A., Marvin H. Agee and Kenneth E. Case, Principles of Engineering Economic Analysis, John Wiley and Sons, New York, 1989.

APPENDIX A: LIFE-CYCLE COST ANALYSIS

Life-cycle cost is an important factor when considering any new system. Not only should development costs be considered, but maintenance, upgrade, test, and disposal costs should be included when discussing system alternatives and their economic feasibility. In many cases, the development cost of a system is a small percentage of the other costs, such as maintenance and testing. Throughout the life cycle cost analysis, several assumptions were made:

- Lockheed Martin Corporation's MARR is 12%
- Taxes and depreciation are ignored
- No salvage values are expected
- Average hourly labor rate is \$50.00.

This life-cycle cost analysis covers all costs for the DLTool system, from development to disposal. The major cost for the DLTool System, due to the decision to leverage existing hardware within the company, are labor costs. In order to determine the labor costs, a high-level lifetime schedule for the DLTool System was built. The schedule identifies the timelines for the major activities in the life of the DLTool system. The schedule is shown in Figure 28.

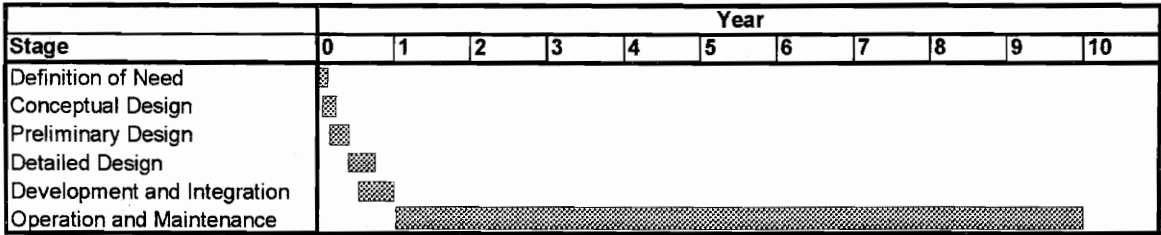


Figure 28: DLTool System Lifetime Schedule

Based on this schedule, the approximate number of man-hours required during each year of the DLTool lifetime may be determined. A summary of the required man-hours is shown in Figure 29.

Stage	Year											Total	
	0	1	2	3	4	5	6	7	8	9	10		
Definition of Need	200												200
Conceptual Design	600												600
Preliminary Design	900												900
Detailed Design	1400												1400
Develop. and Integ.	2000												2000
Maintenance		100	100	100	100	100	100	100	100	100	100	0	900
Total Man-hours	5100	100	100	100	100	100	100	100	100	100	100	0	6000

Figure 29: DLTool System Labor Man-hours by Year

Using the average hourly labor rate, the man-hours may be converted into dollars to produce the cash flow for the lifetime of the DLTool System. The cash flow is shown in Figure 30. Note that all currency values are in today's dollars.

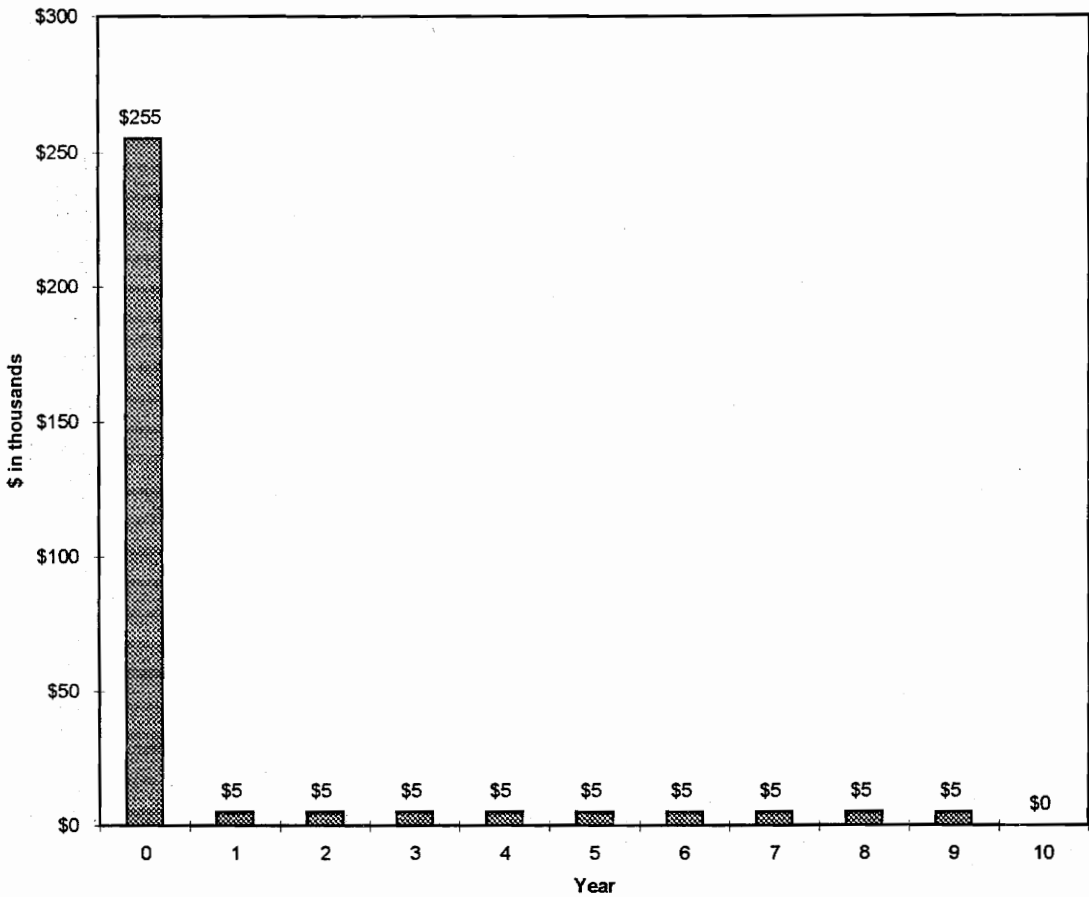


Figure 30: DLTool System Lifetime Cash Flow

In addition to yearly cash flow, costs should be attributed to the various functions identified in the functional analysis of the preliminary design. Typically, the traceability of costs to system functions is accomplished using a Cost Breakdown Structure (CBS). Functional costs of the DLTool System should be investigated and developed further for completeness. However, from the high level functional analysis previously completed, development of the graphical user interface functions appears to be a large driver of cost.

In order to compare the DLTool System to the existing system to perform satellite radio frequency interference analysis, a break-even analysis was prepared. The break-even analysis is based on the premise that four complete simulation runs are required per week. On average, the existing system needs 6 hours to complete the analysis while the DLTool System needs only 1 hour. Figure 31 shows the break-even analysis for the complete costs of developing, maintaining and operating both systems.

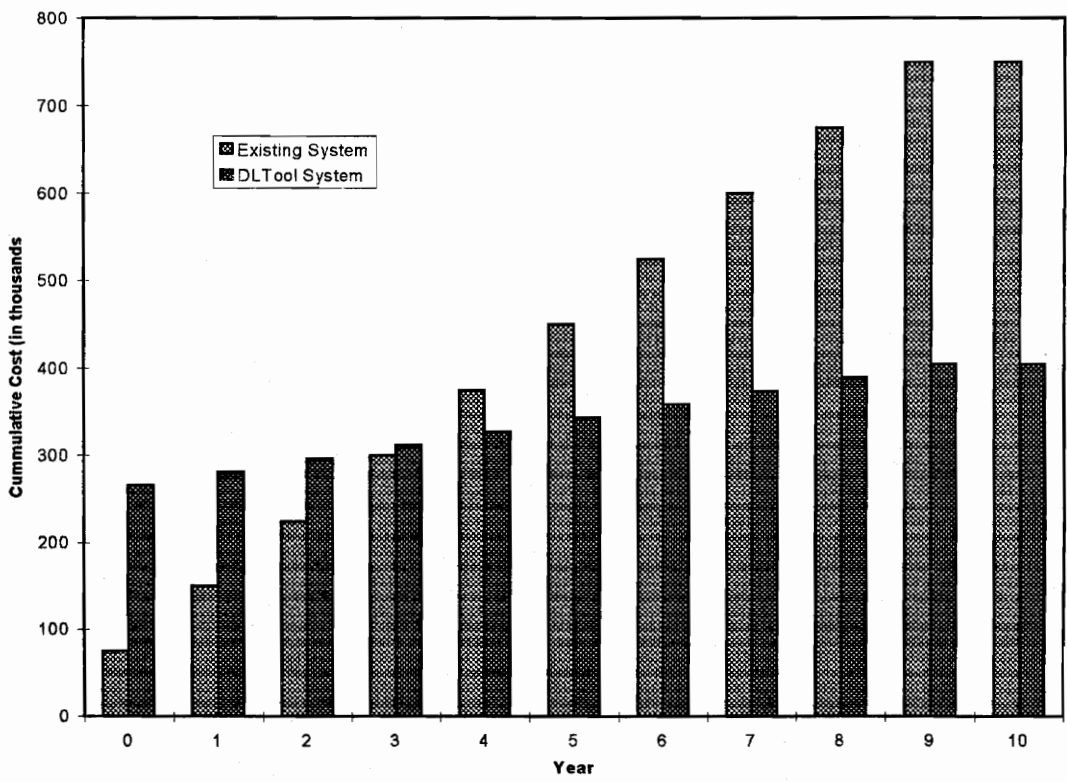


Figure 31: Break-even Analysis For Satellite RF Interference Analysis Systems

From Figure 31, the DLTool begins to recoup the heavy initial investment by year 3. The final comparison between DLTool System and the existing system is the net present worth of each. Using the cash flows for each system and a 12% MARR, the DLTool System has a net present worth of (\$311,000) while the existing system has a net present worth of (\$424,000). Thus, based on the break-even analysis and present worth calculations, the DLTool System is a sound investment for Lockheed Martin Corporation and Program 607.

APPENDIX B: SYSTEM RELIABILITY ANALYSIS

Reliability is the probability that a system will perform to a satisfactory level for a specified amount of time. Reliability is a critical parameter and measure during the design process, as well as throughout the systems engineering cycle. [1]

The reliability of the system is a function of the reliability of the system components. The reliability of an individual component may be computed by the following relationship:

$$R = e^{-t/MTBF}$$

where R is the reliability expressed as a percentage and t is the time over which the percentage is valid. [1] Taking t to be 1000 hours, the following reliabilities may be computed for the system components defined in Figure 27.

Component	Reliability
<i>Database</i>	98.20%
Database Engine	99.00%
Data	99.20%
<i>Hardware</i>	99.50%
Application Server	99.50%
Client Workstations	99.80%
<i>Software</i>	93.10%
Simulation Engine	96.00%
GUI	97.00%

Figure 32: DLTool Component Reliabilities for t=1,000 Hours

Figure 32 shows the component - subcomponent relationships in the DLTool System as well. As an example, the database engine and data are under the more general component, database. The reliabilities in Figure 32 contribute to an overall system reliability of 91.0% over 1,000 hours of use.

Subcomponents must be assigned reliabilities such that the combination of subcomponents meets the requirement of the parent component. As an example, consider the reliabilities of the three major components: database, hardware, and software. Each of these contribute to the overall reliability of the system. Since each component must be working for the system to operate, the components are said to be in series. The reliability of a series is simply the product of the reliabilities of its components. So, for the DLTool System, the following relationship is true:

$$R_{system} = (R_{database})(R_{hardware})(R_{software})$$

Multiplying the three component reliabilities together yields the overall system reliability of 91.0% for 1,000 hours.

APPENDIX C: ACRONYM LIST

COTS	Commercial Off The Shelf
DLTool	Dynamic Link Tool
FFD	Functional Flow Diagram
GUI	Graphical User Interface
LAN	Local Area Network
LMC	Lockheed Martin Corporation
Mct	Mean Corrective Maintenance Time
MDT	Maintenance Downtime
Mpt	Mean Preventive Maintenance Time
MARR	Minimum Average Rate of Return
MTBF	Mean Time Between Failure
OO	Object Oriented
R	Reliability
RAM	Random Access Memory
RF	Radio Frequency
SQL	Sequential Query Language