SYSTEM DESIGN OF A DISCREPANCY REPORTING SYSTEM

by

Frank Michael Pilewski

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:

B. S. Blanchard, Chairman

L. K. Harmon

T. E. Richbourg

May, 1994

Blacksburg, VA

SYSTEM DESIGN OF A

DISCREPANCY REPORTING SYSTEM

by

Frank Michael Pilewski

Committee Chairman: B. Blanchard

Systems Engineering

(ABSTRACT)

The preliminary system design of a discrepancy reporting (DR) system is presented. A customer desires a softcopy discrepancy reporting system for the tracking and storage of these documents. The system would replace the need for an eight to ten person group to track and store these documents in hardcopy form. The preliminary design includes the Conceptual Design with a feasibility study where operational and maintenance requirements have been established, and the Preliminary Design where requirements have been further defined through the use of flow diagrams.

The system is designed to be accessible by a user to generate discrepancy reports against software, hardware or database problems at the user's location. The discrepancy reports are stored in a relational database with a client/server architecture. The DR administrator has the same capabilities as the user, plus the ability to edit the database.

ACKNOWLEDGEMENTS

I wish to acknowledge the help of Professor Harmon, Dean Blanchard and Tom Richbourg in the preparation in this report. I would like to acknowledge the help of Program 212 for their support, time and patience for making this paper possible. I acknowledge the support of my parents over the past twenty-six years. Without their supervision and support, there may have been no drive to attempt this effort. Finally, I wish to acknowledge the support of my fiancee, Kelli Levins. She shared every step of this experience with me, and always had a word of encouragement when one was needed most. Without her, there would have been no need to attempt this project.

TABLE OF CONTENTS

TABLE OF CONTENTS (CONTINUED)

TABLE OF FIGURES

TABLE OF FIGURES (CONTINUED)

TABLE OF TABLES

INTRODUCTION

Background

Program 212 of Martin Marietta is a small contract, currently operating in the Northern Virginia area, that provides systems engineering services and software and engineering development support for Configuration Management tools. The contract was awarded to General Electric Aerospace, now Martin Marietta, in February of 1991. The contract included a prototype development effort for a Discrepancy Reporting (DR) system, with a follow-on contract for system maintenance. The Discrepancy Reporting system consists of customized software which uses a vendor database package and is built to run on specific hardware and operating system platforms. The system, named DRTool, allows users to enter, store and track hardware, software or database problems or discrepancies at the user's business location. The development effort was completed in March 1993, and the system was delivered and installed at three operational sites. Although the DRTool contract was initially started as a prototype effort, the DRTool system is currently being maintained as an operational system.

The current DRTool system consists of a client/server environment with the following components:

- Mega Drive Systems Database Management System (DBMS)

- Sybase server for the database interface
- Sun SPARCstation 2 workstations
- Sun OS 4.1.3 operating system
- Program 212 customized code with Structured Query Language (SQL) statements embedded in C code
- User interface panels developed using the Openlook version of X-Windows

The prototype effort was initiated to provide a softcopy DR system for writing, storing and tracking classified software, hardware and database problems. The initial cost of the system was justified by the fact that the system would replace the 8-10 people normally required to track and file hardcopy DRs at the user's location. By replacing the current hardcopy system, the salaries and security clearance costs of the employees are saved. The softcopy system also saves paper, printing costs for the hardcopy DR forms, and storage space since hardcopy DRs must be stored in file cabinets.

The DRTool system is currently being demonstrated to potential users in an effort to win new business. The effort has resulted in a fourth site becoming operational in early 1994, with a fifth site expected to become operational in the second quarter of 1994. While Program 212 has expanded its business opportunities within one year of the DRTool delivery, the satisfaction level of the original customers has declined. The customer is unhappy with the operational availability of the DRTool system. The system is required to support an operational availability of 98% or better, but several system failures have caused the system to be unavailable for days. One location had two system failures in one week, which resulted in six days of down time and a large loss of data (113 DRs). The current system also fails every time 110 DRs are entered into the system due to memory management problems within the specialized application code. This requires a higher degree of preventive maintenance than the one hour per week that was expected by the customer. One location complained that 5 out of 15 main user options, including searching, sorting and reporting, did not work correctly. Prospective users have requested to see the maintenance concept for the system, but one does not exist.

It was mentioned that Program 212 was in danger of losing funding for the operations and maintenance efforts of the DRTool system. This would have adverse impacts on all future, small systems contracts pursued by Martin Marietta. Since the DRTool system was originally developed as a prototype system, several aspects of the systems engineering life cycle were not incorporated into the system design. Due to the large number of system problems and new user requirements, the DRTool system must be designed from scratch using the systems engineering life cycle approach to establish operational and maintenance requirements.

Objective

The objective of this project is to engineer the Conceptual System Design and Preliminary System Design for the DRTool system. The current system operational requirements are vague and open to a large amount of interpretation. The system maintenance requirements are not currently defined or documented. These requirements must be defined before the Preliminary System Design and Detailed Design and Development can begin.

Scope

The scope of this project is limited to the initial systems engineering design of Martin Marietta's DRTool system. Since most complaints by users are directed at the functionality of the DRTool system, it seems that the requirements must be defined in enough detail so that the developers and users obtain the same understanding of the functionality of the system. Once the requirements are formally defined, the development, maintenance and testing of the DRTool system will be easier and more thorough. This will ultimately lead to a higher degree of customer satisfaction.

The time line for this project is tight due to the declining level of customer satisfaction. The customer will be providing a monetary award fee to Martin Marietta on 29 April 1994 based on their level of satisfaction, and this project will be started immediately to illustrate Program 212's commitment to customer satisfaction.

SYSTEM REQUIREMENTS

The initial steps of the Systems Engineering Life cycle are applied to the Martin Marietta DRTool system design. These steps include:

1. Definition of Need
2. Conceptual System Design
3. Preliminary System Design
4. Detail Design and Development [2]

This project deals directly with the Conceptual System Design and Preliminary System Design of Martin Marietta's DRTool system. The requirements from the Conceptual Design and Preliminary Design will be carried forward into the Detail Design and Development phase of the Life Cycle.

Definition of Need

The definition of need refers to the identification of the reason Program 212 of Martin Marietta is undertaking the DRTool system design project. The original DRTool system need was investigated to determine if the need was still valid. Due to Department of Defense budget cuts, the customer desired a softcopy storage and tracking system for DRs, thus eliminating the need for individuals to store and track hardcopy DRs. The hardcopy DRs required large amounts of paper, storage space and individuals' time to ensure the DRs were stored correctly and routed to the appropriate individuals that were responsible for investigating a DR. A hardcopy DR would take 2-3 days to reach the individuals responsible for investigating the problem cause, and this would often result in the loss of data necessary to troubleshoot the problem. The softcopy system would provide faster data retrieval, storage and routing than the hardcopy DR system. The softcopy system would require the following components to support the DRTool system:

- A database to store the DR data
- Computer terminals for entering and displaying data
- User input panels resembling hardcopy DR forms
- Specialized application code to route and/or retrieve the DR data to/from the database

The initial cost was limited to be less than the sum of the yearly salaries and security clearance costs of the 8-10 individuals normally required to support the hardcopy DR system. This required that the initial costs of the softcopy DRTool system be less than $340,000.

The following high level system requirements were required by the current customer and new users. These requirements are generic guidelines that were later expanded upon to determine operational and maintenance requirements.

The DRTool system shall:

- Have an operational availability greater than 98%
- Have a reliability greater than 90% for 1000 hours of use
- Handle classified data using C2 level access controls with no possibility of security contamination of other data
- Store all DRs since the system installation
- Be designed for 7 years of operational life
- Have access controls preventing users from deleting the operational software or system initialization data
- Store up to 15,000 open DRs
- Have a maximum retrieval time of 9 seconds for one open DR
- Have a maximum retrieval time of 20 seconds for one closed DR when the number of closed DRs is less than 100,000. An additional 10 seconds would be required to support each additional 100,000 closed DRs stored in the system.
- Have menu driven Graphical User Interface panels
- Have an established maintenance concept
- Have an established training manual
- Have an established user's manual
- Track DR status

- Assign unique sequential DR identifiers for each new DR entered into the system

The current DRTool system developed as a prototype by Program 212 supports only seven of the fifteen requirements that were requested by the customer. The system is required to support an operational availability of 98% or better, but several system failures have caused the system to be unavailable for days. One location had two system failures in one week, which resulted in six days of down time and a large loss of data (113 DRs). One location had the operational software deleted three times in the same week, causing 3 days of system down time. The current system also fails every time 110 DRs are entered into the system due to memory management problems within the specialized application code. This requires a higher degree of preventive maintenance than the one hour per week that was expected by the customer. Trouble was experienced adding new users to the system, and system backup requirements were never established, resulting in the loss of three to seven days worth of data when system failures occur. One location complained that 5 out of 15 main user options, including searching, sorting and reporting, did not work correctly. Prospective users have requested to see the maintenance concept for the system, but one does not exist. Due to the large number of system problems and new user requirements, the DRTool system must be designed from scratch using the systems engineering life cycle approach to establish operational and maintenance requirements.

It was required that the system design project be complete by the next customer award fee date, 29 April 1994. This would illustrate Program 212's commitment to high quality and customer satisfaction. To meet this date, the five person team of Program 212 dedicated time toward the design effort. Customer support was required, and the project received a positive response by the customer management since long term costs would be reduced by the short term expenditure of resources. The Program 212 team had just finished a system enhancement activity, and this project was scheduled to be completed prior to the next development effort.

Feasibility Study

A feasibility study was performed to determine the best applications for supporting the high level requirements that were stated in the definition of need. While performing the feasibility study, Program 431 of Martin Marietta was consulted since this program developed and maintains a DR system. The DR system at Program 431 uses IBM databases, IBM terminals and the FORTRAN and Clist programming languages, but recent efforts to move to a client/server environment have started. This provided some insight from a mainframe and client/server perspective into performance characteristics, maintainability, and overall system effectiveness. The results of the feasibility study indicated that the following applications should be used to support the softcopy DRTool system.

- A client/server environment
- A relational database management system (DBMS)
- A back-end server for database access and client/server support
- Specialized C software code
- User panels designed with a Graphical User Interface (GUI) package

These main components of the DRTool system are discussed to illustrate considerations that were made during the feasibility study.

Client/Server Environment

The client/server architecture is an approach to software in which one application (the front-end client) asks for and receives services from another application (the back-end server). The two applications run independently of each other and a software upgrade in one does not require an upgrade to the other. The major characteristics of the client/server architecture include the logical separation of the client processes and the server processes, the operation of each on separate machines, and the ability of the server to support multiple clients simultaneously.[1] Both technical and business reasons have caused the evolution of client/server architecture. From the business perspective, downsizing has been fueled by the promise of dollar savings if organizations change from centralized, mainframe-based applications to distributed PC-based computing. It has become more cost-effective for companies to buy an off-the-shelf business application, than it is to develop and maintain equivalent programs on the mainframe.[2]

From the technical perspective, the client/server architecture evolved in response to performance bottlenecks prevalent in conventional PC LAN applications. In this environment, whole files pass across the network to be processed on each workstation. This causes unneeded network overhead, slow response time, and usually makes data unavailable to all but one user at a time.

---

[1] Lile, Edward A., "Client/Server Architecture: A Brief Overview," Journal of Systems Management, p. 26.

[2] Lile, p. 26.

The client/server architecture runs the database software on the server machine. When the client application needs to retrieve or update information, it sends a command (usually in SQL) to the database server. The server processes the request, retrieves and updates records, and sends records back to the application. The architecture allows application developers to spread resource utilization more evenly across the network and improve response time for local requests by running part of an application on the user's workstation.[6]

For the DRTool system application, the client/server architecture enables front-end development of graphical user interfaces (GUI) with support from multi-vendor server platforms and transparent communication tools used by the servers and clients. The choice to use the client/server architecture for Program 212 is within scope of current business and technology trends. This will allow the DRTool system to receive adequate levels of vendor support, and provides the capability to expand or upgrade the system as new products are released.

<u>Database System</u>

The database must support centralized data storage to provide instant access across the network and must support the client/server environment. A relational DBMS fully supports the system server which can be accessed by SQL statements. The SQL statements can be embedded in certain programming languages so that the database commands are transparent to the user. The DBMS system is more efficient with regard to data storage and performance since records or table spaces can be updated instead of entire files. This reduces system overhead and increases system response times.

Other database systems originally investigated included mainframe systems that used flat files and Direct Access Storage Devices (DASD). The mainframe database systems are outdated in terms of operability, performance and vendor support. The declining level of vendor support would require the database system to be converted in the future to a more current system that provides the required level of vendor support to the customers. Based on cost differences, reliability, vendor support and the amount of storage capacity required, the relational DBMS emerged as the best database for the DRTool application.

Server Application

There are several vendors that provide servers for client/server applications. Since the database system that was chosen is a relational DBMS, a system server would provide the best system performance when accessing the database. This includes loading and indexing times, query times, and read/write capabilities. Several servers provide portablilty across multiple GUIs, object-oriented programming, heterogeneous database access capabilities and upgrade capabilities to support system growth. The servers also provide the capability for secure data storage. The system servers are developed on current technology methodologies to support the relational databases, and this leads to a higher level of vendor support. Based on the possible server types, the one back-end server per database architecture as depicted in Figure 1 was chosen to provide the required data performance and access necessary to support the DRTool system.[7]

Software

The C programming language was chosen as the software language since several vendors and commercial off-the-shelf (COTS) products are written to interface with C. Considered a third generation programming language, C is a general purpose programming language which features economy of expression, modern control flow and data structures, and a rich set of operators. The language is not specialized to any particular area of application, but its absence of restrictions and its generality make it more convenient and efficient for many tasks than supposedly more powerful languages.[4] C code is compiled and provides faster

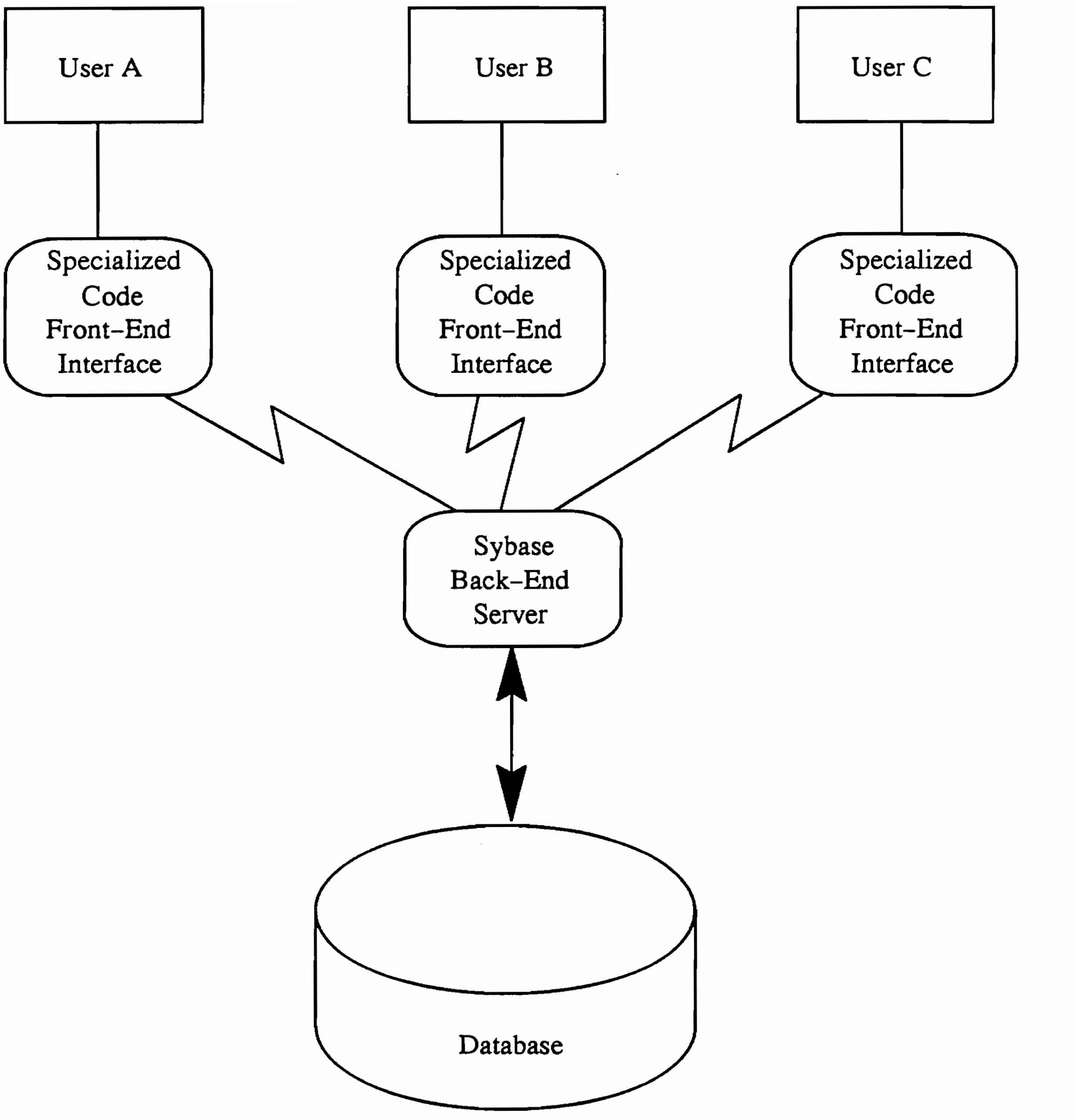


Figure 1: DR System Database Architecture

executables than other software languages. The SQL statements that access the database system can be easily embedded in the C code. Other programming languages, such as FORTRAN HX and VS do not allow for embedded SQL statements. This is important since embedded SQL statements are transparent to the user, so that a minimal amount of operator action is required to retrieve the DR data from the database system. The C programming language also provides excellent memory management capabilities and is easier to support and maintain than other software languages. Older languages such as FORTRAN are becoming outdated, and vendor support is steadily decreasing. Several vendors offer inexpensive training courses for C programming at locations throughout the United States. Training is required for effective programming and customer support and should be considered when choosing a programming language. The Sun SPARC workstations are delivered with the C programming language compiler and licenses.

Computer Display Package

This portion of the DRTool system must support the development of the user panels where data will be entered and displayed. Many companies produce GUI packages that integrate well with different operating systems. The GUI panels provide the user with the capability to select options through pull-down menus and button selections. They also allow for error messages to be displayed when users make an invalid selection or system request. A GUI display allows quick panel changes when necessary, and provides an easy method for users to access functions within the DRTool system. The GUI package also allows the application to close down to an icon so that the user can perform other functions on the screen without having to exit out of the DRTool system.

System Operational Requirements

The System Operational Requirements were established once the analysis of need was performed and the feasibility of various technology applications were considered. For the DRTool system, this lead to the definition of system operating characteristics, the maintenance support concept for the system, and the identification of specific design criteria.[2] The operational concept was broken down into seven areas of design criteria, including: mission definition, performance and physical parameters, use requirements, operational deployment or distribution, operational life cycle, effectiveness factors, and environment.[3] These design criteria were applied to the DR system in the following manner.

Mission Definition - Create a database, software and hardware system for the softcopy storage of DRs. The system must meet the high level requirements identified in the Definition of Need. The user must have the following capabilities:

- Create DRs
- Update DRs
- Retrieve DRs
- View DRs
- Route DRs to other users
- Produce hardcopy prints of the DRs
- Sort DRs by user defined fields
- Search for DRs by user defined fields
- Run reports against Drs in the database

The DR administrator must have the following capabilities:

- All user defined capabilities
- Reject new or updated DRs
- Route DRs back to the users for more information
- Apply the DRs to the database

---

[3] Blanchard, Benjamin S. and Wolter J. Fabrycky, Systems Engineering and Analysis, p. 37.

- Perform preventive maintenance on the system

The DR data flow is depicted in Figure 2.

Performance and Physical Parameters - The components of the DRTool system were identified to have the following performance and physical parameters:

Database

- The nominal database shall have a storage capacity of 10 Gigabytes (GB)
- The power required is 208 volts, 2 phase power with a backup power supply
- The database shall have a 960 CPU with a clock speed of 25 MHz and throughput greater than 100 Mbps
- The database shall have one (1) Megabyte (MB) of Cache Memory
- The database shall hold a maximum of 15,000 open DRs
- The database shall support a DBMS system

Server

- The server requires 400 MB of raw partitioned disk space
- The sever software requires 50 MB of storage space
- The server shall support a maximum retrieval time of 9 seconds for an open DR
- The server shall support a maximum retrieval time of

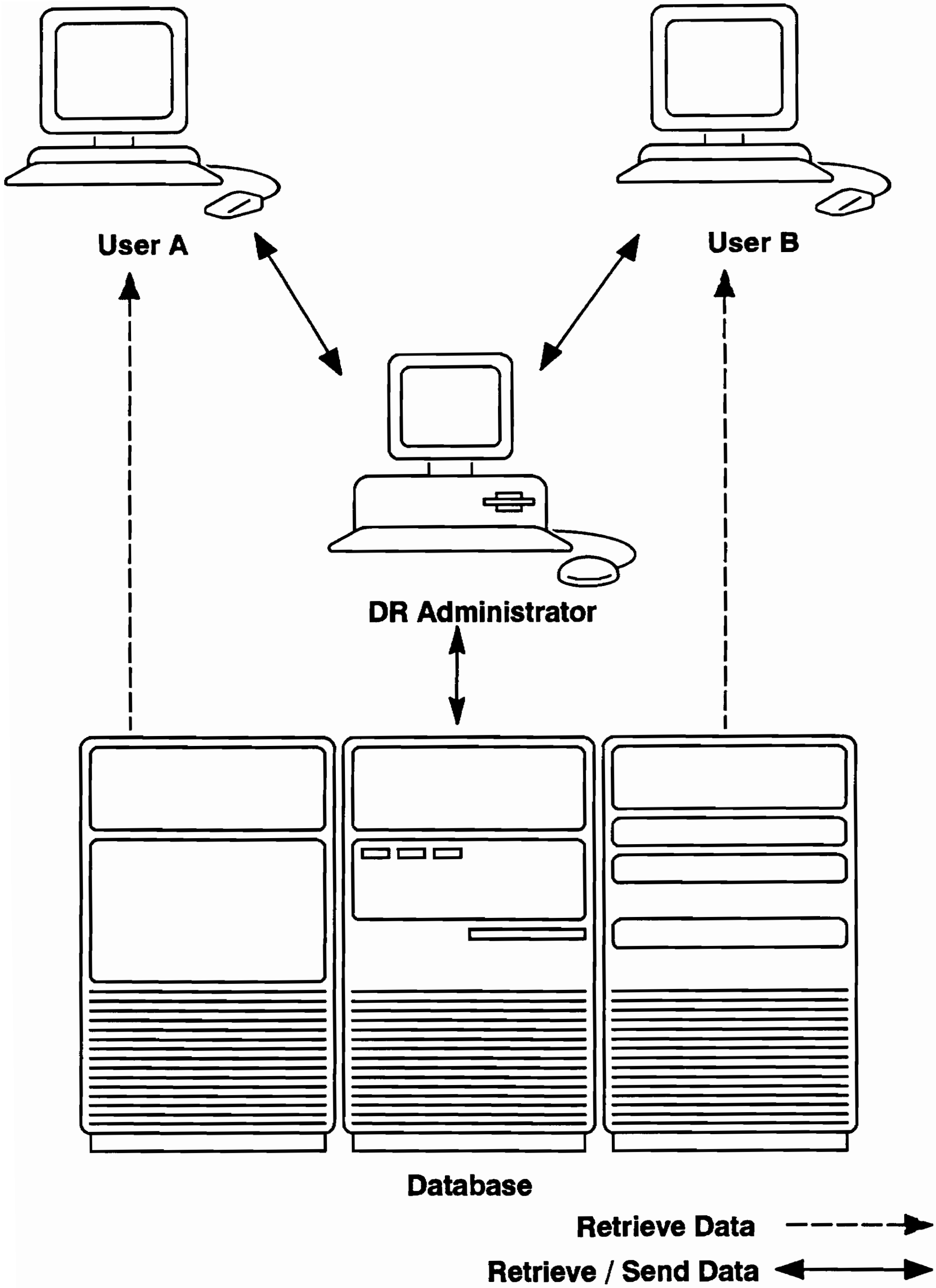


Figure 2: DR System Data Flow

20 seconds for one closed DR when the number of closed DRs stored in the database is less than 100,000. An additional 10 seconds is required to support each additional 100,000 closed DRs stored in the system.

Terminals

- The terminals shall have 32 MB of RAM
- The terminals shall have a 1.0 GB Fast SCSI-2 Disk Drive
- The terminals shall require 300 MB of storage space for the operating system
- The terminals require 50 MB for user's home directories
- The terminals require 50 MB to store the application software
- The terminals shall support GUI panels and menu driven software
- The terminals shall have a keyboard for entering data
- The terminals shall have a mouse for selecting options and pulling down menus
- The monitors shall be 19 inches and display in color
- The terminals shall require normal RG58 coaxial cable for data flow to the database
- The terminals shall require transceivers to convert

data on the coaxial cables to machine readable formats

Tape Backup Storage Device

- The Tape Backup shall have a storage capacity of 5 GB
- The Tape Backup shall use 8 mm tapes for storing the data

Use requirements - The DR system was designed for use 24 hours a day, 7 days a week, except for 1 hour per week scheduled preventive maintenance down time. The 1 hour down time would be scheduled during off-peak hours. The DR database administrator is usually staffed during normal business hours (9 AM to 5 PM). Usage of the system may see peaks during the week or at the end of the month, as weekly or monthly reports are generated.

Operational deployment or distribution - The hardware at the customer site will consist of:

- Two or more workstations
- One relational database
- One system-database server
- Transceivers and RG58 cabling from the database to the workstations
- One laser printer for producing hardcopy reports

- One 5 GB 8 mm Tape Backup Drive Storage device.

The specialized applications software developed by Program 212 to interface with the server will be installed by the Program 212 system operator during the site installation. All components of the system will be ordered by Program 212 and shipped directly to the user location for installation. The system requires one week (7 days) for installation, checkout and test before becoming operational. The system is currently being supported in Los Angeles (CA), Valley Forge (PA) and
Reston (VA).

<u>Operational life cycle</u> - The system was designed to operate for seven years due to the fast changes in computer technologies. The system will be operated by individuals whose primary occupation is data entry or configuration management, and should be simple to use. The DR administrator will require some knowledge of relational databases and client/server architectures, since this person will perform some preventative maintenance tasks, such as weekly backups of the system. The simplicity of the design will aid in the maintenance of the system, which will be centralized at Program 212. Resources at Program 212 are limited, and the maintenance concept will clearly define troubleshooting responsibilities, baseline management and

system upgrade concepts. The system has a parallel design such that if it is necessary to replace or upgrade a workstation, this could be done with minimal impacts to the system (another terminal exists to access the system). If the database fails, the vendor, under maintenance contract, will provide a new piece of equipment. Program 212 will reinitialize the system from the weekly backups so that data loss by the customer is reduced.

Effectiveness factors - The system was designed for effective use by operators with low levels of computer knowledge. Certain performance parameters can be monitored by the DR administrator. such as CPU and I/O statistics, resources allocated vs. resources used, and memory usage. The DRTool system was designed with several effectiveness factors in mind. These factors are presented below, with the calculations given in Appendix C.

- System operational availability shall be 99.80%
- System reliability for 1000 hours of use shall be 91.36%
- The Mean Preventive Maintenance Time (Mpt) shall be one (1) hour/week
- The Maximum Maintenance Down Time (MDT) shall be 48 hours for the database
- The Mean Time Between Failures (MTBF) for the system

is 12000 hours which is driven by the Program 212 specialized software since this is the component with the greatest chance of failure

- The initial cost of the system is limited to $340,000

Environment - After installation, the DRTool system is a non-movable computer system. The hardware components and associated documentation will be shipped from the vendors directly to the user site prior to system installation. The server software, panel displays and specialized application software are saved to 8 mm tapes (with a backup set) and will be brought to the user site by the Program 212 individual that will install and initialize the DRTool system. The hard disks can be configured by Program 212 at their location prior to the system initialization, but the vendor licenses must be installed at the user site during the system setup. The ideal room conditions for system would be maintained between 65 and 75 degrees Fahrenheit with a humidity between 42 and 58 percent. This provides optimal equipment and personnel performance.[1] It was recommended that weekly backup files be stored in a fire retardant location to ensure that the system could be established in the event of a facility damaging event.

System Maintenance Concept

The maintenance concept, which evolved from the definition of operational requirements, establishes: anticipated levels of maintenance support, the basic responsibilities for support, general overall repair policies and/or constraints, major elements of system support as they apply to a new system, and the effectiveness requirements associated with the system support capability. The maintenance concept describes, in general terms, the overall support environment for the system.[4] The DR system maintenance concept will cover three levels of maintenance. They include: organizational maintenance, intermediate maintenance, and supplier or factory maintenance. The levels of support are described as follows.

Organizational maintenance - This maintenance is performed at the customer's operational site. These include tasks performed by the using organization on its own system. For the DR system, the DR Database Administrator will be responsible for performing some preventative maintenance tasks, including:

---

[4] Blanchard and Fabrycky, p. 40.

Server Maintenance

- Monitor CPU and I/O statistics
- Monitor server error log content and size
- Save server error log to tape and delete oldest seven days of data
- Compare resources allocated vs. used
- Backup current system configurations

Database Maintenance

- Monitor allocated space vs. in use space
- Backup the database to tape
- Dump the database transaction logs to tape
- maintain all object creation scripts

The Mean Preventive Maintenance Time ($\overline{M}pt$) is 1 hour/week. Users manuals will also be provided to aid in system operability and maintenance.

Intermediate maintenance - These maintenance tasks are performed by mobile, semimobile, and/or fixed specialized organizations. For the DRTool system, this may include vendor maintenance such as terminal replacement in the event of a failure, or database service in the event of system down time. The objective is to provide on-site maintenance to facilitate the return of the system to its full operational status on an expedited basis.[2] The following maintenance requirements were established for the DRTool system:

- Vendors will respond to hardware problems within 24 hours of receiving a service call as stated in their maintenance contract
- Program 212 will respond to specialized software problems within 24 hours of receiving a HelpLine service call
- Replacement of hardware equipment under the maintenance contract will be performed by the vendor organization
- Program 212 will support system reinitializations in the event of database replacement due to database failure
- All software upgrades or fixes shall be tested at the program 212 site and the user site prior to the software being built to the operational baseline
- Program 212 shall provide the operational site with documentation updates required by software upgrades or fixes
- All hardware upgrades shall be tested by the vendors at the vendor site and the user site prior to the installation into the operational system

- The Mean Corrective Maintenance Time ($\bar{M}ct$) of the system is 4.203 hours
- The $\bar{M}ct$ of the components are:

  Database - 2.1 hours

  Software - 5.2 hours

  Server - 4.0 hours

  Terminal - 0.9 hours
- The Maintenance Down Times (MDT) of the components are:

  Database - 48 hours

  Software - 24 hours

  Server - 24 hours

  Terminal - 30 hours

<u>Supplier/factory maintenance</u> - These tasks require the highest skills, specialized equipment and a large amount of testing that is beyond the capabilities available from the intermediate support. The following falls under this level of maintenance:

- Components that were replaced by the vendor at the customer location where the vendor felt the piece could be fixed at their factory to normal operational mode
- All components that are beyond repair would be

disposed by the vendor or used for parts

- Program 212 system upgrades fall into the supplier level of maintenance since the upgrades are developed and tested at the Program 212 location and are installed on a semi-annual basis

PRELIMINARY SYSTEM DESIGN

The preliminary system design began with the technical baseline for the DRTool system that was defined in the Conceptual Design Phase, and extended throughout the translation of established system level requirements into detailed qualitative and quantitative design requirements. For the DRTool system, this required performing a functional analysis and requirement allocation, the accomplishment of trade-off studies and optimization, and system synthesis.[2]

System Functional Analysis

The functional analysis for the DRTool system provided a basis for the identification of design requirements for each hierarchical level of the system. This ensured all facets of the system development, operation and support were covered, and that all elements of the system were fully recognized and defined.[2]

Functional flow diagrams were developed for the DRTool system to structure system requirements into functional terms. The functional flow diagrams indicate the basic system organization and functional interfaces of the DRTool system as defined in the Conceptual Design Phase. Emphasis was placed on what was to be accomplished, versus how something should be accomplished. For the DRTool system, the top-level diagrams show the gross operational functions of the users and the DR administrator. Once the top-level flows were defined, additional levels were expanded down until the level necessary to define the needs of the system were established. The system flow begins in Figure 3 with the functional flows of the users defined in Figures 4 - 8. The functional flows of the DR administrator are shown in Figures 9 - 12.

Once the operational function flows of the DRTool system were defined, maintenance functional flows were established. This included preventive maintenance flows and corrective maintenance flows. The preventive maintenance flow is shown in Figures 13 - 17, and the corrective maintenance flows are shown in Figures 18 - 22.

The functional flows provided a means for ensuring that all the operational and maintenance requirements that were identified in the Conceptual Design were captured and defined.

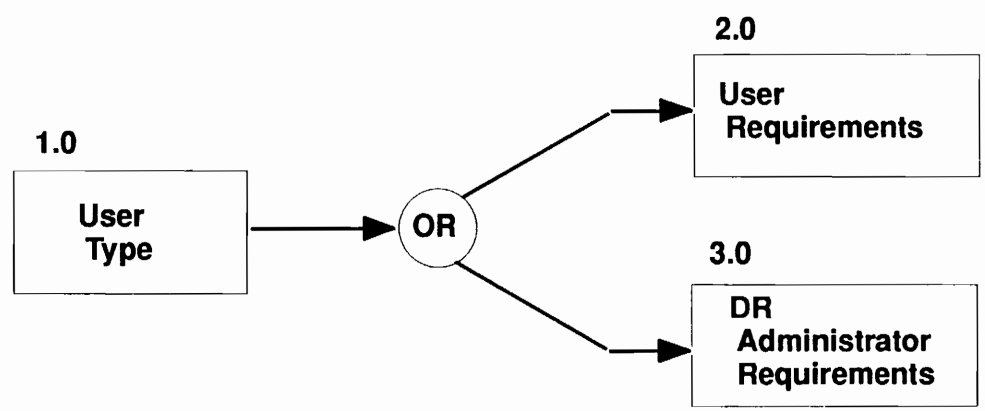


Figure 3: DRTool System Functional Flow

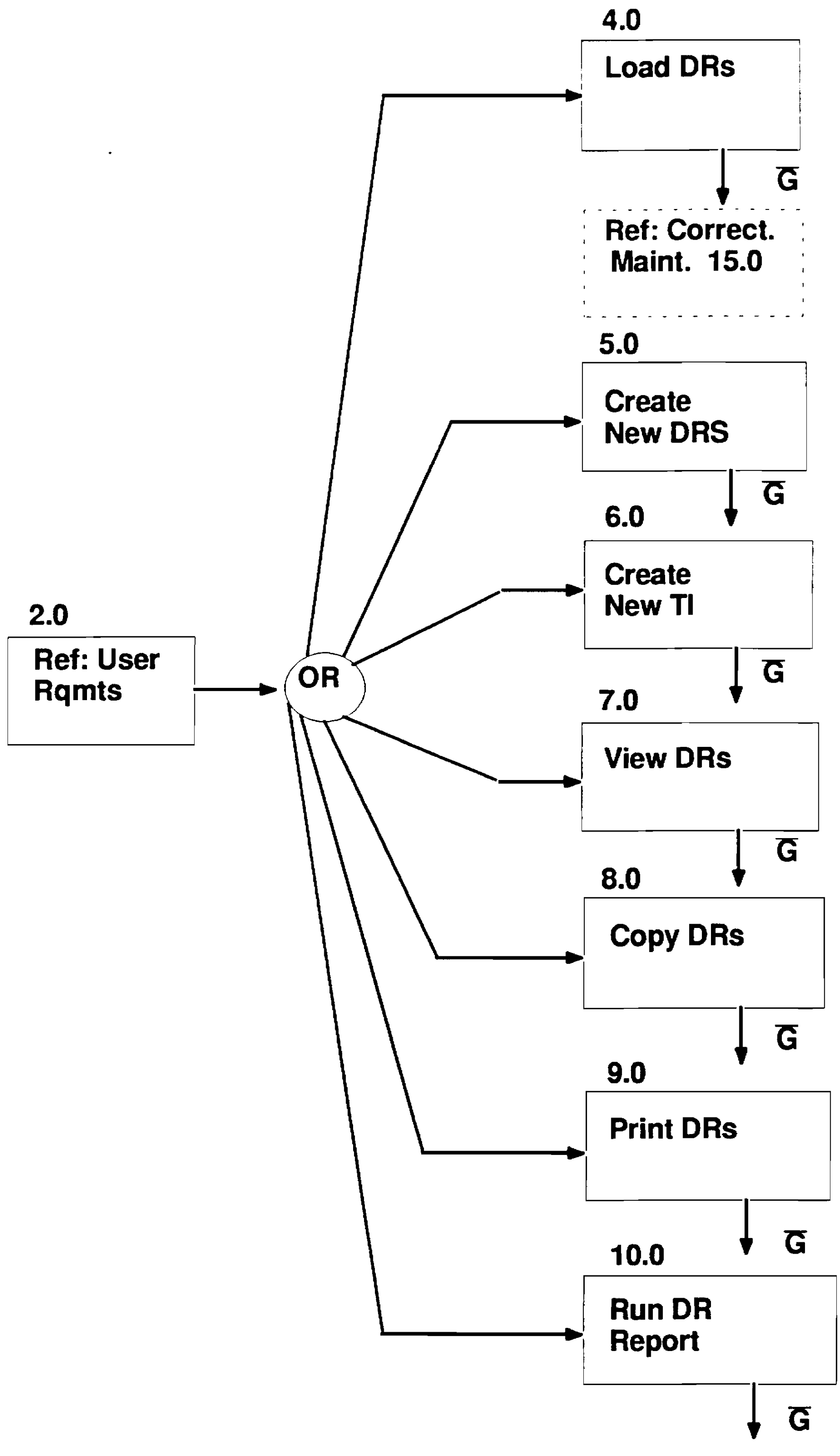


Figure 4: User Functional Flow

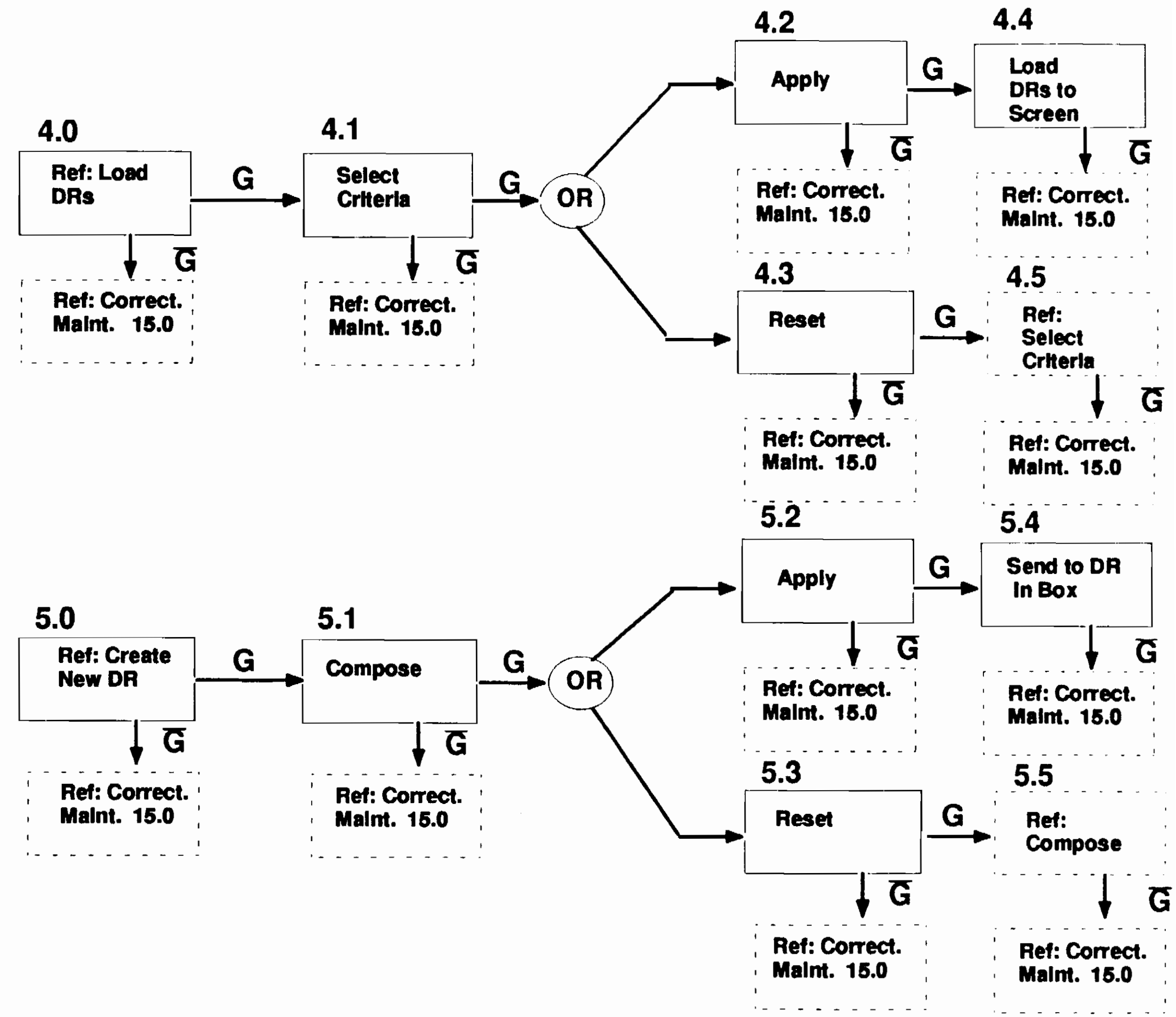


Figure 5: User Functional Flows (continued)

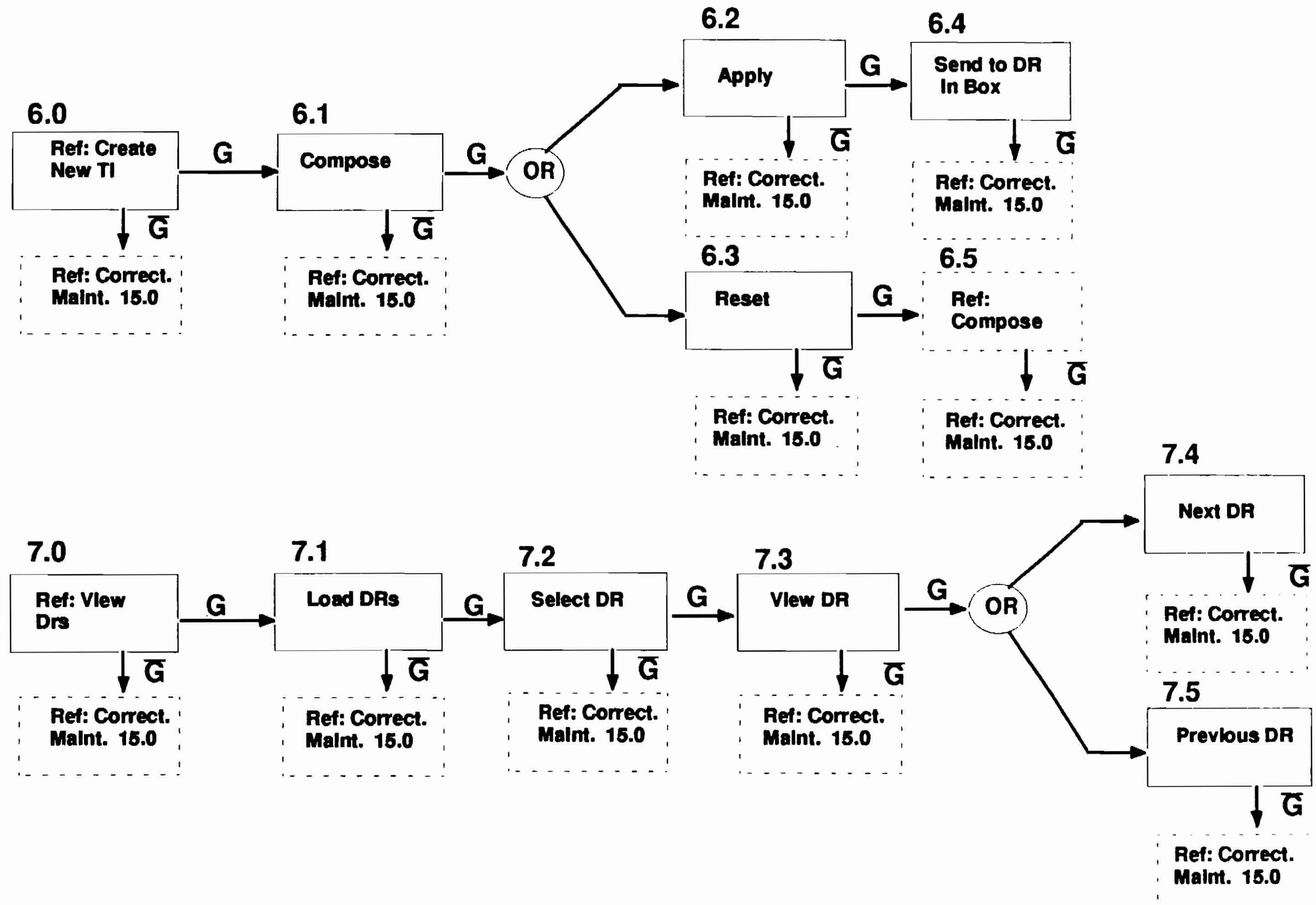


Figure 6: User Functional Flows (continued)

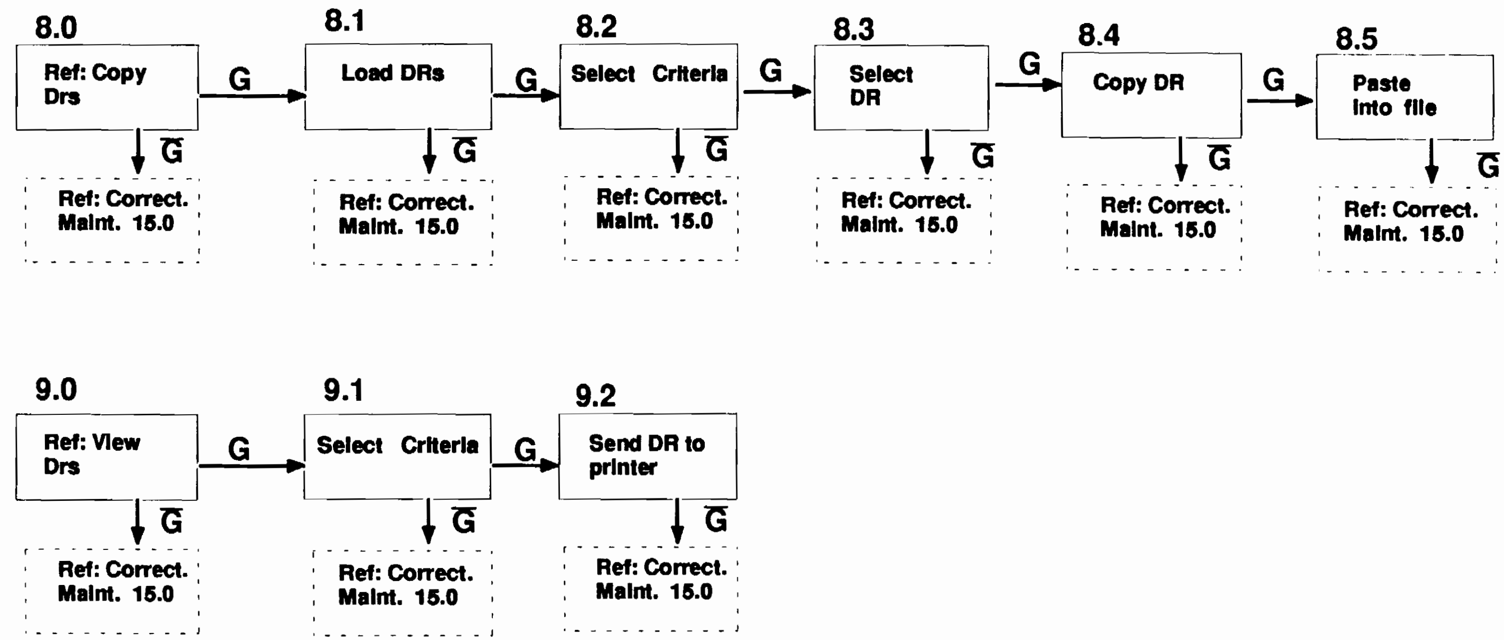


Figure 7: User Functional Flows (continued)

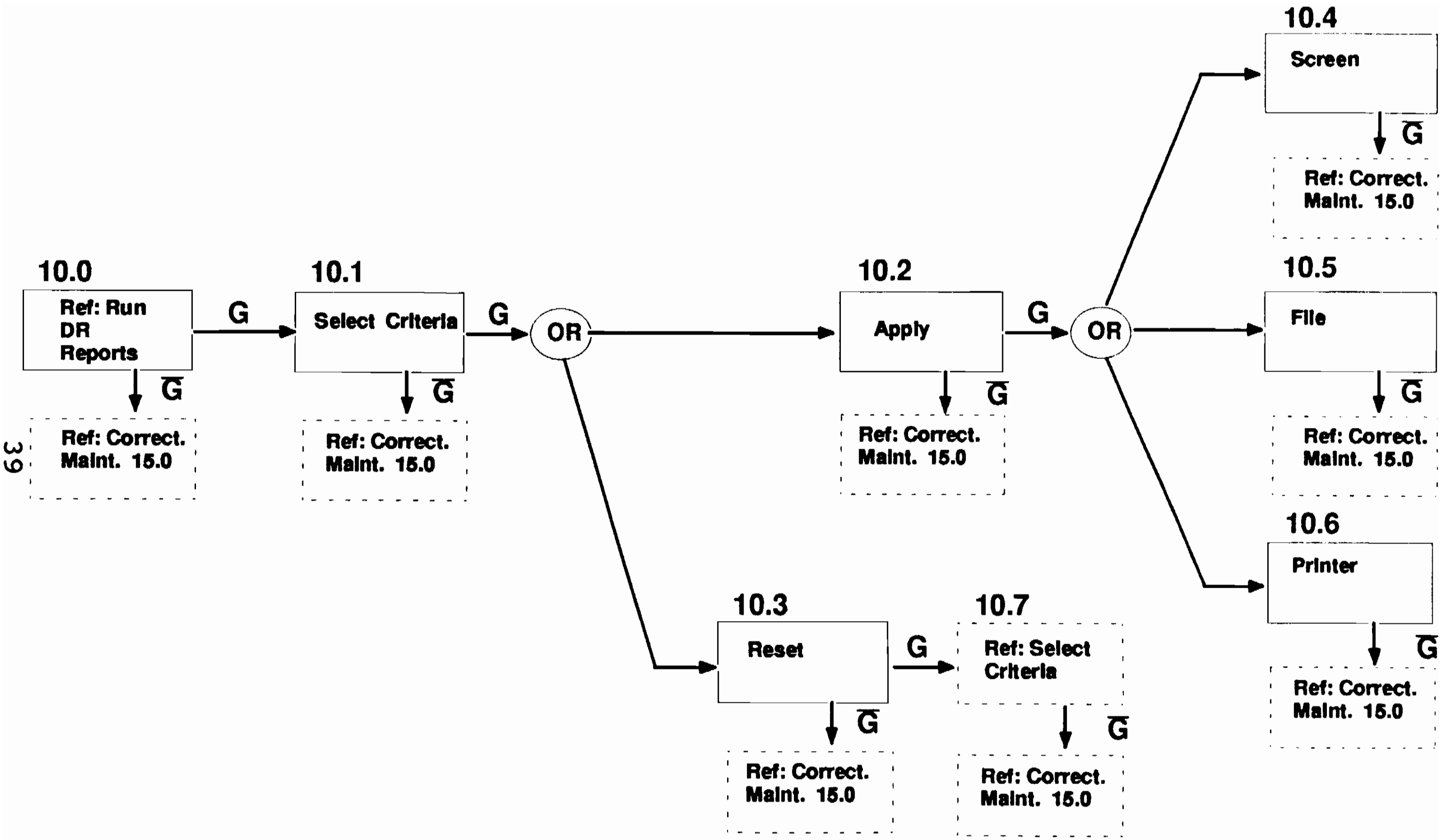


Figure 8: User Functional Flows (continued)

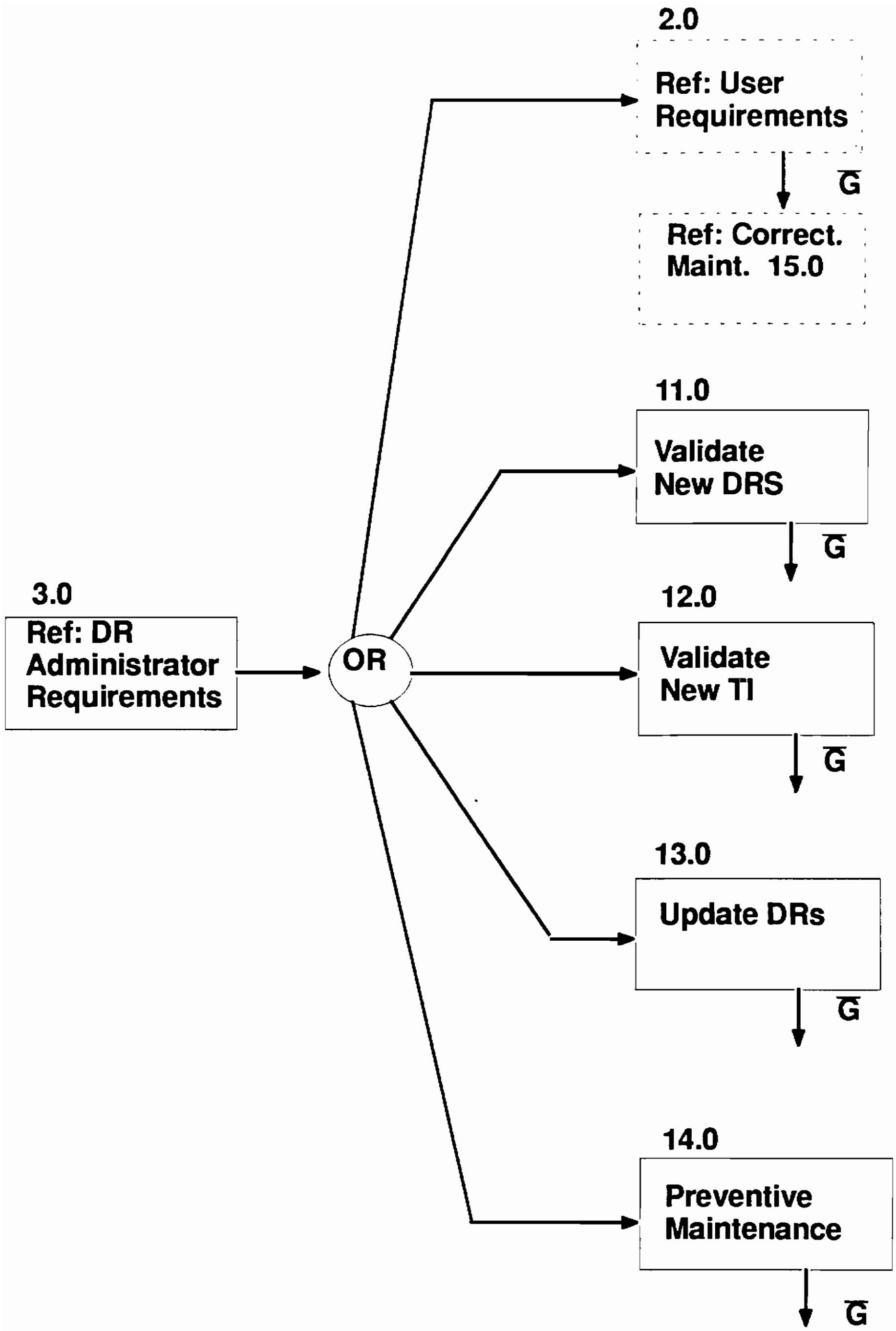


Figure 9: DR Administrator Functional Flows

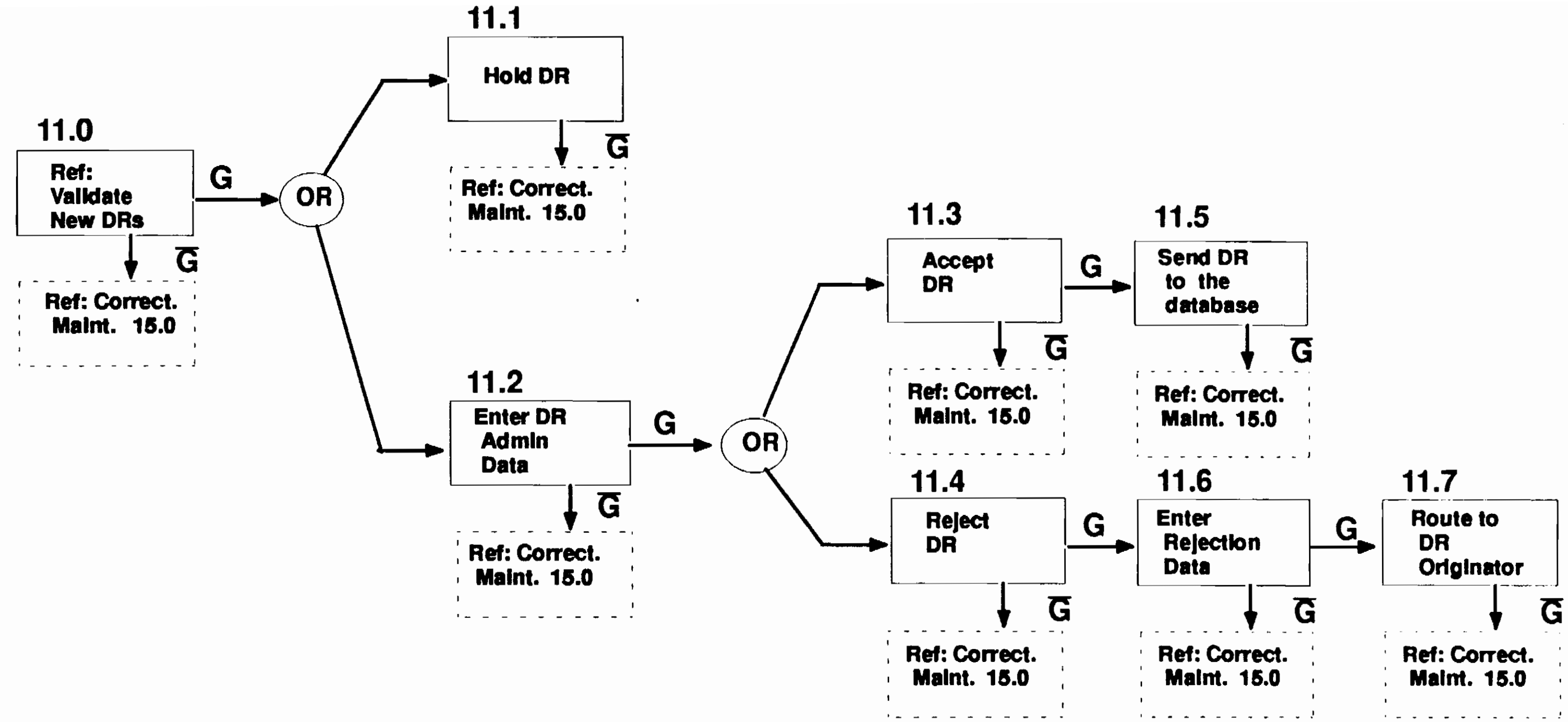


Figure 10: DR Administrator Functional Flows (continued)

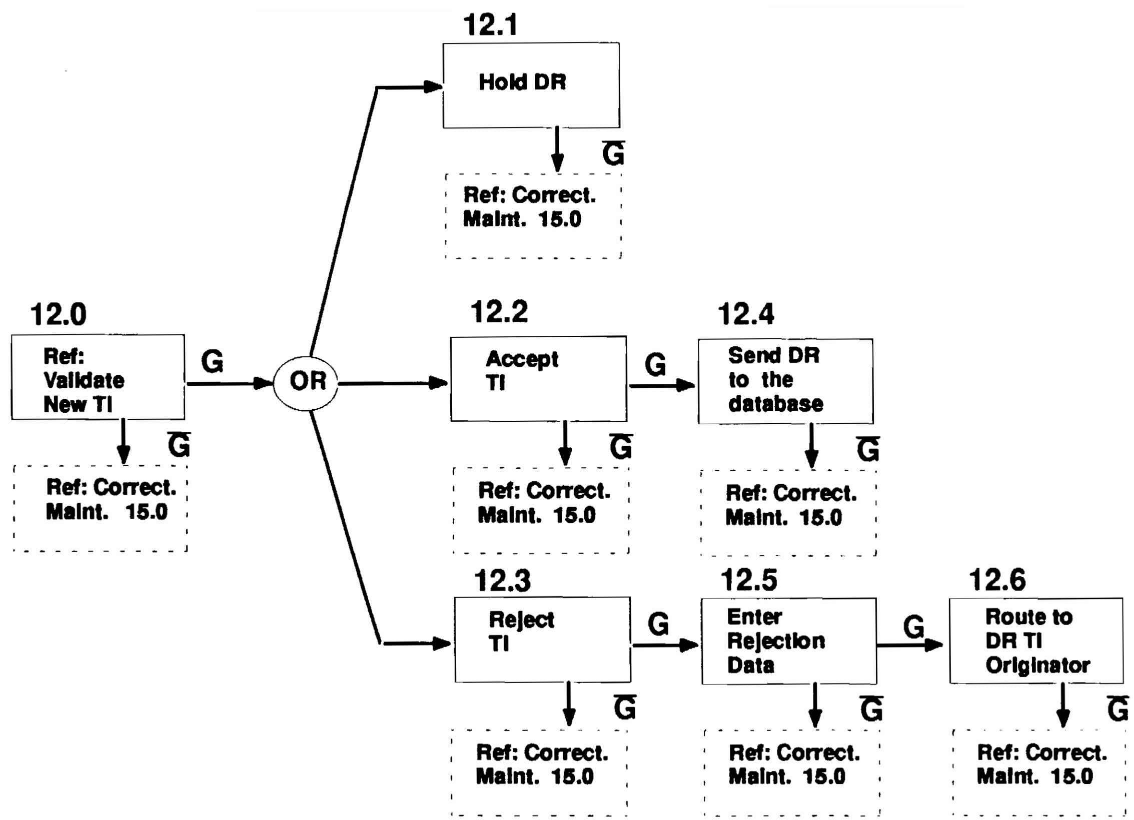


Figure 11: DR Administrator Functional Flows (continued)

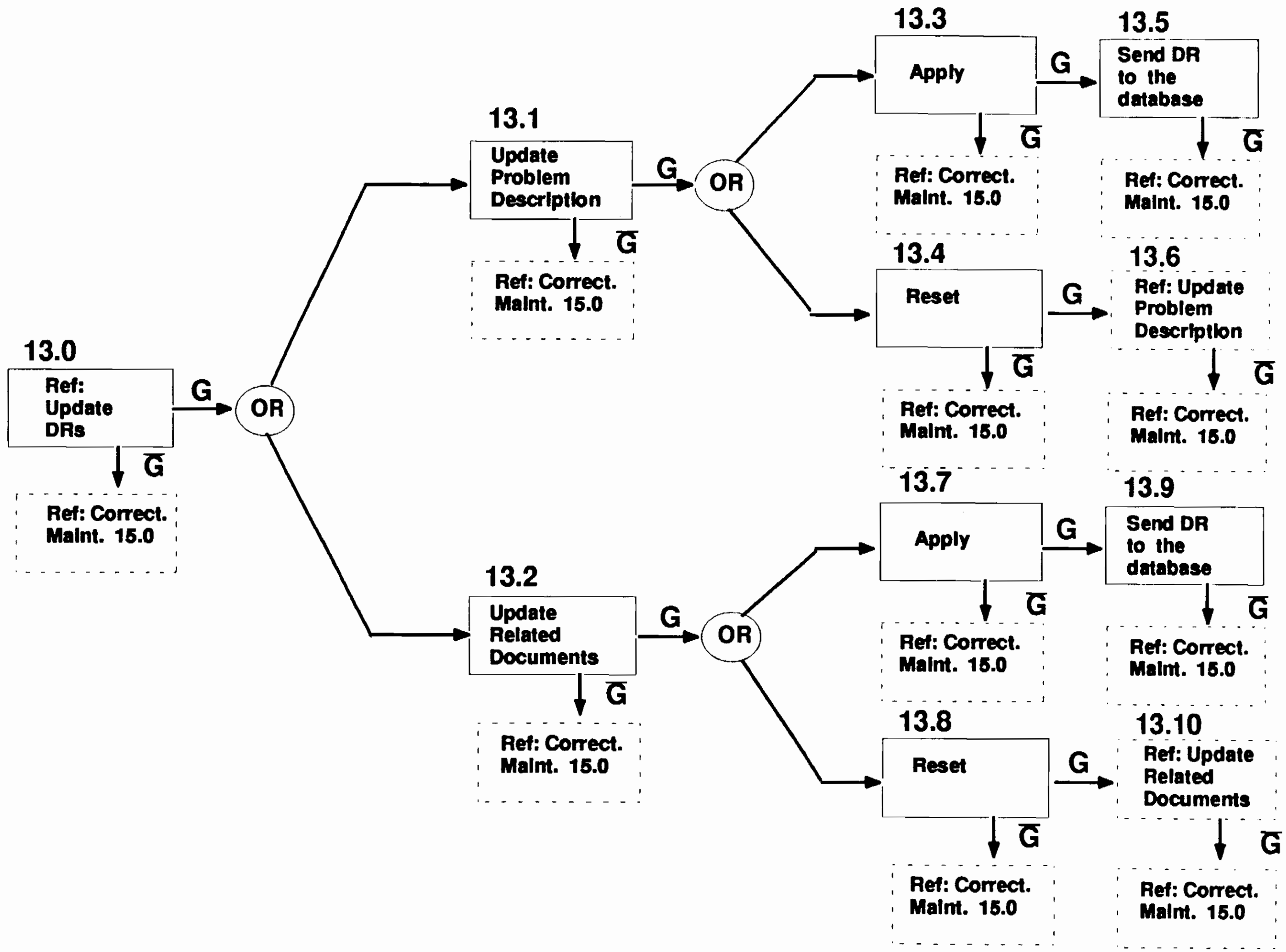


Figure 12: DR Administrator Functional Flows (continued)

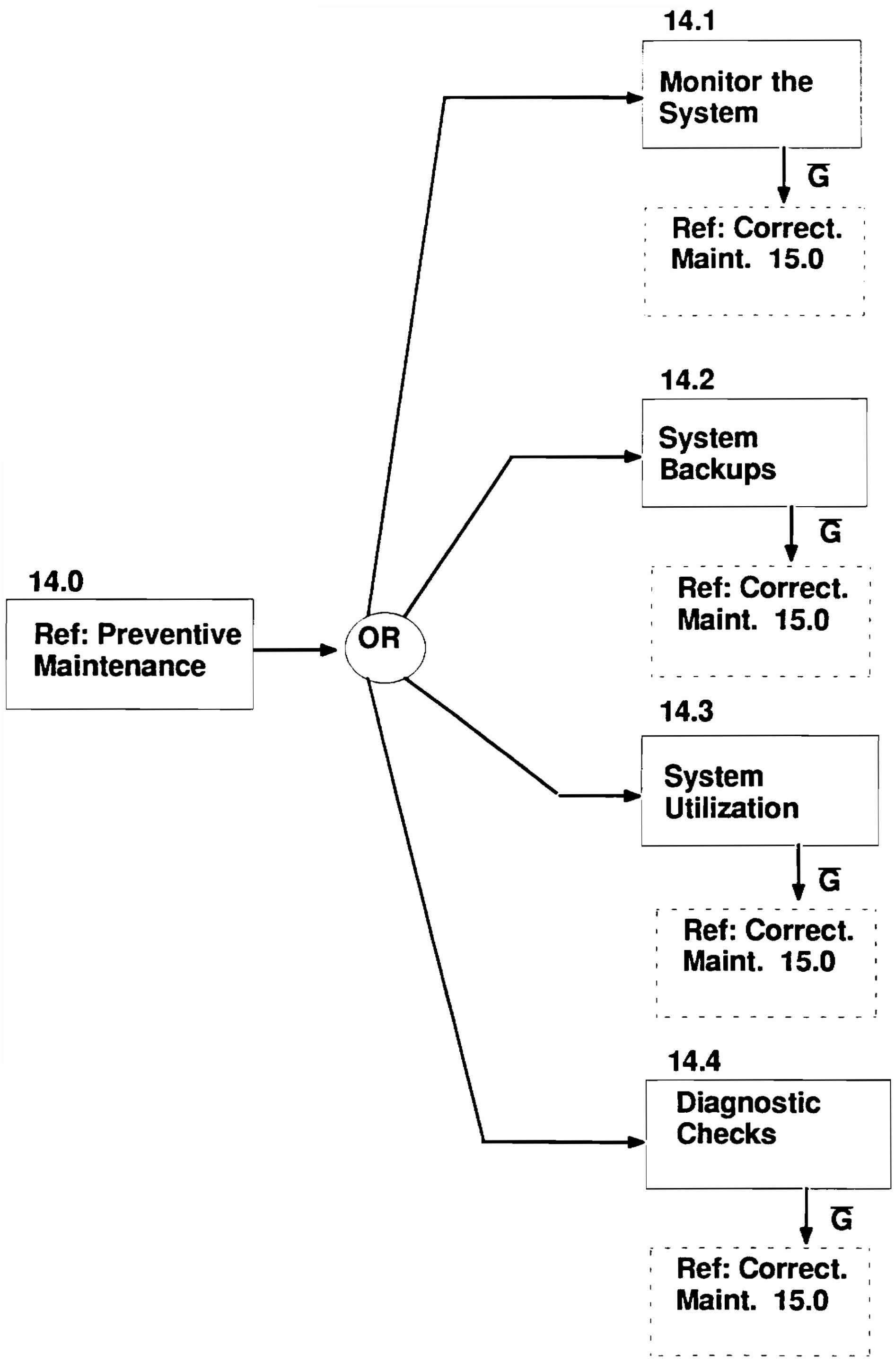


Figure 13: Preventive Maintenance Functional Flows

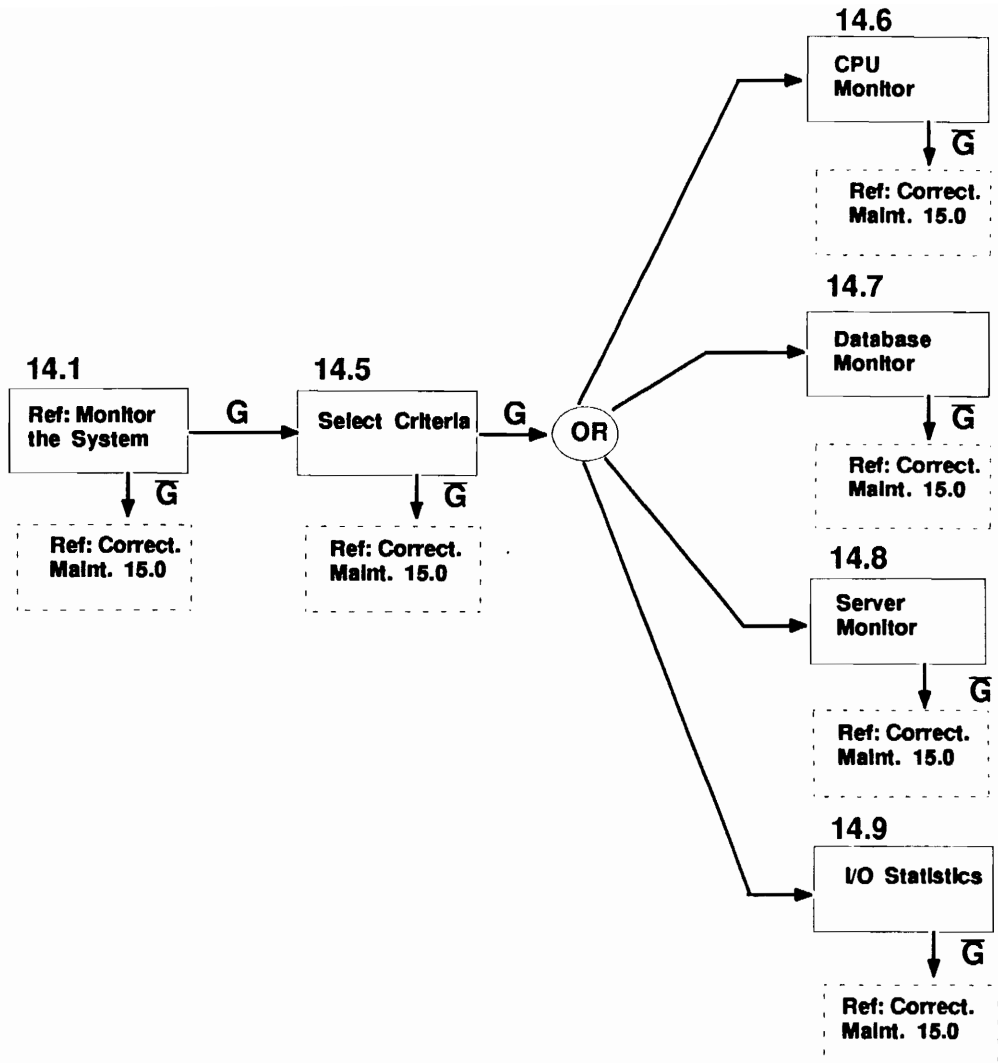


Figure 14: Preventive Maintenance Functional Flows (continued)

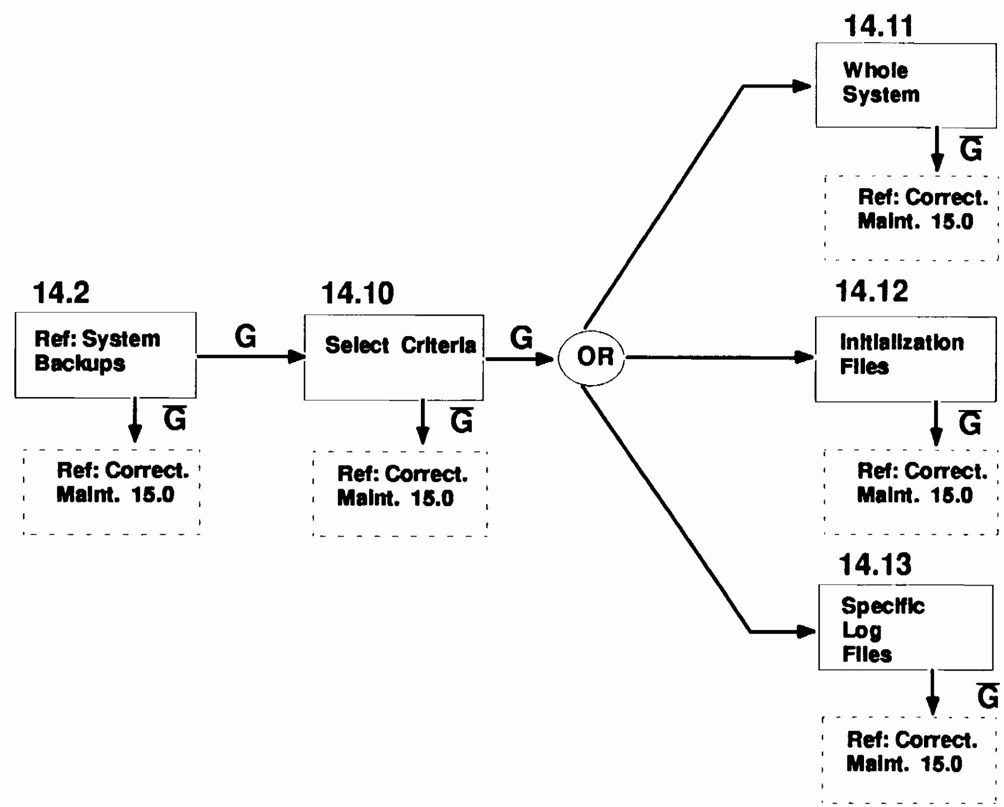


Figure 15: Preventive Maintenance Functional Flows (continued)

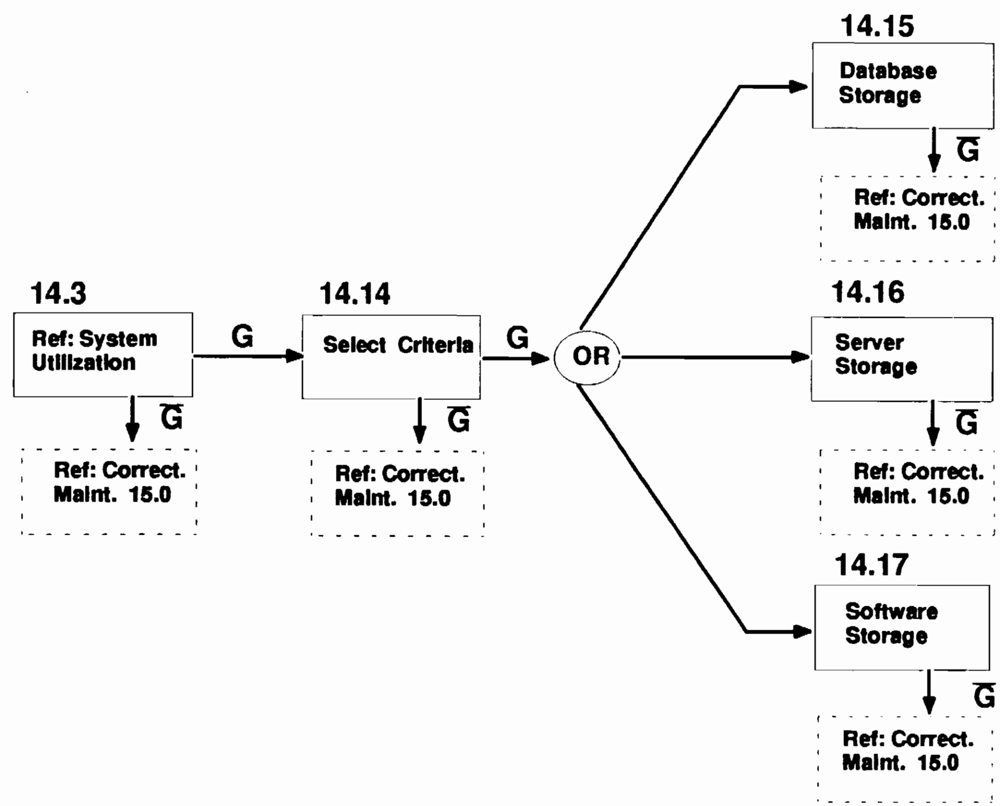


Figure 16: Preventive Maintenance Functional Flows (continued)

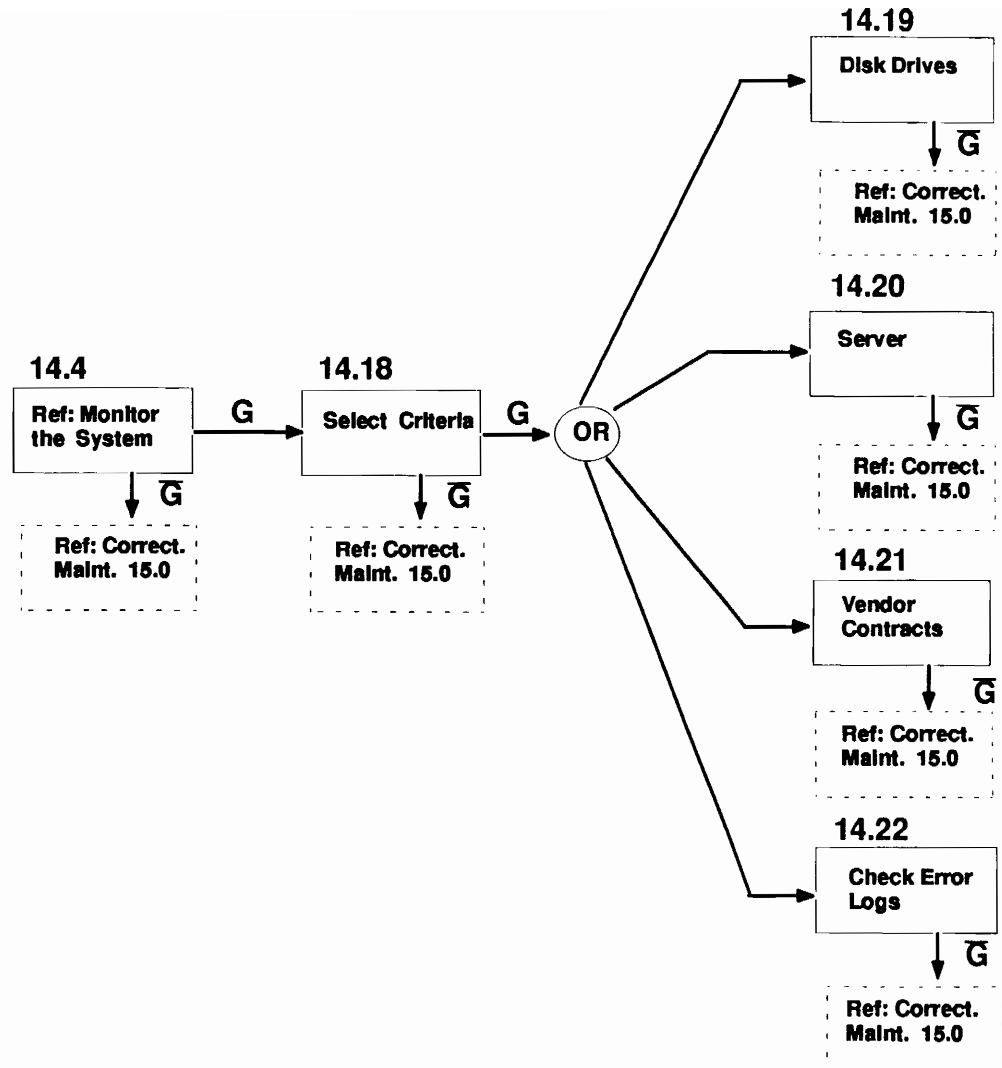


Figure 17: Preventive Maintenance Functional Flows (continued)

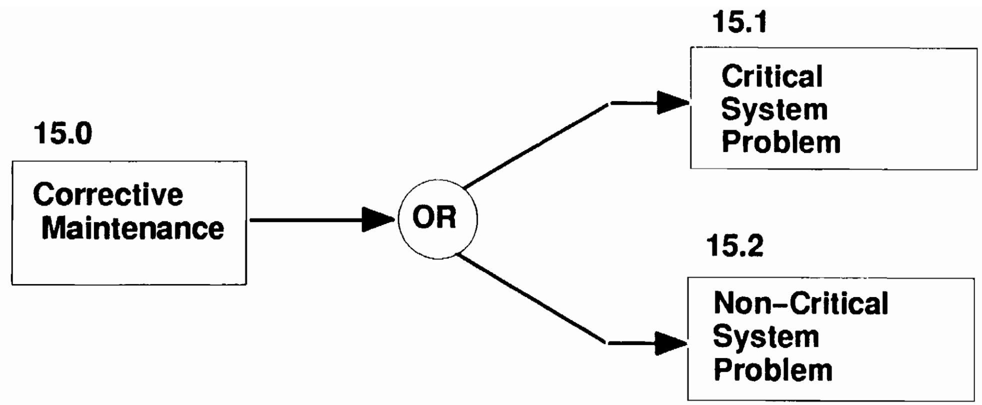


Figure 18: Corrective Maintenance Functional Flow

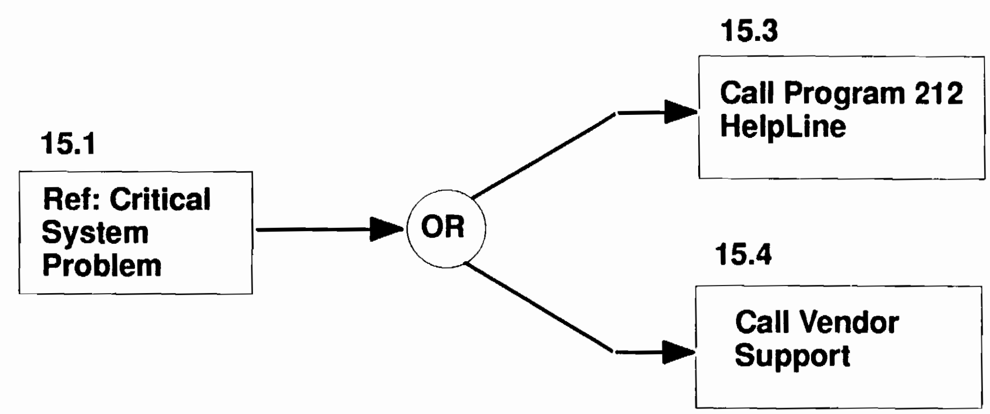


Figure 19: Corrective Maintenance Functional Flow (continued)

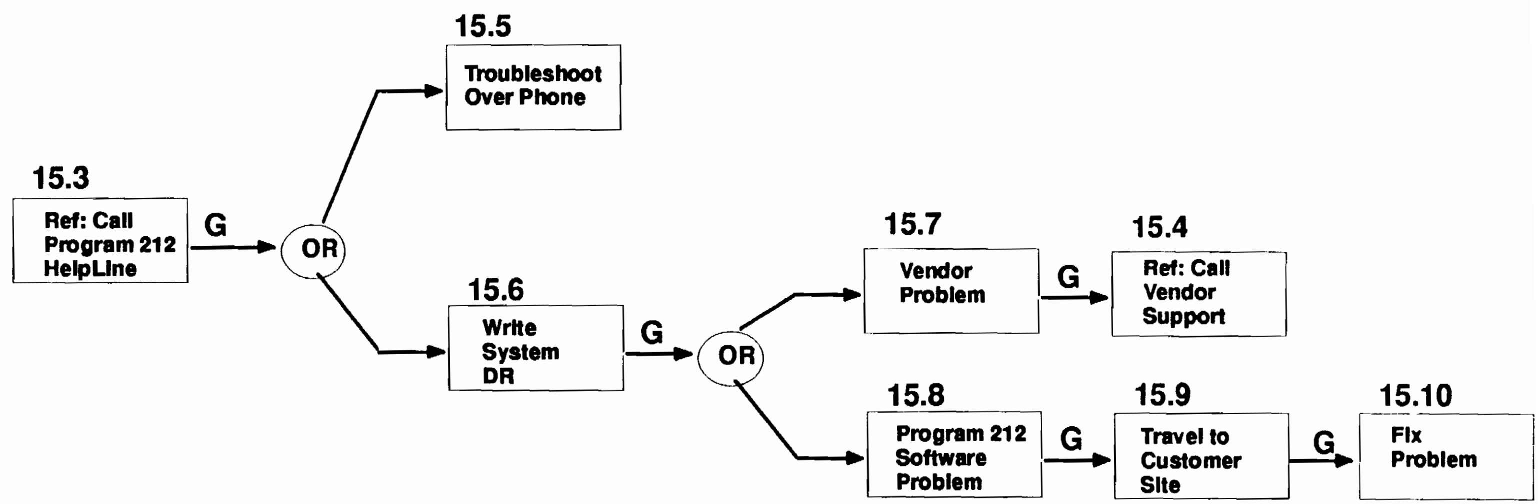


Figure 20: Corrective Maintenance Functional Flows (continued)

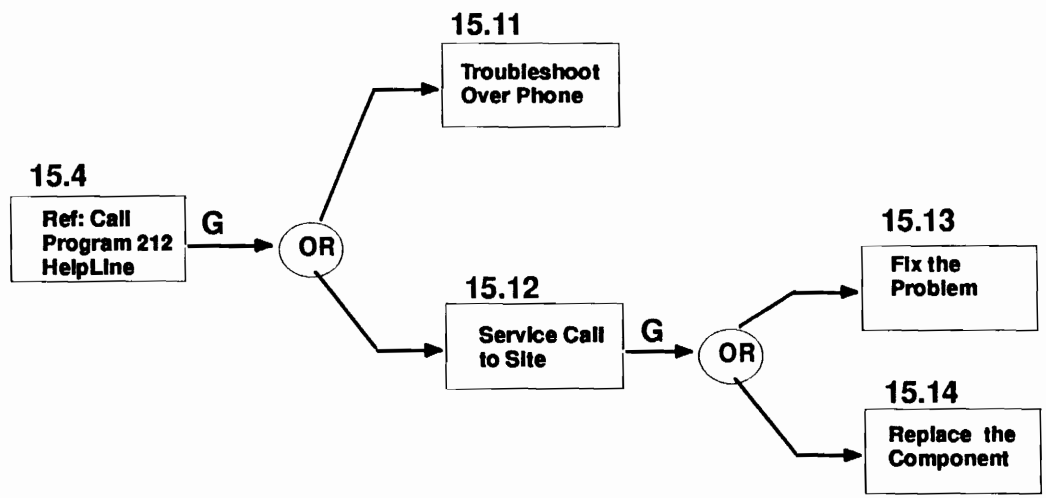


Figure 21: Corrective Maintenance Functional Flows (continued)

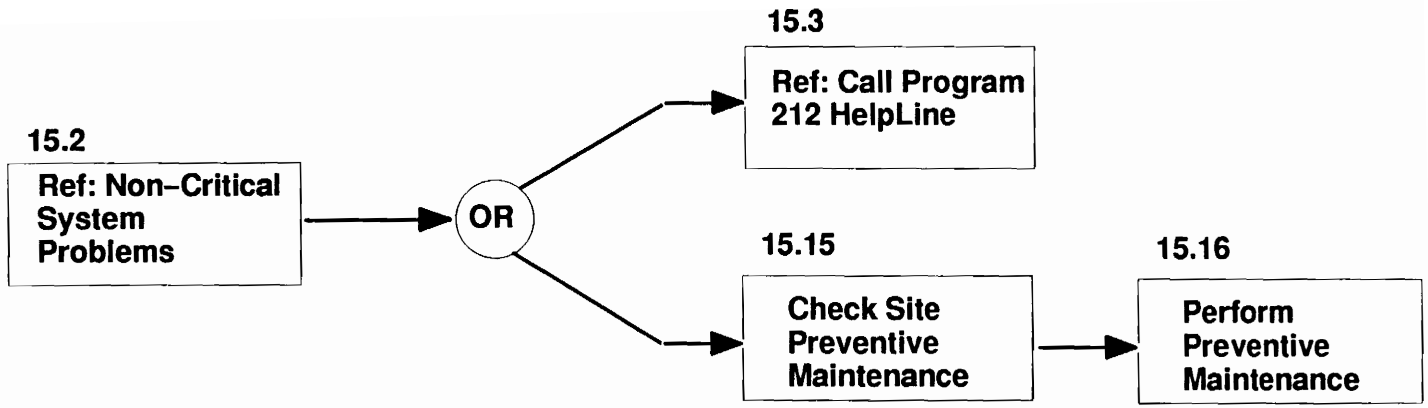


Figure 22: Corrective Maintenance Functional Flow (continued)

Allocation of Requirements

Once the functional analysis was complete and the major system functions were defined, top-level system factors were allocated to various subsystems and lower-level elements of the DRTool system. This included establishing the hierarchy of system components for the DRTool system, and identifying lower-level subsystem requirements that meet the requirements established at the system level. Factors such as system effectiveness, and system performance and physical parameters were identified. The hierarchy for the DRTool system is shown in Figure 23 where the system requirements were established with respect to the system components. Many of the requirements for these elements were identified in the Conceptual System Design in Appendices A and B. The overall reliability of the DRTool system for 1000 hours of use was calculated in Appendix C, where the reliability of the subsystem elements were defined and used in the calculations.

Trade-off and Optimization

For the DRTool system, several criteria were used while performing the feasibility study. Since the system requirements have been established, a study was performed to

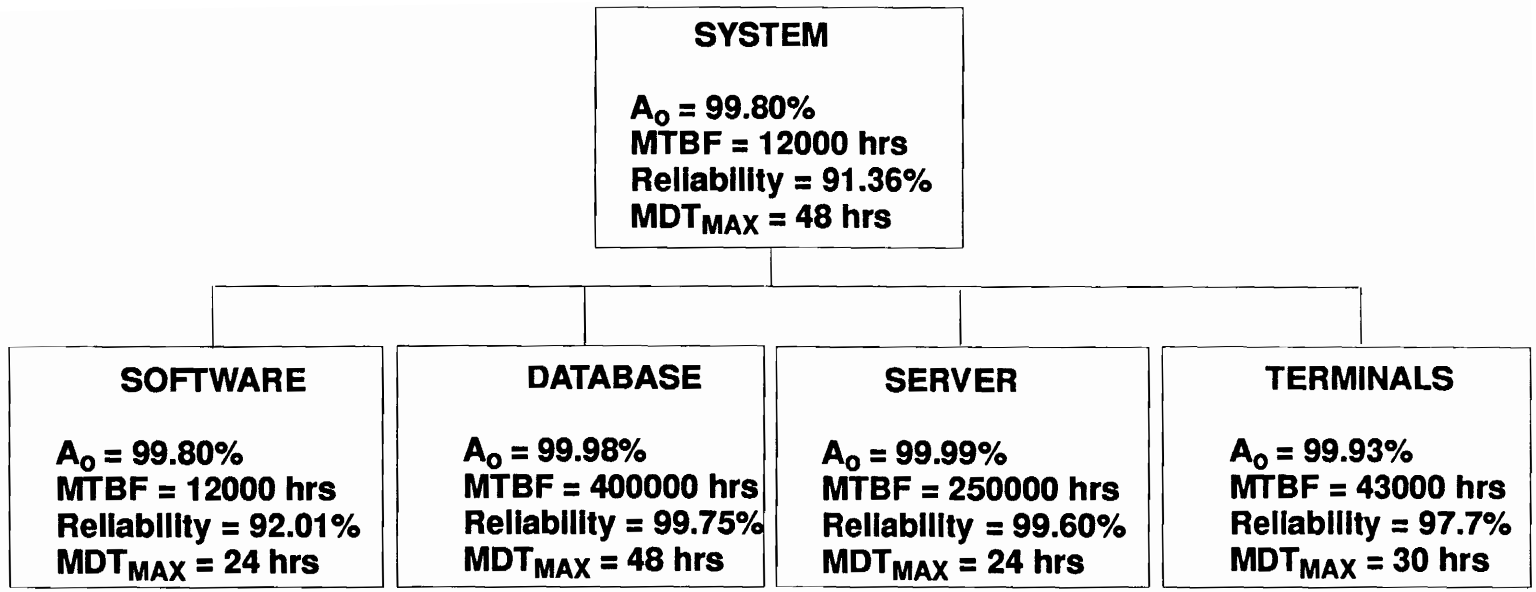


Figure 23: DRTool System Hierarchy

determine the best products for the specific applications identified in the feasibility study. These products were chosen to meet the system and subsystem requirements. The results of the study are provided below:

- Database - Mega Drive Systems MR/5 DBMS
- Server - Sybase 4.9.1
- Terminals - Sun SPARCstation LXs
- GUI Display Package - Motif

These choices are discussed to illustrate considerations that were made during the trade-off and optimization phase of the preliminary design. Decisions were based on performance, life-cycle cost factors, and the level of vendor support offered for each of the alternatives. Vendor support provides a large influence on computer systems since consumers often rely on vendors to provide immediate service, 24 hours a day. Once pieces of hardware or software fall out of the service period, vendor support becomes extremely expensive or may not be provided at all. The specific costs of the components are provided in Appendix A.

Database System

The database system chosen to store the DR information was the Mega Drive Systems Database Management System (DBMS) since it provides integration for the client/server architecture. The database supports centralized data storage which provides instant access across the network. Updates to the database can be made on demand or scheduled automatically to avoid peak traffic loads. The Mega Drive Systems DBMS fully supports the Sybase server which can be accessed by SQL statements. The SQL statements can be embedded in certain programming languages so that the database commands are transparent to the user. The DBMS system is more efficient with regard to data storage and performance since records or table spaces can be updated instead of entire files. This reduces system overhead and increases system response times.

Other database systems originally investigated included IBM mainframe systems that used flat files and Direct Access Storage Devices (DASD), IBM DB2 systems and other DBMS vendor products. The mainframe database systems are outdated in terms of operability, performance and vendor support. The declining level of vendor support would require the database system to be converted in the future to a more current system that provides the required level of vendor support to the customers. Based on cost differences, reliability, vendor support and the amount of storage capacity required, the Mega Drive Systems DBMS emerged as a slight leader among the database vendors for the DRTool application.

Server Application

There are several vendors that provide servers for client server applications. These include Oracle, IBM, Ingres, and Sybase. Factors such as loading and indexing times, query times, and read/write capabilities were considered with other factors such as cost, reliability and current technology developments. Several sources considered IBM the least advanced and comprehensive architecture for enabling client/server applications.[6] Ingres, a pioneer in GUI-based applications, has been quiescent since its acquisition by The ASK Group Inc. in 1990.[5] The new Oracle released Cooperative Development Environment (CDE) suite is expensive, and has been released only for Microsoft Windows.[8] Sybase has been praised for performance and portablilty across multiple GUIs, object-oriented programming, heterogeneous database access capabilities and its ability to upgrade.[8] The SQR Toolkit is popular for UNIX users because it is more flexible in its capabilities than other products.[5] Sybase provides the capability for secure data storage. The Sybase server was developed on current technology methodologies, and this leads to a higher level of vendor support. The Sybase vendor also offers numerous training opportunities in several locations

[5] The, Lee, "Tackle Client/Server Apps With Born-Again 4GLs," Datamation, p. 63.

throughout the United States so that potential customers would be able to attend training classes at a location relatively close to their place of business. The Sybase server supports the one back-end server per database architecture as depicted in Figure 1.[7]

Computer Workstations

The computer workstations chosen were Sun SPARCstation LXs which provide the capability for client/server architecture, and use the Sun OS 4.1.3 operating system. Due to declining vendor support, other terminals such as Wangs and IBMs are being replaced at several government locations by Suns supporting Client/Server environments or IBM PCs supporting Local Area Networks (LANs). The price of the terminals, client/server integration features and available COTS products influenced the decision to use Sun SPARC terminals. Since the Sun OS 4.1.3 operating system is based on the UNIX system, user permissions and software control levels are easily established and maintained. In October 1993 Sun declared the end-of-life for the SPARCstation 2 workstations. This means the SPARC 2 workstations are not available for purchases, and will not be supported by Sun in 5 years. For new customers of the DRTool system, Sun SPARCstation LXs or SPARCstation 10s should be selected as the user workstations. The SPARC 10s are more expensive than the SPARC LXs with the same basic performance capabilities, and this makes the SPARC LX the preferred model. These workstations offer the same vendor service contract as the SPARC 10s and support all the current applications being used by the DRTool system. The current customers that have SPARCstation 2s should wait 2 years before upgrading to a newer versions of the SPARC workstations since the costs of the newer workstations will decline considerably during this time.

Computer Display Package

This portion of the DRTool system required a change from the current package being used. The Openlook package of X-Windows was chosen as the development package for user displays. Motif has become the industry standard for window programming and development. Most COTS products being released support Motif, while vendors are not providing enhancements for the Openlook package, which will not be supported in the near future. The user interface panels should be designed using the Motif GUI package.

Synthesis and Definition

The Synthesis and definition phase of the DRTool design effort included the development and preliminary testing of panel prototypes. This provided the customer, users and Program 212 individuals an opportunity for ensuring all understood the DRTool system operational and maintenance requirements. The panels provide the main user interfaces, and panel prototyping enabled fields on the panels to be specifically defined with respect to input types and formats. The create DR panel is shown in Figure 24. The size and data types of the fields on the DR panel were defined, and are listed in Table 1. Additional panels are depicted in Appendix D.

The prototyping of the panels uncovered some minor differences in the use requirements that were defined in the Conceptual and Preliminary Design phases. The effort stressed the importance of maintaining customer and user participation while developing system requirements. The early prototyping and system definition will reduce the amount of changes required in the advanced system development, and this reduces the costs incurred to make these changes since it costs more to make changes later in the system life-cycle due to additional rework and testing.

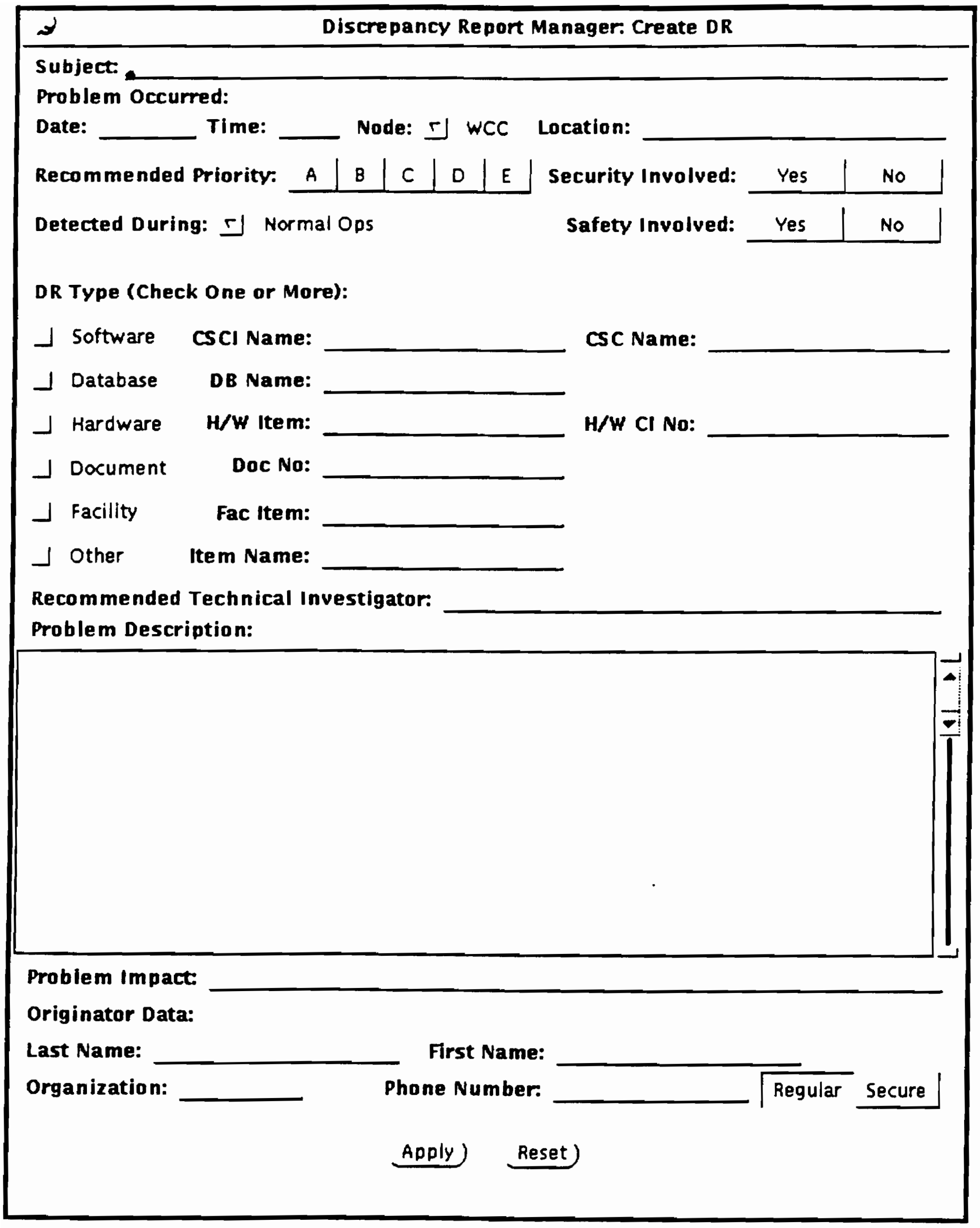


Figure 24: Create DR Panel

Table 1: DR Data Table

Discrepancy Report Data Table

Field	Type	Length
UNIQUE DR ID	CHAR	10
DR NUMBER.	CHAR	10
DATE LOGGED	DATE	8
SUBJECT	CHAR	25
CURRENT STATUS (OPEN/CLOSED)	CHAR	6
DATE & TIME PROBLEM OCCURRED	DATE	8
LOCATION	CHAR	25
DETECTED DURING DESCRIPTION	CHAR	100
SYSTEM PROBLEM OCCURRED ON (SUN, OTHER..)	CHAR	25
S/W PROBLEM (Y/N)	CHAR	3
S/W CSCI	CHAR	25
S/W CSC	CHAR	25
D/B PROBLEM (Y/N)	CHAR	3
D/B NAME	CHAR	25
F/W PROBLEM (Y/N)	CHAR	3
F/W NAME	CHAR	25
H/W PROBLEM (Y/N)	CHAR	3
H/W NOMENCLATURE	CHAR	25
H/W C.I. NUMBER	CHAR	25
DOC PROBLEM (Y/N)	CHAR	3
DOC NAME	CHAR	25
FAC PROBLEM (Y/N)	CHAR	3
FAC ITEM	CHAR	25
OTHER (Y/N)	CHAR	3
OTHER ITEM	CHAR	25
PERSONNEL SAFETY INVOLVED (Y/N)	CHAR	3
ORIGINATOR'S NAME	CHAR	40
ORIGINATOR'S PHONE	CHAR	10
ORIGINATOR'S ORG	CHAR	20
RECOMMENDED PRIORITY	CHAR	1
AUTHORIZED PRIORITY	CHAR	1
PROBLEM DESCRIPTION	UNLIMITED	
PROBLEM IMPACT	UNLIMITED	
SPECIAL PURPOSE IDENTIFIERS	CHAR	25

CONCLUSIONS

The preliminary redesign of Program 212's DRTool system established operational and maintenance requirements that users, customers and Program 212 individuals could use while using and supporting the DRTool system. This included the Conceptual Design Phase and Preliminary Design Phase of the systems life-cycle process. The guidance of Blanchard and Fabrycky, as well as other computer experts have been observed while establishing the system requirements. The users and customer were involved throughout the redesign effort to ensure customer requirements were met. This resulted in requirements that were acceptable to all and a set of preliminary panels that allow the users or DR administrators to execute functions required to edit, read or add to the DR database.

Recommendations for Future Research

There is no on-line help available to the user other than the help panels. The addition of an on-line tutorial would make the system easier for the novice or occasional user. A training manual should also be generated to aid users of the system. The current user's manual provides little information for troubleshooting minor problems or documenting system operability. This may reduce the number of problem calls that Program 212 receives from inexperienced users of the system.

The design effort has improved several aspects of the DRTool system, and the continued use of the systems life-cycle process for additional phases of the DRTool system can only improve the effectiveness and operability of the system. The use of these methodologies in the preliminary design effort has already increased customer satisfaction with the system and Program 212.

REFERENCES

1) Bailey, Robert S., Human Performance Engineering, Englewood Cliffs, NJ, Prentice-Hall, 1989.

2) Blanchard, Benjamin S. and Wolter J. Fabrycky, Systems Engineering and Analysis, Englewood Cliffs, NJ, Prentice-Hall, 1990.

3) Garbus, Jeffrey R., "The Care and Feeding of Your Sybase SQL Server, A Preventive Maintenance Regimen," SQL Forum, Volume 2, Issue 5, September/October 1993, 45.

4) Kernighan, Brian W. and Dennis M Ritchie, The C Programming Language, Englewood Cliffs, NJ, Prentice-Hall, 1988.

5) Leinfuss, Emily, "Client/Server Application Development Tools," InfoWorld, Volume 15, Issue 42, October 18, 1993, 57-60.

6) Lile, Edward A., "Client/Server Architecture: A Brief Overview," Journal of Systems Management, Volume 44, Issue 12, December 1993, 26-29.

7) Rodgers, Ulka, UNIX Database Management Systems, Englewood Cliffs, NJ, Prentice-Hall, 1990.

8) The, Lee, "Tackle Client/Server Apps With Born-Again 4GLs," Datamation, Volume 39, Issue 23, December 1, 1993, 61-65.

Appendix A: System Costs

The DRTool is being demonstrated to potential new customers with the hope of expanding the business opportunities of Program 212. A cost analysis was performed to determine costs for a new system. The cost analysis results will consist of three parts. First, a summary table is provided with the total costs for a new DRTool system for various numbers of users. The table also gives the total costs for the current customer to upgrade to the DRTool components that were recommended in this design. Second, an itemized list of the costs for the various DRTool components is given. Third, the customer cost for a newly installed system for eight users (one DR administrator, seven end users) is calculated for example.

The following assumptions are made to provide a basis from which these cost studies were determined.

1) Program 212's MARR 12%
2) Service life of DRTool system 7 years
3) Taxes and depreciation values are ignored.
4) List prices are used (Vendor discounts not included)
5) No salvage values are expected

New System Costs

Number of Users	Total Costs ($)
2	106,285.32
4	136,677.49
8	197,895.82
12	308,445.55
16	367,916.88

Current System Upgrades

Number of Users	Total Costs ($)
2	31,639.27
4	63,278.54
8	126,557.08
12	189,835.62
16	253,114.16

Itemized Cost List

Qty	Product	Cost($)/Item
1	SPARCstation LX with 19 inch color monitor, 32 MB RAM, 1.05 GB Fast SCSI-2 Disk	12,145.00
1	SunSpectrum Silver Service Contract (/Yr)	64.00
1	Sybase Server (version 4.9.1) supporting Sun OS 4.1.3, including	
	SQL Server	11,880.00
	Open Server	7,130.00
	Open Client	1,490.00
	Embedded SQL	2,970.00
	Data Workbench	1,880.00
	SQL Debug	5,940.00
	SA Companion Server	5,800.00
	SA Companion Client	5,490.00
	Sybase service update fees (/Yr)	
	SQL Server	668.00
	Open Server	401.00
	Open Client	84.00
	Embedded SQL	167.00
	Data Workbench	106.00
	SQL Debug	334.00
	SA Companion Server	326.00
	SA Companion Client	334.00
1	Mega Drive Systems, MR/5 with 10 GB capacity, SCSI/2 Host	9,500.00
1	Maintenance Contract (/Yr)	874.00
1	Hewlett-Packard Laser Printer III	854.00

Qty	Product	Cost ($)
1	Sun 8mm Tape Backup Drive Desktop Storage Module with 5 GB capacity	4,400.00
1	Motif Display version 3.0	2,578.00
1	Transceiver	80.00
1	RG58 Coaxial Cable ($/foot)	1.00
1	Computer Connector	.50

Itemized Upgrade Costs

Qty	Product	Cost ($)
1	SPARCstation 10, model 40s with 19 inch color monitor, 32 MB RAM, 1.05 GB Fast SCSI-2 Disk	11,300.00
1	SunSpectrum Silver Service Contract (/Yr)	168.00

Example Cost Analysis for a New DRTool System

Cost Analysis for a new eight user DRTool system.

Qty	Product	Total Cost ($)
8	SPARCstation LX with 19 inch color monitor, 32 MB RAM, 1.05 GB Fast SCSI-2 Disk	97,160.00
8	SunSpectrum Silver Service Contract (/Yr)	512.00
1	Sybase Server (version 4.9.1) supporting Sun OS 4.1.3, including	
	SQL Server	11,880.00
	Open Server	7,130.00
	Open Client	1,490.00
	Embedded SQL	2,970.00
	Data Workbench	1,880.00
	SQL Debug	5,940.00
	SA Companion Server	5,800.00
	SA Companion Client	5,490.00
	Sybase service update fees (/Yr)	
	SQL Server	668.00
	Open Server	401.00
	Open Client	84.00
	Embedded SQL	167.00
	Data Workbench	106.00
	SQL Debug	334.00
	SA Companion Server	326.00
	SA Companion Client	334.00
1	Mega Drive Systems, MR/5 with 10 GB capacity, SCSI/2 Host	9,500.00
1	Maintenance Contract (/Yr)	874.00
1	Hewlett-Packard Laser Printer III	854.00
1	Sun 8mm Tape Backup Drive Desktop Storage Module with 5 GB capacity	4,400.00

Qty	Product	Total Cost ($)
8	Motif version 3.0	20,624.00
8	Transceiver	80.00
950	RG58 Coaxial Cable (feet)	950.00
18	Computer Connectors	9.00

The initial costs total	180,526.00
The yearly maintenance costs total	3,806.00

The Present Equivalent Cost (PEC) of the DRTool system for seven years is computed as:

PEC = $180,526 + $ 3,806(P/A,12%,7)

PEC = $180,526 + $ 3,806(4.5638)

PEC = $197,895.82

Appendix B: Equipment Requirements

The DRTool system consists of several pieces of hardware for data storage, data backup and user interfaces. A majority of the equipment is listed in Table 2. Specifications such as size, weight, cooling requirements and power requirements are provided. The mean time between failures (MTBF) and mean corrective maintenance time ($\bar{M}ct$) are also provided. Table 2 could be used by customers to determine space and power impacts when purchasing the DRTool system.

Table 2: Equipment Specifications

Equipment	Physical Dimensions			Weight (LBS)	Cooling Rqmts			Nominal Volt/Ph	Break Amps	MTBF X1000	Mct (hr)	MDT (hr)
	Length (IN)	Width (IN)	Height (IN)		Type	Flow (CMF)	Heat Dissip (KW)					
Sun SPARC 2	19.9	18.9	21.5	108	A	CONV	0.4	120/1	20	43	0.9	30
Sun SPARC LX	19.9	18.9	21.5	105	A	CONV	0.4	120/1	20	43	0.9	30
Sun SPARC 10	19.9	18.9	21.7	105	A	CONV	0.4	120/1	20	43	0.9	30
HP Laser Ptr III	14.9	21.0	16.0	46	A	CONV	0.4	120/1	20	38	0.7	30
Sun Tape Bkup	21.0	18.0	12.0	52	A	CONV	0.4	120/1	20	43	0.9	30
Mega MR/5	26.6	14.5	24.0	117	A	CONV	1.45 kbtu/hr	120/1 208/2	20	400	2.1	48
Sybase Server	-	-	-	-	-	-	-	-	-	250	4.0	24
Program 212 SW	-	-	-	-	-	-	-	-	-	12	5.2	24

Appendix C: System Reliability

The DRTool system reliability is based on the assumption that the cables and transceivers are 100% reliable. The DRTool system reliability configuration is shown in Figure 25. The SPARC LX workstations are shown in parallel since the failure of one will not prevent the system from functioning. One of the workstations could be used as a backup for the other workstation. This causes the DRTool system reliability to be computed as follows where:

$$R_{component} = e^{-(1/MTBF)t}$$

The following reliabilities are calculated when t=1000 hrs.

$$R(Database) \times R(Server) \times R(P212\ SW) \times [1 - (1 - R(SPARC))^2]$$

Using the derived numbers:

$$(.9975) \times (.9960) \times (.9201) \times [1 - (1 - (.9770))^2] = .9136$$

The system Operational Availability is calculated as follows, where the system is limited by the Program 212 specialized software.

$$A_o = \frac{MTBF}{MTBF + MDT} = \frac{12000}{12000 + 24} = .9980$$

Figure 23 contains the component performance characteristics and effectiveness factors.

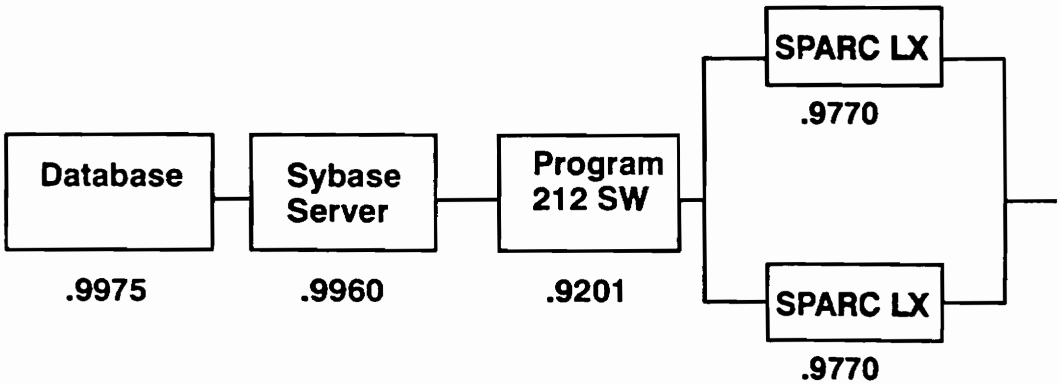


Figure 25: DRTool System Reliability

Appendix D: User Panels

The DRTool system offers several capabilities to the users and DR administrators. During the definition and synthesis phase of the Preliminary Design, several prototype panels were created to determine if the DRTool requirements were well established and understood by all involved in the redesign effort. Some minor changes were suggested, but the overall redesign effort was well accepted by the users, customers and Program 212. Several of the prototype panels are presented for completeness.

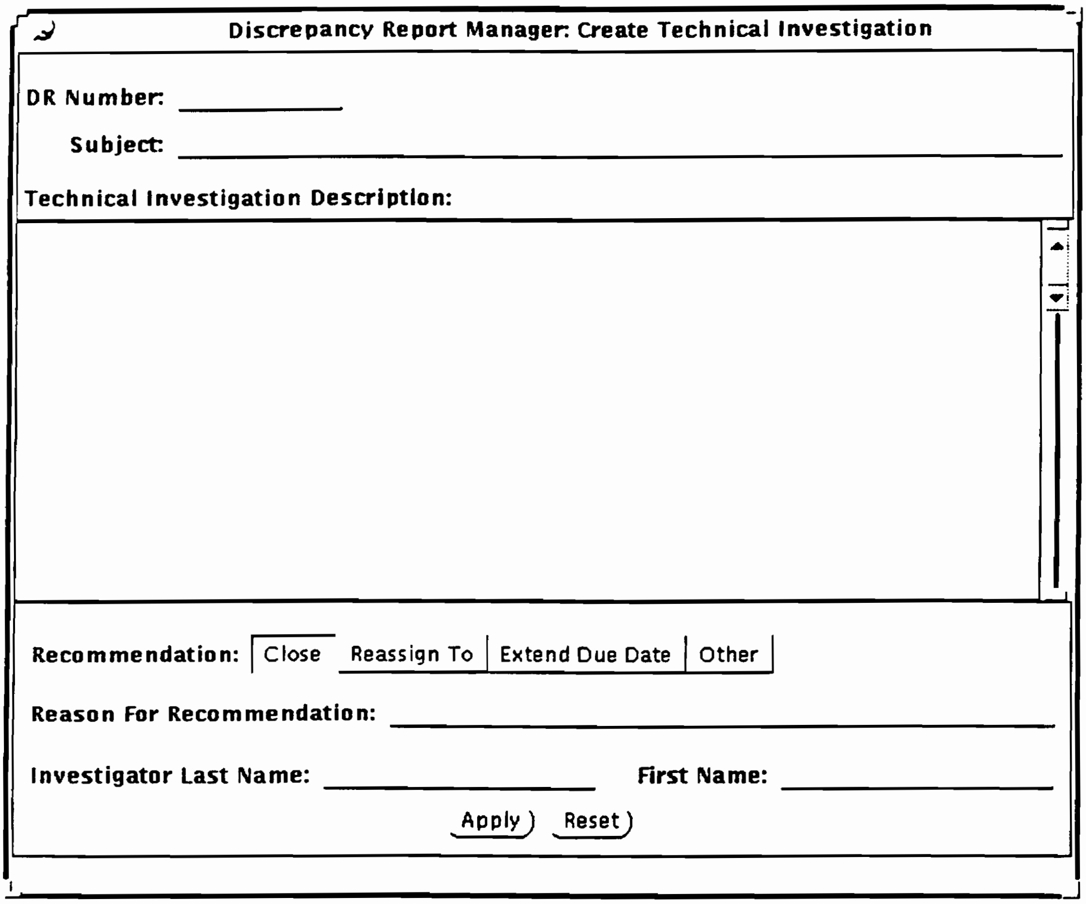
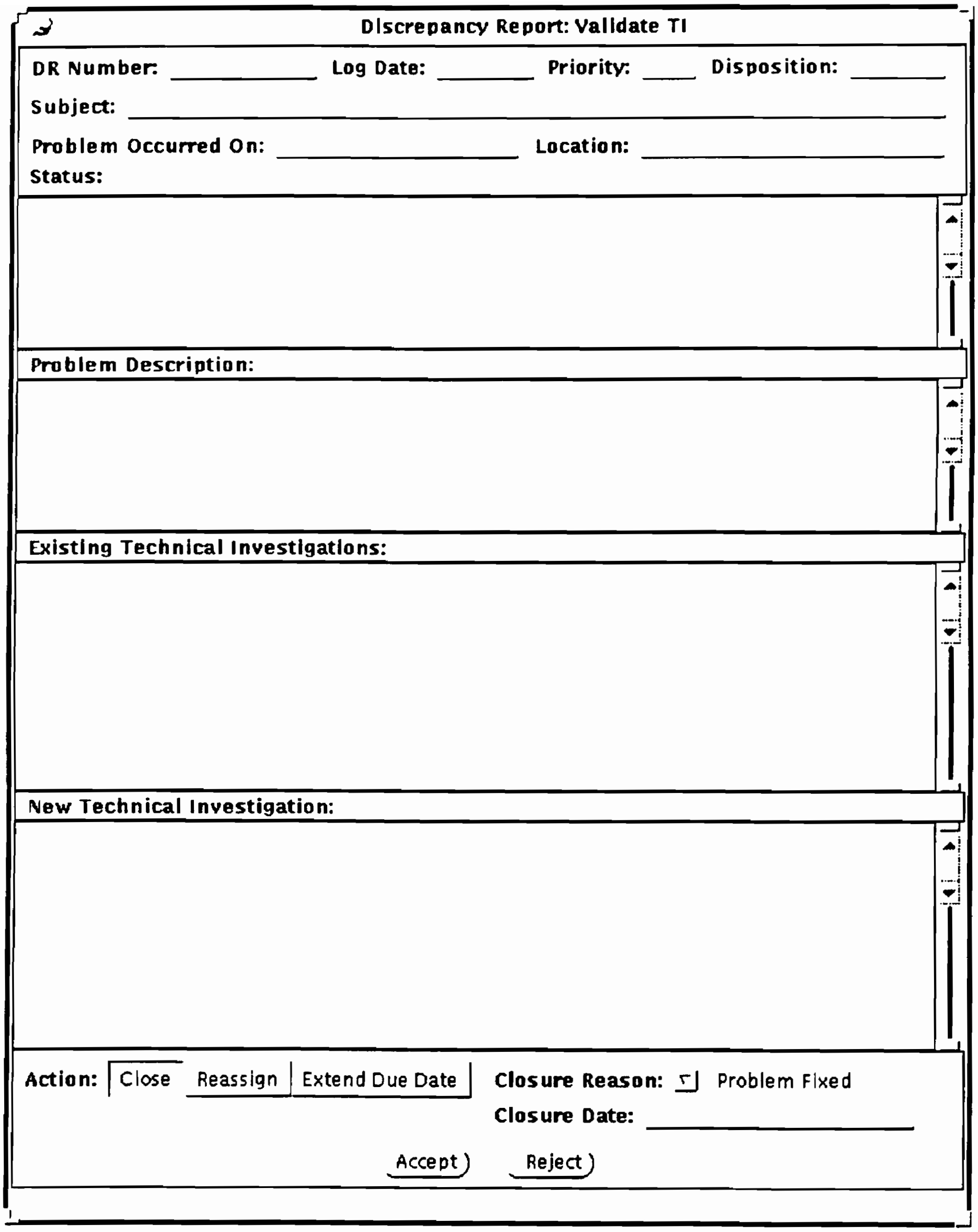


Figure 26: Create Technical Investigation (TI)


Discrepancy Report: Validate TI

DR Number: ______ Log Date: ______ Priority: ______ Disposition: ______

Subject: ______

Problem Occurred On: ______ Location: ______

Status:

Problem Description:

Existing Technical Investigations:

New Technical Investigation:

Action: Close | Reassign | Extend Due Date

Closure Reason: Problem Fixed

Closure Date: ______

Accept Reject

Figure 27: Validate TI

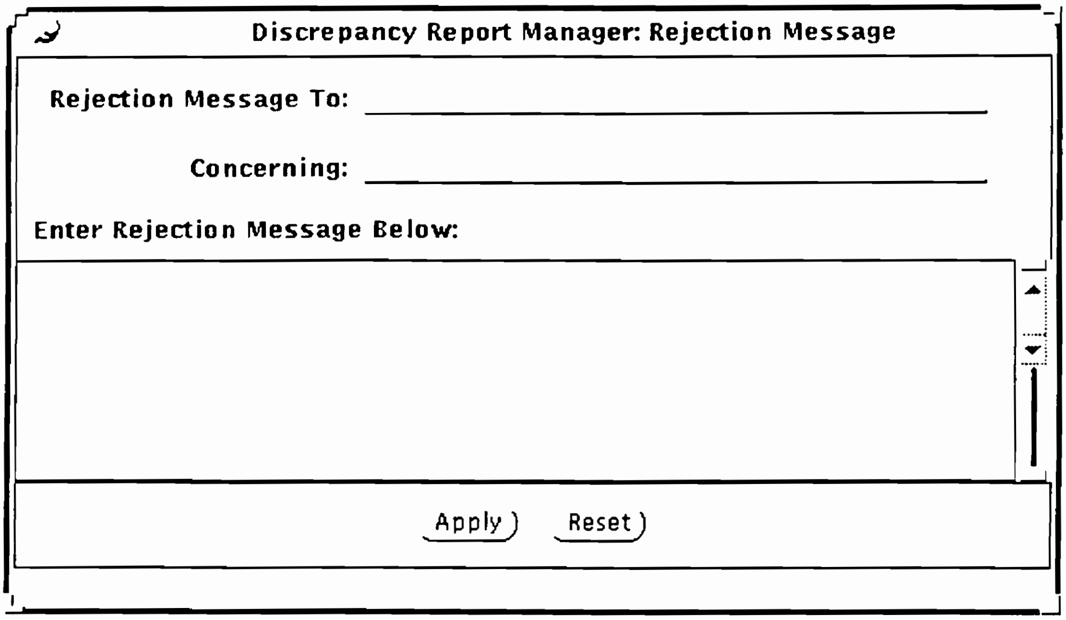


Figure 28: Rejection Message

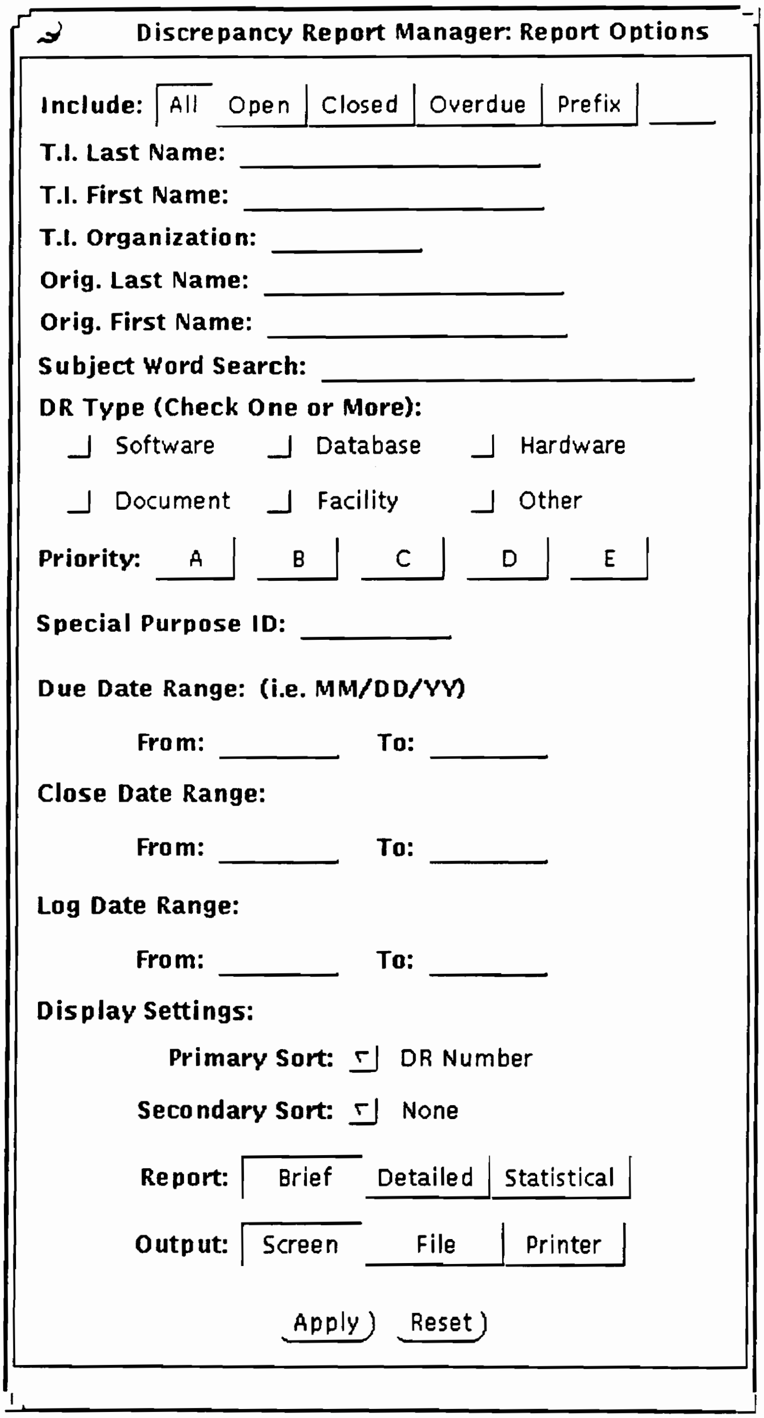


Figure 29: Report Options

Appendix E: Acronym List

A_o	Operational Availability
COTS	Commercial Off-The-Shelf
DASD	Direct Access Storage Device
DBMS	Database Management System
DR	Discrepancy Report
GB	Gigabytes
GUI	Graphical User Interface
LAN	Local Area Network
MB	Megabytes
Mbps	Mega bits per second
Mct	Mean Corrective Maintenance Time
MDT	Maintenance Down Time
Mpt	Mean Preventive Maintenance Time
MTBF	Mean Time Between Failures
PC	Personal Computer
RAM	Random Access Memory
SQL	Structured Query Language
TI	Technical Investigation