RADC-TR-89-165
In-House Report
October 1989

AD-A215 298

RADC FAULT TOLERANT SYSTEM RELIABILITY EVALUATION FACILITY

Joseph Caroli, Frank Fieldson, Jack Hewitt, Matt Plano

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, NY 13441-5700

This report has been reviewed by the RADC Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC TR-89-165 has been reviewed and is approved for publication.

APPROVED:

ANTHONY J. FEDUCCIA
Chief, Systems Reliability & Eng Div
Directorate of Reliability and Compatibility

APPROVED:

JOHN J. BART
Technical Director
Directorate of Reliability and Compatibility

FOR THE COMMANDER:

JAMES W. HYDE III
Directorate of Plans and Programs

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (RBET) Griffiss AFB NY 13441-5700. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document requires that it be returned.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION: UNCLASSIFIED	1b. RESTRICTIVE MARKINGS: N/A
2a. SECURITY CLASSIFICATION AUTHORITY: N/A	3. DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE: N/A	
4. PERFORMING ORGANIZATION REPORT NUMBER(S): RADC-TR-89-165	5. MONITORING ORGANIZATION REPORT NUMBER(S): N/A
6a. NAME OF PERFORMING ORGANIZATION: Rome Air Development Center; 6b. OFFICE SYMBOL (If applicable): RBET	7a. NAME OF MONITORING ORGANIZATION: Rome Air Development Center (RBET)
6c. ADDRESS (City, State, and ZIP Code): Griffiss AFB NY 13441-5700	7b. ADDRESS (City, State, and ZIP Code): Griffiss AFB NY 13441-5700
8a. NAME OF FUNDING/SPONSORING ORGANIZATION: Rome Air Development Center; 8b. OFFICE SYMBOL (If applicable): RBET	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER: N/A
8c. ADDRESS (City, State, and ZIP Code): Griffiss AFB NY 13441-5700	10. SOURCE OF FUNDING NUMBERS: PROGRAM ELEMENT NO. 62702F; PROJECT NO. 2338; TASK NO 02; WORK UNIT ACCESSION NO 2X

11. TITLE (Include Security Classification): RADC FAULT TOLERANT SYSTEM RELIABILITY EVALUATION FACILITY

12. PERSONAL AUTHOR(S): Joseph Caroli, Frank Fieldson, Jack Hewitt, Matt Plano

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
In-House	FROM Aug 88 TO Feb 89	October 1989	102

16. SUPPLEMENTARY NOTATION: N/A

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Fault Tolerance Reliability Modeling Computerized Techniques
13	08		

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The Systems Reliability and Engineering Division of the Rome Air Development Center (RADC/RBE) is in the process of developing a Fault Tolerant System Reliability Evaluation Facility. The facility has the analysis/evaluation capability necessary to perform reliability modeling for virtually any fault tolerant system. RADC/RBE offers modeling expertise as a support service to DOD agencies with fault tolerant system reliability modeling needs.

The facility relies heavily on the use of computer-aided reliability modeling tools. Since no single tool is perfect for every application, RADC/RBE has acquired several. They are CARE III, FASTER, HARP, MIREM, REST, and R&MA^{2}T^2. This functional capabilities plan analyzes the capabilities, strengths and weaknesses of each of the above tools. The overall capabilities of the facility are also discussed and sample modeling cases are provided. A survey of tools was also undertaken as part of this study. Short abstracts are provided for 27 modeling tools both new and old.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT: ☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION: UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL: JOSEPH A. CAROLI	22b. TELEPHONE (Include Area Code): (315) 330-4205; 22c. OFFICE SYMBOL: RADC(RBET)

SPECIAL NOTE

This report documents the work accomplished during a six month internship by three graduates of the Air Force Systems Command (AFSC) Product Assurance Engineering (PAE) intern program. The AFSC PAE intern program was initiated in 1987 in response to the increasing awareness and importance of reliability and maintainability in AF systems. Graduating engineers are recruited and sent to school for one year of post-graduate study in product assurance followed by a six month on-the-job training internship at an AFSC facility. The authors of this report: Frank Fieldson, Jack Hewitt and Matt Plano served a six month internship at RADC in the Systems Reliability and Engineering Division (RADC/RBE) from Jul 88 to Feb 89.

Readers who want more information on the RADC Fault Tolerant System Reliability Evaluation Facility or who have questions/comments on this report should contact Joe Caroli, RADC/RBET.

TABLE OF CONTENTS

	PAGE
1.0 INTRODUCTION	1
2.0 DESCRIPTIONS OF RESIDENT TOOLS	4
2.1 CARE III	6
2.2 FASTER	12
2.3 HARP	16
2.4 MIREM	24
2.5 REST	32
2.6 R&MA^{2}T^2	37
3.0 EXAMPLE MODELING APPLICATIONS	41
3.1 MODELING A DIGITAL FLIGHT CONTROL SYSTEM USING CARE III	42
3.2 MODELING A FAULT TOLERANT RADAR SYSTEM USING HARP	52
3.3 MODELING A MULTIFUNCTION SYSTEM UNDER DIFFERENT REPAIR PHILOSOPHIES USING REST	61
3.4 MODELING A LOCAL AREA NETWORK (LAN) USING R&MA^{2}T^2	64
4.0 OTHER RELIABILITY MODELING TOOLS FOR FAULT TOLERANT SYSTEMS/TOOL SURVEY	69
5.0 SUMMARY AND CONCLUSION	86
B.0 BIBLIOGRAPHY	B-1

LIST OF TABLES AND FIGURES

		PAGE
FIGURE 2-1	Characteristics of Fault Tolerant System Reliability Modeling Tools Resident at RADC Evaluation Facility...	5
FIGURE 2-2	CARE III General Fault Handling Model.....	7
FIGURE 2-3	FASTER Modeling Methodology.....	13
FIGURE 2-4	MIREM Program Structure.....	26
FIGURE 2-5	MIREM Terminology.....	27
FIGURE 2-6	Sample System to Illustrate REST.....	34
FIGURE 3-1	Digital Flight Control System Rel Block Diagram.....	42
TABLE 3-1	Fault Handling Model Parameters For Digital Flight Control System.....	44
TABLE 3-2	Fault Occurrence Model Parameters For Digital Flight Control System.....	45
FIGURE 3-2	System Fault Tree For Digital Flight Control System...	47
FIGURE 3-3	Fault Tree Representation For Critical Pairs.....	48
FIGURE 3-4	CARE III Output for Digital Flight Control System Analysis.....	50
FIGURE 3-5	Radar System Reliability Block Diagram (RBD).....	52
FIGURE 3-6	Fault Tree Corresponding To Figure 3-5 RBD.....	53
TABLE 3-3	Power Supply FEHM.....	55
TABLE 3-4	Remote Processor FEHM.....	55
TABLE 3-5	Modem FEHM.....	56
TABLE 3-6	Results of HARP Analysis on the Remote Radar System with Imperfect Fault Coverage.....	58

LIST OF TABLES AND FIGURES Continued

		PAGE
TABLE 3-7	Results of HARP Analysis with Perfect Fault Coverage..	60
FIGURE 3-7	System Configuration for REST Example.....	61
FIGURE 3-8	Set Configuration for REST Example.....	61
FIGURE 3-9	"Function 2" Configuration for REST Example.....	62
FIGURE 3-10	Local Area Network Functional Flow Diagram.....	65
FIGURE 3-11	Local Area Network Block Diagram.....	65
FIGURE 3-12	Sample R&MA^{2}T^2 Coding.....	66
FIGURE 3-13	LAN R&MA^{2}T^2 Output.....	68

1.0 INTRODUCTION

Modern day systems in various applications such as space, flight control and computer networks often employ fault tolerance in order to obtain ultra high reliability. Many computer-aided reliability modeling tools for fault tolerant systems are currently available. The tools have several unique capabilities as well as limitations (i.e., ease of use, R&M figures of merit evaluated, input parameters, and mission scenarios considered). Use of the tools require a moderate to large computational facility and a good working knowledge of the modeling rationale and procedures embedded in each. RADC/RBE has developed a Fault Tolerant System Reliability Evaluation Facility consisting of a number of these tools. The required resources are now available at RADC to model a wide range of fault tolerant systems and to assess and evaluate the reliability characteristics for such.

The purpose of the facility is to provide computer-aided R&M modeling expertise in support of DoD system design and development activities. It is intended that the facility resources will serve to verify the results of contractor applied R&M modeling tasks and to help conduct design tradeoff analyses. Design tradeoffs involve determining the appropriate degree of fault tolerance required to achieve a desired availability, mean-time-between-critical failure (MTBCF) or reliability over a given mission time. The assessment of different fault tolerant strategies can be accomplished to aid in conceptual design studies, engineering changes or demonstration and validation activities.

This report is intended to give an overview of the facility. The capabilities and potential applications of the six software packages which are part of the facility are outlined within. Section 2 provides individual synopses of the resident tools depicting the capabilities, strengths and weaknesses of each. Also found in this section is a matrix comparing and contrasting the various attributes inherent to the tools. Section 3 contains example modeling cases that were conducted using some of the tools. Section 4 documents a survey of other software tools that are available for reliability modeling of fault tolerant systems. In the future some of these will also be evaluated for inclusion in the facility. Both old and new tools were surveyed. Short abstracts of 27 tools are found within. Reliability modeling of redundant/fault tolerant systems has received a great deal of attention for many years. There is a need for standardization in this area. Even though many of the tools have unique capabilities, many also have overlapping capabilities. New modeling tools and techniques are always appearing in government, industry and academia. The wheel is constantly being reinvented. A more standardized approach would solve this problem and would center the focus on developing needed modeling technology as opposed to duplicating that which is already existing. (At the present time, models do not exist to handle every fault tolerant system conceivable in a timely and cost efficient manner.) To give an idea of how much work was done in this area, consider reference #13, RADC-TR-77-287, "A Redundancy Notebook" - this document was written in 1977 and it addresses 32 methods of modeling. Hundreds of papers on this topic have appeared in the International Reliability & Maintainability Symposium Proceedings over the past 10-15 years. There are many knowledgeable people in this area, off working in their own direction. We

stress the need for a unified, standardized approach within the DoD. Section 5 summarizes and concludes this report.

The following tools are presently resident in the facility.

NASA "CARE" - Computer-Aided Reliability Estimation

RADC "FASTER" - Fault Tolerant Architecture Simulation Tool for Evaluating Reliability

NASA "HARP" - Hybrid Automated Reliability Predictor

AFHRL "MIREM" - MIssion REliability Model

RADC "REST" - REliability Simulation Tool

RADC "R&MA^{2}T^2" - Reliability, Maintainability and Availability Analysis and Tradeoff Tool

The facility presently has the capability to model many different fault tolerant system configurations ranging from a simple series - parallel configuration to a complex path repairable system with imperfect fault coverage. Various R&M figures of merit can be computed such as reliability; lower and upper bounds on Reliability; Availability; Mean-Time-Between-Critical-Failure (MTBCF) under various repair scenarios; Mean-Time-Between-Maintenance-Action (MTBMA); Mean-Time-To-Repair (MTTR); and others.

2.0 DESCRIPTIONS OF RESIDENT TOOLS

This section provides an independent summary of each resident tool in the RADC Fault Tolerant System Reliability Evaluation Facility. Figure 2-1 is a matrix outlining the resident software tools and their specific modeling traits. The matrix along with the individual summaries can be used to help determine the most appropriate modeling tool(s) for a specific fault tolerant system configuration/scenario.

FAULT TOLERANT SYSTEM RELIABILITY MODELING TOOL

ATTRIBUTE	CARE III	HARP	MIREM	R&MA^{2}T^{2}	REST	FASTER
Developer/Sponsor	NASA Langley	NASA Langley	AFHRL	RADC	RADC	RADC
System Size	Large	Small-Med	Large	Med-Large	Small-Med	Large
Maturity/Verification	Mature/Verified	Rel New/Und Ver	New/Und Ver	Mature/Ver	New/Und Ver	New/Und Ver
Results/Outputs	Unreliability for hardware depletion Unreliability for Imperfect fault Handling	Reliability Phase-by-Phase Reliability Rel Bounds State Prob & associated error terms Rel for imperfect fault handling	Reliability Phase-by-Phase Reliability MTBCF under various repair scenarios MTBF Rel Bounds MTBMA	MTFF Steady-State Mean Time Between Fail MTTR Availability	Reliability MTBCF Availability MTTR - non-concurrence repair durir a downtime MIL-STD-781 decision risks	Reliability MTBCF Availability MTBF
Model/Solution Type	Semi Markov/ Numerical Integration	Non-Homogeneous Markov/Numerical Integration	Rel Equations & Algorithms	Rel Equations & Algorithms	Monte Carlo Simulation	Monte Carlo Simulation
Architecture Description	Fault Tree	Fault Tree or System State Diagram	Reliability Block Diagram	Reliability Block Diagram	Reliability Block Diagram	Reliability Block Diagram
User Friendliness	Very Friendly	Very Friendly	Friendly	Very Friendly	Friendly	Friendly
Failure Rate Distn	Exponential or Weibull	Exponential or Weibull (for non-repairable Sys)	Exponential	Exponential	Exponential	Exponential or Weibull
Fault Handling/ Imperfect Coverage (Testability)	Separate Model which is independent of system size	Choice of 7 models one of which is a simulation	Handles Frac of False Alarms and Fraction of Faults Detect.	No	Accounts for failure rates & modes of switching & diagnostic circuitry for limited configurations	Flexibility will allow for different types of user created fault handling
Repairable Systems	No	Yes	Yes	Yes	Limited at this point	Yes
Periodic Maintenance	No	No	Yes	No	Yes	Yes
Repair at Degraded Level	No	No	Yes	No	Yes	Yes
Dissimilar Redundancy	Yes	Yes	No	Yes	Yes	Yes
Standby Redundancy	No	Yes-Markov Chain Only	Yes	No	Yes	Yes
Partially Powered Redundancy (flexed spares)	No	No	No	No	Yes	
Complex Path Sets (Nonseries-Parallel)	Yes	Yes	No	No	Yes	Yes
Critical Pairs	Yes	Yes	No	No	No	Yes
Internal/Multiple Levels of Redundancy	Yes	Yes	Yes-Limited to 2 Levels without "Group" feature	Yes	Yes	Yes
Graphics Capabilities	With DISPLA or TEMPLATE Package can graph time vs unrel. & some coverage functions	No	Yes	No	No	No
Handles Distributed Processing Architect. i.e. LANs, Ring Network, Star etc.	Yes	Yes(nonrepairable)	No	No	Yes	Yes

Figure 2-1
Characteristics of Fault Tolerant System Reliability Modeling Tools Resident at RADC Evaluation Facility

2.1 CARE III

CARE III (Computer-Aided Reliability Estimation) is a fault tolerant system reliability analysis software tool codeveloped by the NASA Langley Research Center and the Raytheon Corporation. Although CARE III was originally designed to analyze digital flight control systems, it is applicable to a wide range of very large, ultra-reliable, fault tolerant systems. The CARE III model solution technique is Markov Analysis.

The input file for CARE III is generated by using the CARE3MENU user-friendly interface program. CARE3MENU is a menu driven program which is designed to reduce input time and user errors. The input of a CARE III system model is broken into the following procedures:

- Stage descriptions
- Fault handling models
- Fault occurrence models
- Fault-tree descriptions
- Output control/format selections.

The system to be analyzed is described by breaking it into stages. Each stage is composed of N identical modules, of which M must be functioning for the system to function.

Internal redundancy can be modelled in the stage description section of the input. This is when a module contains a group of redundant components or submodules. Using this feature can greatly reduce input time and computation time for more complex systems.

After the stage descriptions, the user can input up to five fault handling models for the system. The CARE III general fault handling model is shown in Figure 2-2 (reproduced from P. 11, CARE III Model Overview & User's Guide).

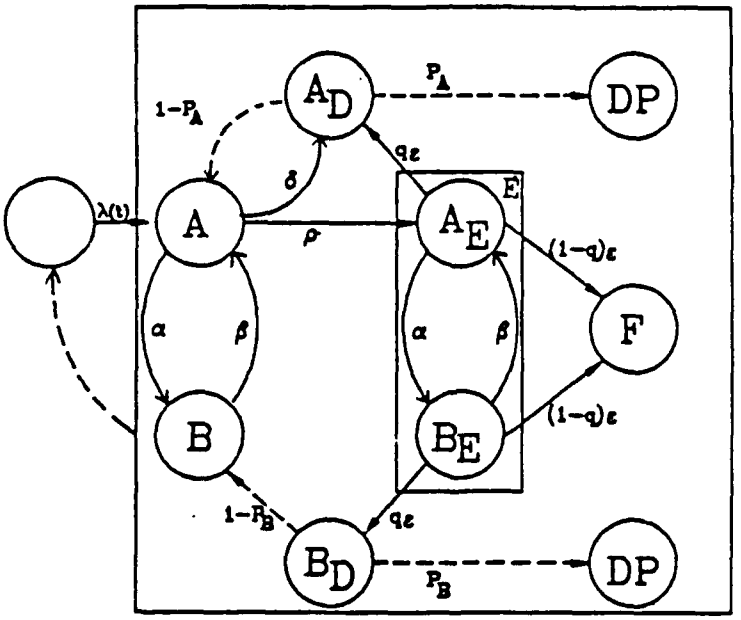


FIGURE 2-2

CARE III General Fault Handling Model

Model States

A - Active Fault - Latent fault state with no errors

A_D - Active Detected - Detected fault/error state

A_E - Active Fault Error - Latent fault and error state

DP - Detected as Permanent - Module permanently isolated from system

F - System Failure

B - Benign fault state, no errors

B_E - Benign fault, latent error state

B_D - Benign Fault, detected error

Fault-handling Model Variables

$\delta(t')$ = Self-test rate (exponential or uniform)

$\rho(t')$ = Error generation rate (exponential or uniform)

$\epsilon(\tau)$ = Error detection rate (exponential or uniform)

∂ = Intermittent or transient duration rate (exponential)

β = Intermittent benign-to-active rate (exponential)

C = Error recovery probability

P_A = Retire module (active fault) probability

P_B = Retire module (benign fault) probability

NOTE: ∂ = 0 for permanent faults

β = 0 for permanent or transient faults

The next step is to define fault occurrence models for each stage. A fault occurrence model consists of a failure density function, which can be either an exponential or a Weibull distribution, and the associated fault handling model. Up to five different fault occurrence models (i.e. permanent, transient, intermittent) may be defined for each stage. For a stage with internally redundant modules, fault occurrence models must be defined for the submodules and for the nonredundant portion of the modules.

The system configuration is described by building a system fault tree. The fault tree describes all possible stage failure combinations which lead to system failure. This notation can be used to model either the common series/parallel combinations or more complex configurations which cannot be represented as a serial or parallel configuration.

CARE III also allows for the definition of "critical pairs". A critical pair is a pair of modules, either in the same stage or in different stages which can cause a system failure if they both contain a fault. A critical pair fault tree is input to define the critically coupled modules.

In the final input procedure, the user sets the system's mission time and various output control parameters.

The CARE III output file gives the user the unreliability (probability of failure) for each stage at the user defined mission time t. The system unreliability at time t is calculated and output in two parts -unreliability due to module depletion and due to imperfect fault handling. The total unreliability is also included in the output.

The user can also obtain plots of system unreliability (both components and the total) from time 0 to mission time t by interfacing with one of two commercially available graphics packages: DISSPLA or TEMPLATE. At the time of this writing, RADC does not have this capability.

CARE III offers the user the following capabilities and advantages:

- The user-friendly interface, CARE3MENU, makes inputting a model quick and easy, and helps minimize input errors. It also allows the user to alter existing models, making corrections or tradeoff analyses easy.

- CARE III can be used to model very large systems on the order of 10^6 Markov states.

- Exponential and Weibull failure distributions can be used. The Weibull distribution allows the user to model wearout failures common to mechanical and some electronic components.

- The user can define up to five fault occurrence models per state and up to five fault handling models per system. This allows the user to model many types of faults (permanent, transient, intermittent, software faults, etc.).

- The fault tree notation used to model the system configuration allows concise descriptions of very large, complex systems.

However, the use of CARE III has the following limitations and disadvantages:

- CARE III cannot be used to model repairable systems.

- CARE III cannot model systems with standby redundancy. Only "hot spares" which have the same fault occurrence and fault handling characteristics as the in-use module can be modelled. The only difference is that a hot spare cannot contribute to a critical pair system failure.

- The fault handling model is defined in terms of parameters like the self-test rate, error generation rate, error detection rate, etc. These parameters are not always easy to quantify, so caution should be taken to make conservative estimates when exact data is not available.

- CARE III does not calculate MTBF or MTBCF, which are reliability parameters often called out in a specification or contract requirement. CARE III results must be used as inputs to a numerical integration to estimate MTBF or MTBCF.

- The CARE III mathematical model includes some assumptions and approximations which introduce some uncertainty in the accuracy of the calculations. CARE III does not give any estimate of the possible error.

2.2 FASTER

FASTER (Fault Tolerant Architecture Simulation Tool for Evaluating Reliability) is a brand new tool that was recently developed by Sanders Associates under an RADC contract. FASTER uses Monte Carlo simulation to compute various reliability figures of merit for complex systems. The results of the FASTER program can be summarized in several different fashions. The nature of the output is dependent on the specific details or properties that the user wishes to obtain. FASTER uses a "timer probe" approach which allows the user to select certain system variables to be monitored. In any given simulation, there are many possibilities to examine. Such possibilities include: system and subsystem MTBF, reliability, availability, MTBCF and specific figures of merit relative to various operating and failure modes inherent to the given system.

The "timer probe" is used to determine the length of time a particular system or subsystem is in a specific mode (i.e. on, off, standby, degraded, failed, etc.). "Timer probes" can be placed at the end of the system to measure overall system "up time" as well as on internal subsystems to obtain dependency information. The probes can be set up to compute the various figures of merit desired.

Complex systems are represented by using a hierarchial building block approach. A block diagram is used to model subsystem interaction. Each subsystem has a mode graph and each mode in the mode graph has a "functional" transfer function. The combination of state transitions and "functional" transfer functions form the basic unit of subsystem representation. A subsystem is referred to as a primitive. A complex

system can be composed of many primitives which interact through connections or interfaces.

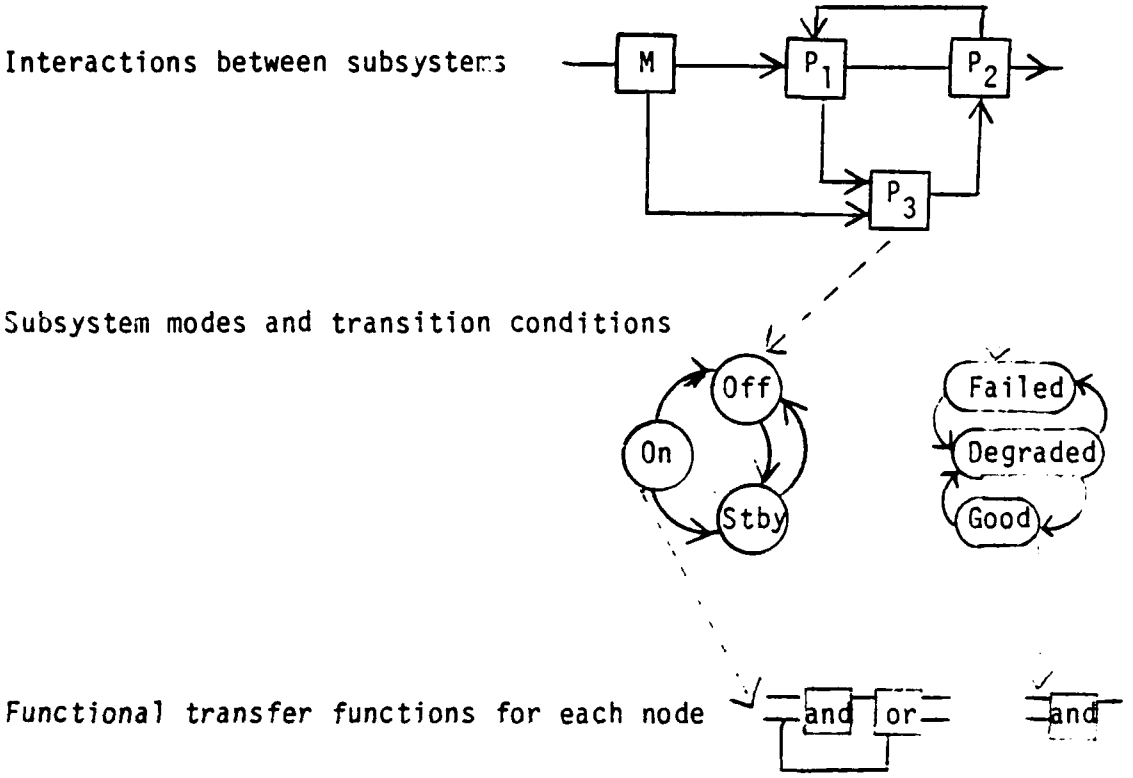


FIGURE 2-3

FASTER Modeling Methodology

Figure 2-3 illustrates a sample system with three interacting subsystems (primitives P_1, P_2 and P_3). The "M" primitive signifies Mission and is necessary for every modeling activity. The subsystem modes for primitives P_2 and P_3 are illustrated as well as some functional transfer functions. Functional transfer functions serve to "wire up" the system diagram so as to take into consideration all system and subsystem interactions. These interactions are described in terms of input and output resources. Subsystems may not be able to perform their functions if they

do not receive the proper input resources from other subsystems. This feature allows a "rich" description of subsystem interoperability and a near exact relation to a mission scenario. This also allows for the consideration of feedback from one subsystem to another. Feedback is used to model imperfect switching testability features.

This method of system representation allows the user to use a system engineering approach. The FASTER system model closely relates to the actual system being simulated. This makes it easy for the user to understand the modeling process and to see how the relationships between subsystems impact overall system performance.

There are two editors in FASTER which are used to define a system for simulation. These are the "primitive editor" and the "top level editor". The "primitive editor" is used to define system primitives which represent the subcomponents of a system to be simulated. The "top level editor" is used to combine primitives together to form the overall system description. The primitive editor obtains this information from the user. For "exit conditions", the primitive editor requests appropriate information which is dependent on the specific exit condition selected. For example, if the exit condition is a "failure", the editor requests which failure equation is to be used. If a constant failure rate is selected, the editor then requests a single number relating to the MTBF. If a non-constant distribution is used, the editor will prompt the user for a set of numbers which represent the distribution of failures. Thus, the editor guides the user by identifying what type of information is needed.

The primitive editor also requires that the user describe a functional transfer function for each mode in the graph. A functional transfer function is formed by selecting a set of operators from a list of simple logic and threshold functions. These operators are connected together to form the functional transfer function. To assist the user, the primitive editor prompts the user for connection data which describes how to combine the selected elements. The resulting primitives are stored on a disk file for later use by the high level editor. An example of the operation of the primitive editor is presented in Section 3.

FASTER also has a mission editor which is used to define a scenario or mission for the system being simulated. The user specifies the "external" inputs to the system being simulated. Examples of external inputs are control inputs (which turn the system on and off) and mission load.

The "top level editor" is used to combine primitives together to form the system to be simulated. In a fashion similar to the primitive editor, it generates displays which indicate what information or inputs are to be supplied to the user. This information deals with connecting and interfacing the primitives together.

The "simulation engine" is the heart of FASTER. This is where subsystem and system failures are randomly generated. The simulation type is Monte Carlo. Details of the "simulation engine" are not included in this writing.

2.3 HARP

HARP (Hybrid Automated Reliability Predictor) is a reliability prediction software package developed by Duke University under sponsorship of NASA Langley Research Center. The software package uses a concept called behavioral decomposition in its solution technique. That is, the system is divided into submodels according to behavior rather than structure. Each submodel is solved individually, and the results are integrated to obtain the system reliability. The HARP model solution technique is a Markov analysis. The user's input is converted to a Markov chain, which produces a series of ordinary differential equations. These differential equations are solved to generate the state probabilities of the Markov chain. The reliability is then determined from the sum of the state probabilities of the system's operational states.

The two submodels utilized by HARP are FORM (Fault Occurrence and Repair Model) and FEHM (Fault/Error Handling Model). FORM is the model which contains information concerning system architecture such as series-parallel configuration, redundancy, failure rates, and repair rates. FEHM stores data on fault coverage such as parameters for imperfect switching, false alarms, and error detection. Use of the Fault/Error Handling Model is optional, allowing the user to model systems with the assumption of perfect coverage for all faults in redundant modules. The user would then be assuming that all faults are hard faults, and that the system will switch perfectly to a new operational configuration after each fault. Therefore, HARP allows the user to model simple systems and complex systems using the same solution technique.

The fault-occurrence and repair model is input to the program in the form of either a fault tree or a Markov chain (for repairable systems). In the case of a fault tree, each component or module is entered as a basic event. Basic events are entered with replication factors, so a group of similar parts can be entered at the same time. These basic events all feed into logic gates at the next level. Logic gates used are AND, OR, and K/N gates. AND gates are used for parallel configurations while OR gates are used for series configurations. K/N gates are used where K failures of N parallel components will result in failure of the parallel set.

Markov chains are entered by states. In any given system, all possible operational and failure states must be considered and accounted for. Each state is entered along with the probability of entering another state (known as a "state transition"), either by failure of a component or by repair of a component. A Markov chain must be used (instead of a fault tree) if the system is repairable or if the system has standby redundancy.

When the user wants to account for the effects of imperfect fault coverage, a fault error handling model is included. This can be used with either the fault tree or Markov chain FORM. HARP automatically converts fault trees into Markov chains for solution. The FEHM is then placed between states in the Markov chain to account for the probability of transients, intermittents, and errors in detecting hard faults. Different FEHMS and parameters may be used at each state transition to allow for different coverage characteristics at different locations in the systems configuration. Based on the user's input of imperfect coverage

parameters, the FEHM calculates at each transition the following probabilities:

1. R - Probability of restoring the system after correctly recognizing the fault as a transient and before a second interferring fault occurs.

2. C - Probability of reconfiguring the system successfully after recognition of a permanent, intermittent, or transient fault, where the transient is mistaken as permanent.

3. S - Probability of system failure due to a single point failure.

4. N - Probability of system failure due to the occurrence of a second fault before one of the previous 3 exists is reached. This is referred to as a "near coincident fault".

When a fault occurs, the FEHM is entered at the state transition and an exit is chosen based on the above probabilities.

HARP offers the user a choice of seven FEHMS to calculate the exit probabilities. Each one can be used to model a different type of coverage situation. Following is a list of the models:

1. ESPN - Extended Stochastic Petri Net
2. CARE III - Computer-Aided Reliability Estimation
3. Probabilities and Distributions
4. ARIES - Automated Reliability Interactive Estimation System
5. Probabilities and Moments
6. User - Defined Exit Probabilities
7. Perfect Coverage

The user specifies a FEHM to go with each component type in the system. HARP automatically calculates the exit probabilities and the distribution of time to exit. The probability of a near coincident fault is also calculated. These probabilities are figured into the FORM to reflect the impact of coverage on the state transitions in the Markov chain.

The Extended Stochastic Petri Net is the HARP default FEHM. ESPN is a simulation which temporarily assumes all faults are transient and simulates the recovery process. User inputs include distribution of time for each simulation activity, probabilities of correct error detection, fault detection, fault isolation and reconfiguration. Also included in user input are the number of transient recovery attempts, the percentage of transient faults, the desired simulation confidence level, and the desired percentage of error. The coverage factors are derived from the probabilities of reaching points in the simulation that correspond to the different factors.

The CARE III model is the Markov process FEHM described in the CARE III section of this report.

Probabilities and Distributions are simply user-defined exit probabilities for transient restoration, permanent coverage, and single point failure. Time to exit distributions for each of these probabilities are also input and used to determine the probability of a near coincident fault. HARP calculates the coverage factors (including the effect of near coincident faults) that are reduced to a branch point in the Markov chain.

The Probabilities and Empirical Data option also utilizes user-defined exit probabilities. However, the time to exit distributions are determined from user-supplied histograms of time to exit data. Again, the coverage factors are reduced to a branch point in the Markov chain.

The ARIES coverage model is a phase-by-phase transient recovery process. The user supplies the number of phases, the phase duration, and the effectiveness at each phase (duration and effectiveness are constant for all phases). If a phase proves effective at detecting a transient fault, normal processing resumes. If it is ineffective, the model goes on to the next phase. If all phases are completed, indicating a nontransient fault, permanent fault recovery is initiated. System failure occurs when a phase is ineffective at detecting a transient and ineffective at proceeding to the next phase. The probability of a near coincident fault is determined from the phase duration and the number of phases needed for permanent fault recovery or transient restoration.

Probabilities and Moments is similar to the Probabilities and Distributions model in that the user defines exit probabilities for permanent coverage, transient recovery, and single point failure.

However, instead of specifying distributions for time to exit, the user inputs the first three moments or points in time when each of the three exits can be made. Near coincident fault probability is then derived from these times to exit and exit probabilities.

The user also has the option of specifying perfect coverage for a specific component type or for all component types. Additionally, if a certain transition in the Markov chain is characterized by a specific FEHM different from the component type FEHM, the component type FEHM may be overridden for that transition. In this way, the user has the ability to define a different FEHM for every transition in the Markov chain.

User inputs to HARP are varied. For a simple perfect coverage model, the only required inputs are failure rates, repair rates, and system architecture in the form of a fault tree or Markov chain. The configuration of the system can be input textually or graphically (on a Vectrix or IBM PC AT). The failure and repair rates can be constant, $\pm$ variation (for repairable systems), exponential (for nonrepairable systems), or Weibull (also for nonrepairable systems). As mentioned before, however, repairable systems can only be modeled with a Markov chain. Both the fault tree and Markov chain FORM's are stored in data files so that changes can easily be made. For an imperfect coverage model, the additional user inputs are those required by the FEHM for each component type. An option to eliminate consideration of near coincident faults is given to the user, and this greatly reduces the number of input parameters.

HARP's output is expressed as reliability at a specific time. This time can be divided so that the reliability is given incrementally over a period of time. The reliabilities and times are stored in a data file as x and y coordinates that can be used to plot the reliability as a function of time. Also provided at the user's option is a listing of reliability bounds (upper and lower) that are calculated using a parametric sensitivity analysis. For Markov chain FORM's, HARP provides state probabilities for any states of interest to the user. Failure state probabilities are automatically provided, as well as the probabilities of single point failure and failure by redundancy exhaustion.

HARP offers the user the following capabilities and advantages:

- Ease of input
- Ease of parameter changes
- Ability to model repairable systems
- Ability to model complex path sets
- Ability to model dissimilar redundancy
- Ability to model wide range of fault handling characteristics

Unfortunately, HARP has the following limitations and disadvantages:

- Markov chains are required to model repairable systems and standby redundant configurations. This can be cumbersome, if not impossible for anything but small systems
- Markov chains are limited to 10,000 states
- No scheduled maintenance scenariors

- Failure rates must be constant for repairable systems
- Not useful for steady state analysis
- Model is sensitive to stiffness (relative difference between failure rates)

2.4 MIREM

The MIREM (Mission Reliability Model) was developed by members of the Georgia Institute of Technology and The Analytic Sciences Corporation (TASC) in accordance with work sponsored by the Air Force Human Resources Laboratory (AFHRL) from the period of March 1982 to March 1986. MIREM is a fault tolerant system reliability modeling tool which accounts for the impacts of redundancy, self repair, scheduled maintenance, imperfect Built-In Test (BIT) and switching, and various repair scenarios. It was developed to evaluate mission reliability, sustained operating capability and availability of electronics systems. MIREM was initially developed to model the Integrated Communication, Navigation and Identification Avionics (ICNIA) System. The ICNIA system makes use of modular avionics to integrate many functions into a highly reconfigurable design. MIREM is most useful for modeling systems similar to ICNIA, but can be used for other applications as well.

The program can best be described by explaining its structure and the basic terminology used for inputting a system architecture.

The program structure consists of two basic programs, a data entry program and a computational program. The data entry program consists of two main files. The first file is an architectural file which is used to describe the system configuration. The second file is a scenario file which describes the mission and the basic run parameters. Once the data entry program creates the two files, the computational program then uses them to compute the MIREM outputs (Reliability, MTBCF, etc.). An overview

of the program structure is shown in Figure 2-4. This figure was extracted from Page 2 of the "Mission Reliability Model Users Guide".

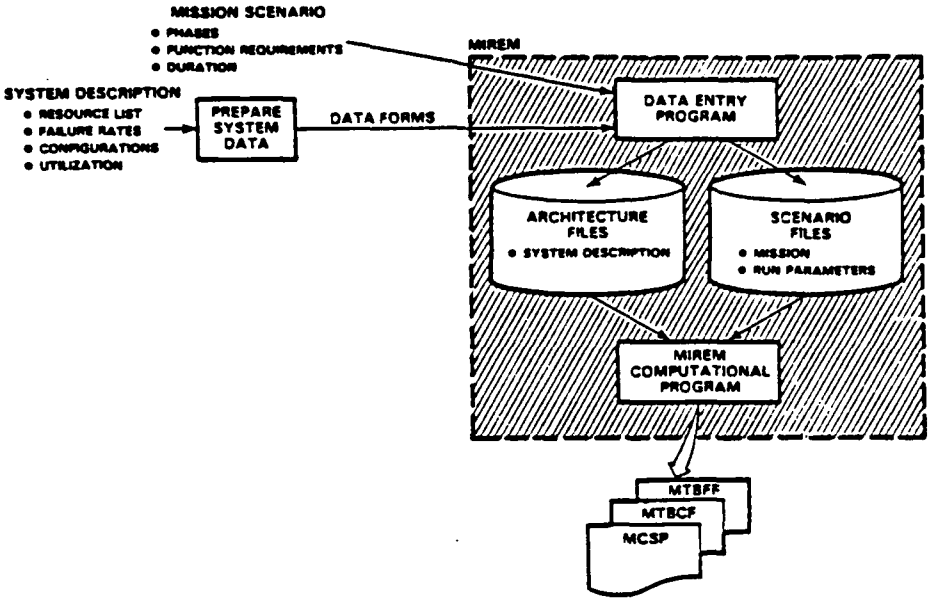


FIGURE 2-4

MIREM Program Structure

To effectively use the MIREM program, a user must have a good grasp of the basic terminology utilized in describing a system architecture. A resource is at the lowest level and is the basic unit for a system structure. Each unique resource is assigned a failure rate and a mean-time-to-repair along with other characteristics which will be discussed in the input section of this report. A pool is a group of resources arranged in a parallel manner. Branches are alternate identical paths within a specified pool. Each branch contains one or more resources in series. A chain is a group of resources and/or pools in a series configuration. A pictorial display of the MIREM terms is displayed in Figure 2-5. This figure was extracted from page 10 of the "Mission Reliability Users Guide".

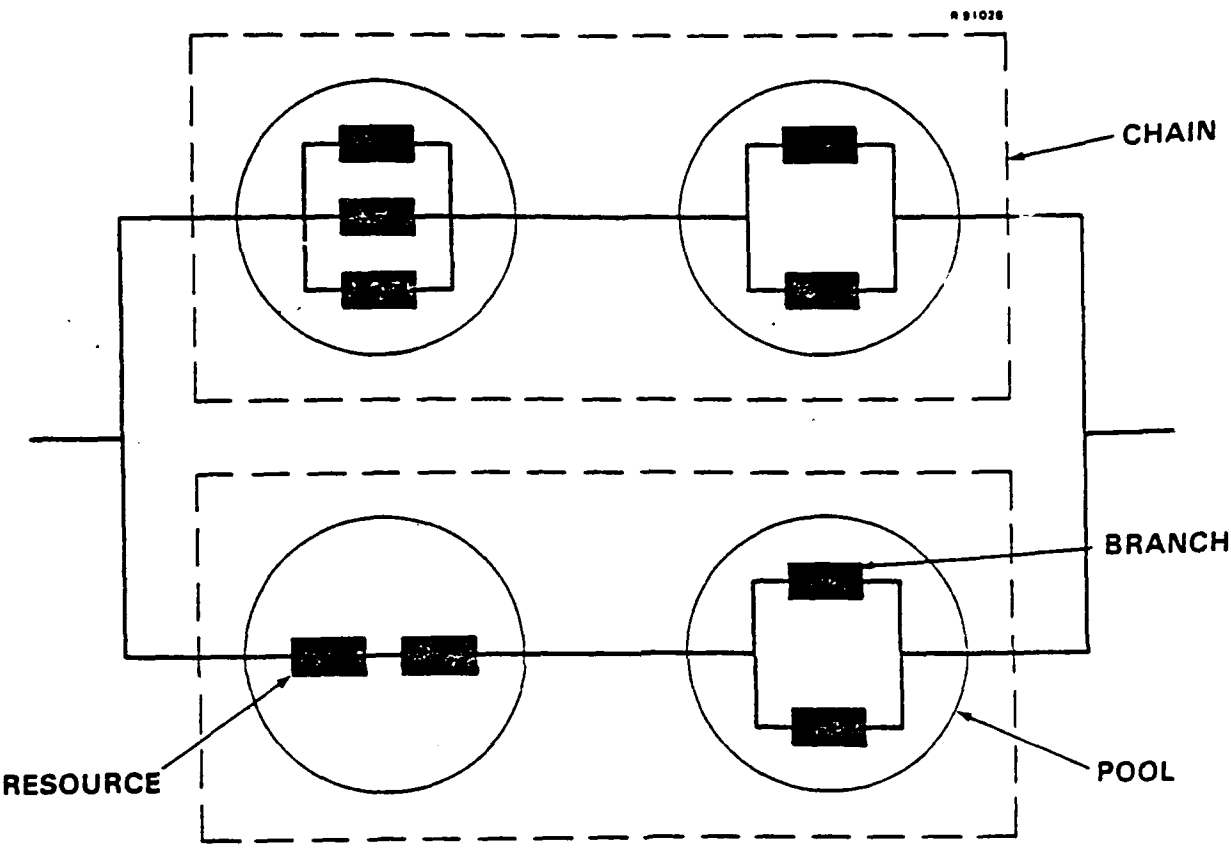


FIGURE 2-5

MIREM Terminology

MIREM Input/Output

The MIREM input is accomplished by using the data entry program named "DATAIN". As stated in the introduction, the data entry program creates two files, an architectural and a scenario file. The files can be created by an experienced MIREM user without the use of DATAIN, however, its use is recommended for an inexperienced user. The inputs to the architectural file must be accomplished before the scenario file because the scenario file utilizes some of the input from the architectural file. The architectural file consists of the following six main topics:

- Functions
- LRM/LRUs
- Resources
- Chains
- Pools
- Architectural File Name

Each of these main topics required the input of many variables associated with them. The "Functions" and "LRM/LRU" only need an inputted name for each group. The "LRM/LRU" is an optional input. The "Resources" topic requires a resource identification number, the quantity of resources, the resource failure rate (per million hours), the distinction of resource or interconnection, the mean-time-to-repair, and the resource name. The topic of "Chains" requires a chain number, a parallel chain number, a chain pair name and the number of functions in the chain. The "Pools" topic requires information about all pools within the system. The information needed for the "Pools" topic is a pool index number, the chain number the

pool is within, a LRM/LRU name (if option is selected), a pool type (contending, noncontending, shared pools, chain failed pools), the number of branches within the pool, the distinction of active or standby redundancy, the undetected failure rate, false alarm rate, and the minimum acceptable level of repair. The final input is the name of the architectural file in which all the above data is to be saved.

The scenario file is a file which retains data on the mission and the basic run parameters. The scenario file consists of the following inputs:

- total operating time
- simultaneous functions (Yes/No)
- Failure Rate Scale Factor
- Scheduled Maintenance
- Repair Sequence (Series/Parallel)
- Various output options
- Mission phase list (Phase number, Length, Phase name, Critical functions)
- name of the file to save all scenario file information.

The MIREM output is created after the data has been stored in "DATAIN" and run in the "MIREM" program. The output is stored in "MIREM OUT". There are eight output options. Any or all of these output options can be selected to print out data. As stated above, the options are selected in the scenario file. The eight output options are titled:

1) Phase-by-Phase Mission-Completion-Success-Probability (MCSP)

2) Mean-Time-Between-Critical-Failure (MTBCF) Report

3) MCSP & Budget Output Option

4) Mean-Time-Between-Functional-Failure (MTBFF) Report

5) LRM/LRU Budget Report

6) Repair Policy Report

7) Testability Factors Report (BIT Option)

8) Testability Factors Report (BIT MTBCF Option)

The contents of the outputs are self-explanatory, but if further information is desired consult the "Mission Reliability Model Users Guide" (reference 23).

MIREM offers the user the following capabilities and advantages:

- Can be used to evaluate a specific architecture under various repair scenarios (i.e. deferred, immediate, scheduled/preventive maintenance, repair at degraded level).

- Computes many reliability figures of merit (i.e. MTBCF under various repair scenarios, Reliability, Availability, MTTR, and Mean-Time-Between-Maintenance-Actions (MTBMA).

- Accounts for imperfect BIT and multifunctional shared resources.

- Low computation time for simple systems.

- Reliability block diagram input.

Unfortunately, MIREM has the following disadvantages and limitations:

- MIREM documentation is confusing to a new user.

- The "DATAIN" data entry program is hard to use and learning it is time consuming.

- "DATAIN" makes no provision for using the "group" feature of MIREM. Groups must be inputted directly into an architectural file. Such a feature is needed for modeling internal levels of redundancy greater than two.

- Limited to ten mission phases and two redundant shared chains.

- Caution should be taken when using MIREM since it hasn't been formally or fully verified. At times, certain features have been found to yield inconsistent results. MIREM verification is currently being conducted at RADC.

2.5 REST

REST (REliability Simulation Tool) is a Monte Carlo simulation program developed by engineers at the Rome Air Development Center. Given a fault tolerant system configuration, component MTBFs, and repair rates, the REST program calculates the system MTBCF, MTTR, reliability and availability. Preliminary verification efforts have shown that the simulation results are accurate to within 2% of the actual values.

REST also synthesizes reliability demonstration plans for fault tolerant systems. This is useful because the reliability demonstration plans in MIL-STD-781C are based on exponential failure distributions, so they are invalid for systems with redundancy or fault tolerance since such systems do not follow an exponential failure distribution.

REST can be used to model systems with the following attributes:

1. Full or partial standby redundancy
2. Multifunction operation
3. Complex series parallel configurations
4. Scheduled maintenance

The user describes the system by breaking it into sets. A set can be a single component or it can be a group of N components of which M must be functioning for the set to function. The user inputs MTBFs for each component as well as other information which varies depending on the system attributes chosen.

If attribute 1 (full or partial standby redundancy) was chosen by the user, a failure rate ratio must be input for every set with two or more components. This ratio is a number between 0 and 1 which defines the failure rate of a component when it is in standby. For example, if a component has an active failure rate of .001 failures per hour and a standby failure rate ratio of .1, the standby failure rate will be .0001 failures per hour. A standby failure rate ratio of 0 indicates that there can be no failures of components in standby. If a set employs fully active redundancy, the standby failure rate ratio will be equal to 1.

If the system to be analyzed performs multiple functions, or if the user wishes to study degraded modes of operation, the multifunction operation option can be invoked. For each set the user would define N, the number of components in the set and also M_1, $M_2 \ldots M_n$, where:

M_i = Minimum number of components needed for the set to perform function i

$$0 \leq M_i \leq N$$

n = Number of functions

REST will output MTBCF, reliability, and availability for each of the system functions. REST automatically assumes the system is a straight series combination of the defined sets unless attribute 3 - Complex series parallel configuration is invoked. In this case, the user must specify each of the possible paths of sets which allow for system operation.

For example:

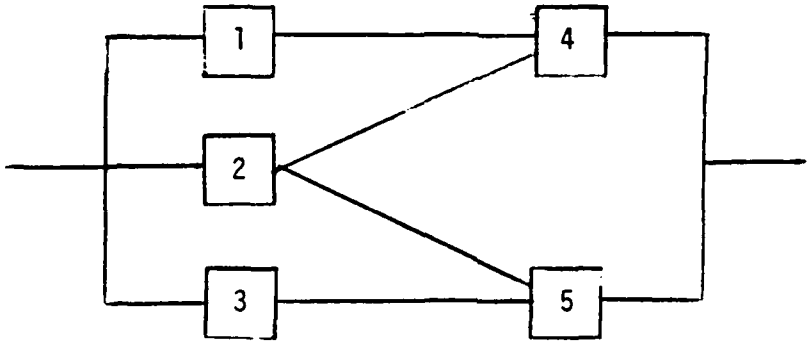


FIGURE 2-6

Sample System To Illustrate REST

This system contains five sets which may be individual components or redundant groups of components. There are four functional paths for this system. They are:

Path 1: Sets 1 and 4

Path 2: Sets 2 and 4

Path 3: Sets 2 and 5

Path 4: Sets 3 and 5

If functional paths are to be defined for a multifunction system, a separate group of paths must be defined for each function.

Attribute 4: Scheduled Maintenance is invoked when the user wants to simulate system repair at set intervals. This simulates the reliability of a system which is periodically inspected and repaired.

The user of REST has the option to calculate system mean-time-to-repair and availability. To do this, MTTRs must be input for each component. REST then simulates repair after every system critical failure. Availability is calculated by dividing system uptime by the total time. System MTTR is calculated by dividing the downtime by the number of repair actions performed.

Another useful maintenance scenario possible for simulation is the case of concurrent repair. This is when repair occurs whenever a component fails and the system continues to function during the repair action. This enhancement to the REST program is currently under development and should be completed in the near future.

If the user wishes to formulate a reliability demonstration plan, REST gives the following options:

1. Sequential tests where the user inputs the slope, intercepts maximum number of failures and discrimination ratio.

2. Sequential tests where the user specifies which MIL-STD-781 test is to be used. (Tests Ic-VIIIc)

3. A fixed length test where the user inputs the maximum test time, the maximum number of failures and the discrimination ratio.

4. A fixed length test where the user specifies which MIL-STD-781 test is to be used. (Tests IXc - XVIIc)

REST will then simulate the reliability demonstration and calculate the producer's and consumer's risks. For sequential tests, REST also calculates the average times to reach an accept decision, a reject decision, and the overall time to reach a decision. This allows the user to try several demonstration plans to find the optimum solution, knowing what the allowable risks and time constraints are.

REST offers the following capabilities and advantages:

- Handles full or partial standby redundancy.
- Handles complex configurations.
- Being a Monte Carlo simulation, the program offers a high degree of flexibility.
- Can be used to simulate repairable systems.
- Models multiple system functions.

However, the use of REST has the following limitations and disadvantages:

- Since REST is a Monte Carlo simulation, a small degree of error results in the outputs.
- The program hasn't been completely verified.
- At this point in time, the user's manual hasn't been completed.

Efforts to verify, document, and add enhancements to the REST program are currently ongoing at RADC.

2.6 R&MA^{2}T^2

The Reliability, Maintainability and Availability Analysis Trade-off Tool (R&MA^{2}T^2) is a fault tolerant system analysis software package that was developed by the Rome Air Development Center. The program is designed to calculate the reliability parameters of repairable systems with series-parallel configurations. To run the program, the user adds a set of Fortran IV instructions which describe the system architecture, repair rates, and failure rates. A knowledge of Fortran is helpful in using the program, but not necessary. The instruction set is simple and allows the user to describe any type of series-parallel system with ease. After adding the instructions, the program is run as a whole and provides the user with the system availability, steady state mean-time-between-failure, mean repair rate, and mean time to first failure.

The solution technique used by R&MA^{2}T^2 is based on a series of algorithms developed in RADC-TR-77-287, "A Redundancy Notebook", by Jerome Klion (Reference 13). The algorithms of the program are an alternative to the classical Markovian approach to the solution of redundant/fault tolerant system reliability (which is also used for both repairable and non-repairable systems). Instead, a set of general expressions is used to describe the availability of series-parallel configurations. These expressions are developed from probability equations based on the assumption that all units have exponentially distributed times to failure. For example, in the case of a two unit active redundant parallel system, the steady state availability is calculated using the following equation:

$$Asys = \frac{\mu^2+2\lambda\mu}{(\lambda+\mu)^2}$$

Where:

μ = repair rate

λ = failure rate.

This equation is the limiting form of the probability that the system is operational at any time t. This type of analysis is applied for similar blocks which represent redundant subsystems and is then combined to determine the overall system parameters. Because the expressions are not time dependent and don't involve lengthy matrix calculations peculiar to Markovian analysis, large systems can be solved quickly with relatively small amounts of computer resources.

Inputs to R&MA^{2}T^2 include system architecture, failure rates, and repair rates. The Fortran instruction set basically defines each of these parameters for each component or module in the system. Also included in the instructions are calls to subroutines which calculate the reliability parameters of each parallel or series block in the system. For instance, to calculate the failure rate of a parallel 2 out of 3 similar redundant block (block "B"), the user would include the following instruction:

BF = CPARL (3,2).

The user would then use "BF" as the failure rate for block "B" at the end of the instruction set when determining the system failure rate. No actual calculations are required in the instruction set. Instead, the purpose of the instructions is to define the variables in the system, and to define the block and system parameters based on their architectures. The last

instructions in the set describe the system in terms of the subsystem blocks that have been built previously.

R&MA^{2}T^2 outputs results for each subsystem block as well as overall system results. Outputs are in the order of block input, and finish with the system level evaluation. Included in the results are block steady state failure rate, block steady state mean-time-between-failure, block availability, block mean repair rate, and block mean time to first failure. For those blocks that contain redundant elements, the program computes state failure rates and state availabilities, where each state represents a certain number of operational elements.

R&MA^{2}T^2 is included in the Optimum Reliability and Component Life Estimator (ORACLE), a reliability prediction program developed by the Rome Air Development Center. ORACLE is a computerized version of MIL-HDBK-217 which calculates component level failure rates, and then calculates the system failure rate for each module by a series analysis. These modules can then be combined into a series-parallel system using R&MA^{2}T^2. A query routine is provided which allows the user to build the system configuration without having to insert the actual Fortran code. Unfortunately, in order to use the query routine, the user must start at the part level to build the system.

R&MA^{2}T^2 offers the user the following capabilities and advantages:

- Ability to model repairable systems.

- Ability to model large systems with a large number of parallel components.
- Easy to change parameters for tradeoff analysis.
  Useful for steady state analysis, availability
- Can be used in conjunction with ORACLE for analysis from part to system level.
- Solution time is very small.

However, the use of R&MA^{2}T^2 has the following limitations and disadvantages:

- Limited in scope to repairable systems with exponentially distributed times to failure and time to repair.
- Instruction set can be tedious for large systems.
- Cannot model complex paths, standby redundancy, or preventive maintenance scenarios.
- No provision for imperfect fault coverage.

3.0 EXAMPLE MODELING APPLICATIONS

This section provides some sample fault tolerant system reliability analyses that were conducted using the RADC Fault Tolerant System Reliability Evaluation Facility. The purpose here is to further illustrate the capabilities of the facility.

The following system analyses are found within:

1. CARE III is used to model a rather complex digital flight control system. The system is hypothetical and was created to illustrate many of the CARE III features.

2. HARP is used to compute the reliability of a remote radar subsystem with imperfect fault handling and coverage.

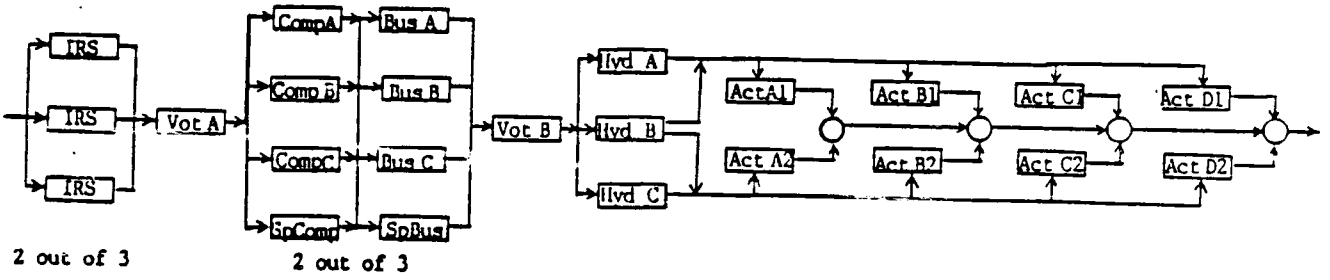
3. REST is used here to demonstrate the effects of two different maintenance philosophies for the same system.

4. R&MA^{2}T^2 is used to determine the steady state availability, MTBF, MTTR and Mean-Time-To-First-Failure (MTTFF) of a Local Area Network (LAN) that is part of a world wide command and control system.

3.1 MODELING A DIGITAL FLIGHT CONTROL SYSTEM USING CARE III

System Description

Figure 3-1 depicts the configuration of the example system. The first stage consists of three inertial reference sensors arranged in a 2 out of 3 voting scheme, followed by a voting circuit which relays the sensor signals to the digital computer stage.



Legend

IRS - Inertial Reference Sensor

Vot A - Voter A

Comp - Computer

Sp - Spare

Hyd - Hydraulic Subsystem

Act - Actuator

Figure 3-1

Digital Flight Control System Reliability Block Diagram

The digital computers and their associated buses are also arranged in a 2 out of 3 voting configuration. There is one spare computer and one spare bus which can be brought on line in the event of a fault in one of the original units. A second voting circuit follows the bus stage, to relay signals to the control surface actuators.

The mechanical portion of the system consists of four pairs of control surface actuators and three hydraulics subsystems. Each hydraulics subsystem consists of a hydraulic pump and two independent sets of hydraulic lines. The system is configured so that each set of four actuators is powered by one of these hydraulics subsystems. The third hydraulic subsystem is a redundant set which can power either or both actuator sets.

Fault Handling Models

All stages of the system will exhibit perfect fault handling characteristics with the exception of the computer and bus stages. Each of these will exhibit permanent, transient and intermittent faults. The parameters of the fault handling models are listed in Table 3-1.

Table 3-1

Fault Handling Model Parameters For Digital Flight Control System

FAULT TYPE	COMPUTER PERMANENT	BUS PERMANENT	TRANSIENT (BUS OR COMP)	INTERMITTENT (BUS OR COMP)
α	0.0	0.0	36000	2100
β	0.0	0.0	0.0	3000
δ	360	10000	360	360
ρ	180	0.0	180	180
ϵ	3600	0.0	3600	3600
P_a	1.0	1.0	1.0	1.0
P_b	0.0	0.0	0.0	0.0
C	.999	1.0	1.0	1.0

Footnote: Parts of this example are taken from examples in:

NASA Tech Memo 4011 "Tutorial and Hands-On Demonstration of a Fluent Interpreter for CARE III", by A.L. Martensen and S.J. Bavuso, Nov 87.

Fault Occurrence Models

The hydraulic subsystems and control surface actuators are modeled using Weibull failure distributions. This will simulate wearout, which is common in mechanical components. All other stages in the system are electronic, therefore, they are modeled using exponential failure distributions. The parameters of the fault occurrence models are listed in table 3-2.

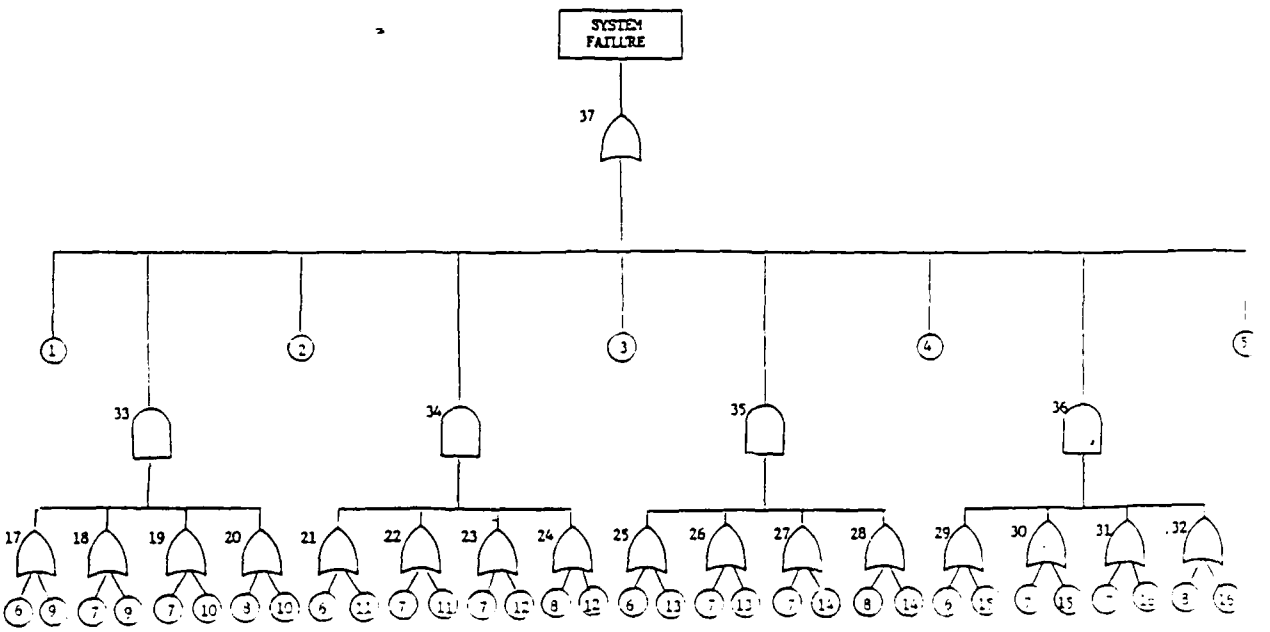
Table 3-2

Fault Occurrence Model Parameters For Digital Flight Control System

STAGE	FAULT HANDLING MODEL	λ (Failures/Hr)	W
Inert. Ref. Sensor	(None)	1.5×10^{-5}	1.0
Voter A	(None)	2.0×10^{-8}	1.0
Dig. Computer	Computer Perm.	2.4×10^{-4}	1.0
	Transient	3.6×10^{-3}	1.0
	Intermittent	1.6×10^{-4}	1.0
Computer Bus	Bus Perm.	2.7×10^{-6}	1.0
	Transient	6.2×10^{-4}	1.0
	Intermittent	3.7×10^{-6}	1.0
Voter B	(None)	2.0×10^{-8}	1.0
Hydraulics Mod.	(None)	1.0×10^{-4}	1.3
Hydraulics Submod.	(None)	2.0×10^{-7}	1.2
Actuators	(None)	3.7×10^{-5}	1.1

System Fault Tree

Figure 3-2 shows the system fault tree, which is used to describe all the possible combinations of stage failures which can lead to system failure. If any of the events leading into the OR gate at the top (#37) occur, the system will fail. Input events 1 thru 5 are the failure of the inertial reference sensor stage, the computer stage, the bus stage, and the two voting circuits. Each of the AND gates (#33-#36), and the events leading into them, depict the possible failure combinations which will cause the loss of both actuators of a pair (and therefore, system failure). The loss of all three hydraulics sets will also trigger the AND gates. It should be noted that fault tree notation is the only way to model this configuration. It would be impossible to model this system using software which only models series, parallel, and m out of n configurations.



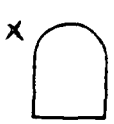
Legend

i Input Event, failure of stage i

Stages

1. Inertial Reference Sensors
2. Voting Circuit A
3. Digital Computers
4. Computer Buses
5. Voting Circuit B
6. Hydraulics Set A
7. Hydraulics Set B
8. Hydraulics Set C
9. Control Actuator A1
10. Control Actuator A2
11. Control Actuator B1
12. Control Actuator B2
13. Control Actuator C1
14. Control Actuator C2
15. Control Actuator D1
16. Control Actuator D2

- "or" gate - occurrence of any input event causes output event "x"



- "and" gate - occurrence of all input events causes output event x.

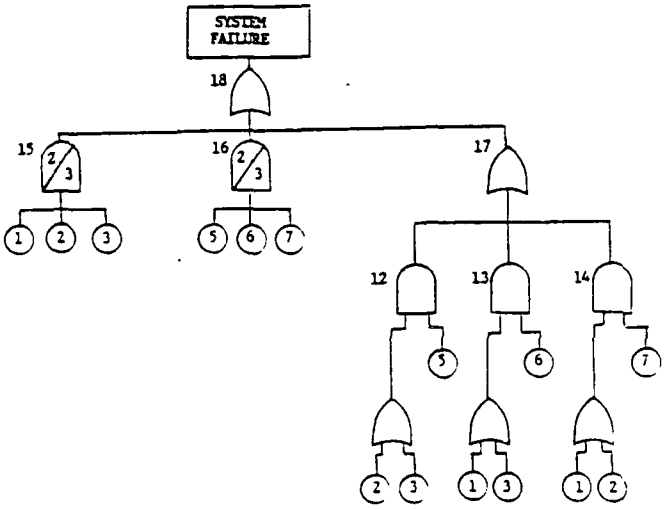
Figure 3-2

System Fault Tree For Digital Flight Control System

Critical Pairs Fault Tree

The computer and bus stages of this system are susceptible to critically paired failures. If two of the three on-line computers experience faults before the spare computer can be brought on line, the voting circuit will not reach a majority vote and the system will fail. The same is true if two of the three on-line buses fail before configuration occurs.

Another type of critically coupled failures are those which occur between stages. If a fault occurs in a bus and in one of the two on-line computers not connected to that bus before reconfiguration occurs, there will be two bad signals entering the voting circuit, which will cause system failure. The fault tree representation of these critical pairs is shown in Figure 3-3. This is needed any time there are critical pairs present.



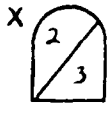
Legend

i - Input Event - failure of module i

1. Computer A
2. Computer B
3. Computer C
4. Spare Computer
5. Bus A
6. Bus B
7. Bus C
8. Spare Bus

"or" gate - occurrence of any input event causes the output event x.

"and" gate - occurrence of all input events causes the output event x.



"2 out of 3" gate - occurrence of any 2 input events cause the output event x.

Figure 3-3

Fault Tree Representation For Critical Pairs

Input/Output

The following was input into the CARE III program:

Failures rates as shown in Table 3-1

mission time = 20 hrs

Fault error handling models and fault tree descriptions

Figure 3-4 Illustrates output of the CARE III program.

SUMMARY INFORMATION:

	TIME (HOURS)	Q SUM	Pe SUM	Q SUM + Pe SUM
1	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
2	9.80392E-03	2.09330E-09	3.92249E-10	2.48555E-09
3	1.96078E-02	5.41035E-09	7.84700E-10	6.19505E-09
4	2.94118E-02	8.62814E-09	1.17737E-09	9.80551E-09
5	3.92157E-02	1.18694E-08	1.57026E-09	1.34396E-08
6	4.90196E-02	1.51107E-08	1.96738E-09	1.70741E-08
7	5.88235E-02	1.83521E-08	2.35674E-09	2.07088E-08
8	6.86275E-02	2.15935E-08	2.75034E-09	2.43438E-08
9	7.84314E-02	2.48348E-08	3.14420E-09	2.79790E-08
10	8.80392E-02	3.13174E-08	3.93267E-09	3.52501E-08
11	1.17647E-01	3.78000E-08	4.72219E-09	4.25222E-08
12	1.37255E-01	4.42825E-08	5.51278E-09	4.97953E-08
13	1.56863E-01	5.07650E-08	6.30449E-09	5.70695E-08
14	1.76471E-01	5.72474E-08	7.09734E-09	6.43447E-08
15	1.96078E-01	6.37298E-08	7.89135E-09	7.16211E-08
16	2.15686E-01	7.02121E-08	8.68654E-09	7.88986E-08
17	2.35294E-01	7.66943E-08	9.48296E-09	8.61773E-08
18	2.74510E-01	8.96586E-08	1.10795E-08	1.00738E-07
19	3.13726E-01	1.02623E-07	1.26813E-08	1.15304E-07
20	3.52941E-01	1.15587E-07	1.42883E-08	1.29875E-07
21	3.92157E-01	1.28550E-07	1.59009E-08	1.44451E-07
22	4.31373E-01	1.41514E-07	1.75192E-08	1.59033E-07
23	4.70588E-01	1.54477E-07	1.91433E-08	1.73620E-07
24	5.09804E-01	1.67440E-07	2.07734E-08	1.88214E-07
25	5.49020E-01	1.80403E-07	2.24097E-08	2.02813E-07
26	6.27451E-01	2.06328E-07	2.57013E-08	2.32030E-07
27	7.05882E-01	2.32252E-07	2.90196E-08	2.61272E-07
28	7.84314E-01	2.58176E-07	3.23656E-08	2.90541E-07
29	8.62745E-01	2.84098E-07	3.57407E-08	3.19839E-07
30	9.41176E-01	3.10020E-07	3.91459E-08	3.49166E-07
31	1.01961E+00	3.35941E-07	4.25823E-08	3.78523E-07
32	1.09804E+00	3.61861E-07	4.60513E-08	4.07912E-07
33	1.17647E+00	3.87780E-07	4.95538E-08	4.37334E-07
34	1.33333E+00	4.39619E-07	5.66642E-08	4.96280E-07
35	1.49020E+00	4.91448E-07	6.39223E-08	5.55370E-07
36	1.64706E+00	5.43276E-07	7.13372E-08	6.14614E-07
37	1.80392E+00	5.95102E-07	7.89173E-08	6.74019E-07
38	1.96078E+00	6.46924E-07	8.66713E-08	7.33595E-07
39	2.11765E+00	6.98742E-07	9.46079E-08	7.93350E-07
40	2.27451E+00	7.50557E-07	1.02736E-07	8.53293E-07
41	2.43137E+00	8.02369E-07	1.11062E-07	9.13431E-07
42	2.74510E+00	9.05982E-07	1.28349E-07	1.03433E-06
43	3.05882E+00	1.00958E-06	1.46535E-07	1.15612E-06
44	3.37255E+00	1.11317E-06	1.65683E-07	1.27885E-06
45	3.68627E+00	1.21674E-06	1.85860E-07	1.40260E-06
46	4.00000E+00	1.32030E-06	2.07138E-07	1.52743E-06
47	4.31373E+00	1.42384E-06	2.29572E-07	1.65341E-06
48	4.62745E+00	1.52737E-06	2.53232E-07	1.78060E-06
49	4.94118E+00	1.63089E-06	2.78183E-07	1.90907E-06
50	5.56863E+00	1.83788E-06	3.32210E-07	2.17009E-06
51	6.19608E+00	2.04482E-06	3.92133E-07	2.43697E-06
52	6.82353E+00	2.25170E-06	4.58547E-07	2.71024E-06
53	7.45098E+00	2.45852E-06	5.31865E-07	2.99039E-06
54	8.07843E+00	2.66529E-06	6.12626E-07	3.27792E-06
55	8.70588E+00	2.87201E-06	7.01315E-07	3.57332E-06
56	9.33333E+00	3.07867E-06	7.98437E-07	3.87710E-06
57	9.96078E+00	3.28527E-06	9.04490E-07	4.18976E-06
58	1.12157E+01	3.69831E-06	1.14536E-06	4.84367E-06
59	1.24706E+01	4.11112E-06	1.42780E-06	5.53893E-06
60	1.37255E+01	4.52372E-06	1.73573E-06	6.27944E-06
61	1.49804E+01	4.93608E-06	2.13299E-06	7.06907E-06
62	1.62353E+01	5.34822E-06	2.54344E-06	7.91166E-06
63	1.74902E+01	5.76013E-06	3.05082E-06	8.81096E-06
64	1.87451E+01	6.17182E-06	3.59901E-06	9.77083E-06
65	2.00000E+01	6.58328E-06	4.21180E-06	1.07951E-05

K STAGE FAILURES	AFTER EXACTLY K STAGES HAVE FAILED BY 2.00000E+01 HOURS, PORTION OF THE UNRELIABILITY CAUSED BY: FAULT HANDLING	EXHAUSTION OF MODULES
0	6.58328E-06	0.00000E+00
1	X	3.67960E-06
2	X	5.30283E-07
3	X	1.92481E-09
4	X	2.64414E-12
5	X	6.01879E-18

TOTAL SYSTEM UNRELIABILITY AT 2.00000E+01 HOURS = 1.07951E-05

Figure 3-4

CARE III Output For Digital Flight Control System Analysis

Figure 3-4 illustrates Q sum - the unreliability due to improper fault handling, and P sum - the unreliability due to exhaustion of models. Q sum + P sum equals the total unreliability. Outputs were generated at 65 steps ranging from t=0 to t=20 hours. Also provided are unreliabilities for each stage of the system. The total system unreliability is printed at the bottom.

3.2 MODELING A FAULT TOLERANT RADAR SYSTEM USING HARP

A simplified radar remote subsystem has been chosen to demonstrate the use of HARP in the determination of fault tolerant system reliability. The system is a simple version of what might be found at an antenna site of a radar system. It is assumed that no repair occurs on the system for this demonstration. Repair could be included by modeling the system with a Markov chain, but this would involve a much larger model than observed here. Because a nonrepairable system is the one under study, a fault tree can be used to describe the system architecture. When converted to a Markov chain by HARP, this system consists of 40 independent Markov states. Figures 3-5 and 3-6 show the reliability block diagram and resulting fault tree for the system. As can be seen, the redundant power supplies are an example of dissimilar redundancy. This is easily handled by the fault tree Fault/Occurrence Repair Model (FORM) utilized by HARP.

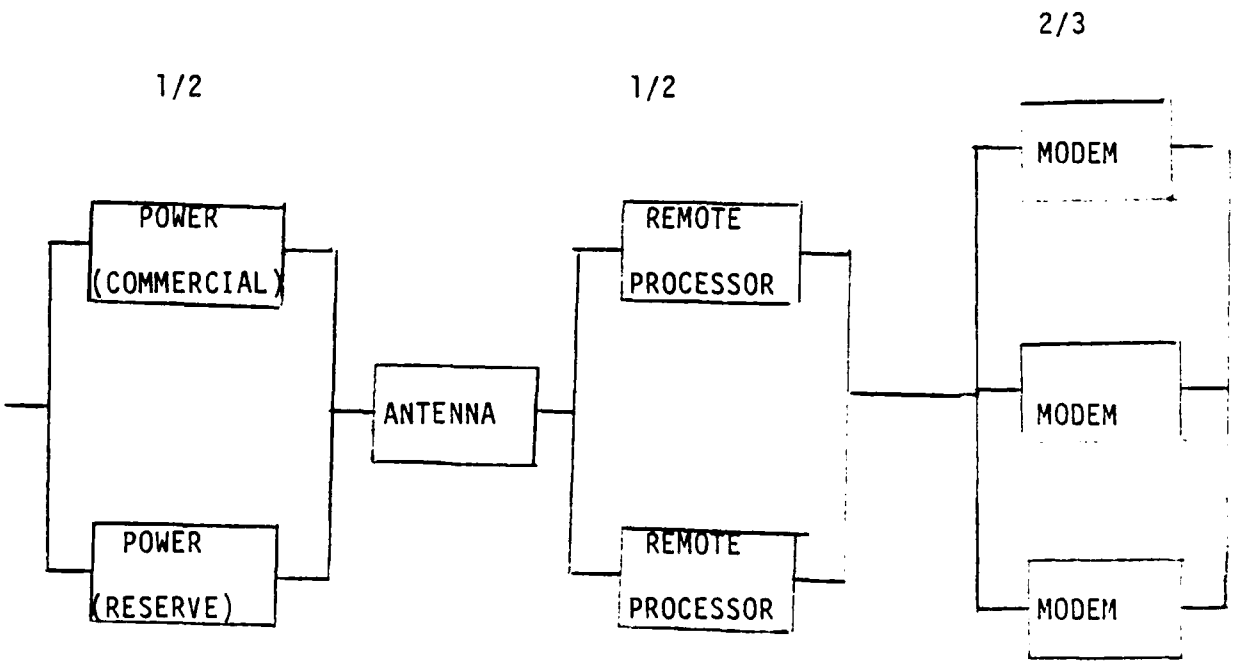


Figure 3-5

Radar System Reliability Block Diagram (RBD)

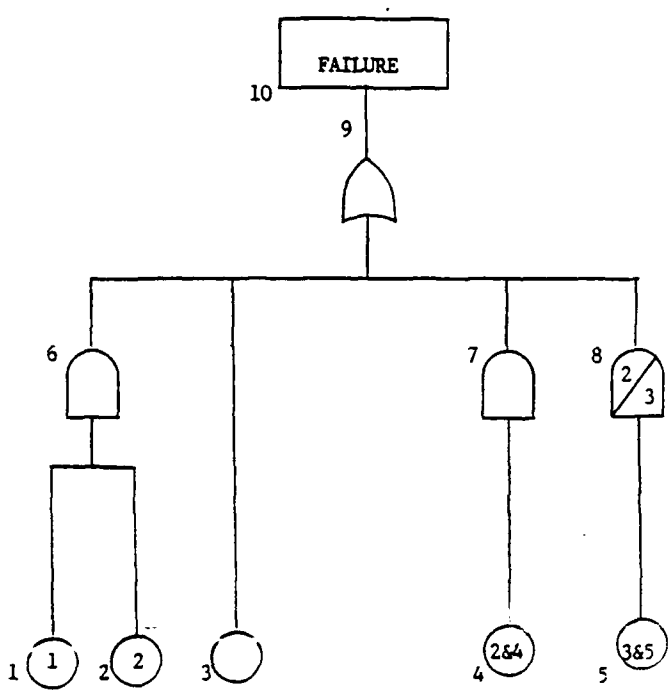


Figure 3-6

Fault Tree Corresponding To Figure 3-5 RBD

The fault tree was created to describe the series-parallel configuration of the system. All components are basic events denoted by circles with the component type listed inside. In this diagram type 1 corresponds to the commercial power supply, type 2 to the reserve power supply, type 3 to the antenna, type 4 to the remote processor and type 5 to the modems. Multiple parallel units are described with one circle (for similar redundancy) enclosing the replication factor times the component type. "And" gates are used for parallel units indicating that all components must fail before the parallel block fails. "Or" gates are used for series units indicating that any failure will cause failure of the series block. Each

node is numbered for use in identification during data input and interpretation of results. For this system, the parallel combinations of power supplies and remote processors feed into "and" gates, while the modems feed into a "2/3" gate, indicating that 2 modems must be operational for operation of the parallel block. In turn, the series combination feeds into a "or" gate which ultimately leads to system failures.

Each component type has a specific failure rate as well as the definition of its fault handling properties. HARP offers the user a wide variety of Fault/Error Handling Models (FEHM's) in order to approximate the fault coverage or switching characteristics of a system. This enables the user to choose a FEHM based on his knowledge of the system. Each component type has a unique FEHM to account for its specific fault handling abilities. Tables 3-3, 3-4 and 3-5 list the FEHM and associated parameters for the power supplies, processors and modems respectively. The tables are listings of the textual files created by HARP that define the FEHM for each component type. Files are created allowing the user to easily change parameters for tradeoff analysis. These FEHM's were selected for demonstration purposes, but in the case of a real system, the FEHM's would be selected based on data collected for each component in the system.

```
DISTRIBUTIONS AND PROBABILITIES

TRANSIENT RESTORATION EXIT:

  EXIT PROBABILITY:   0.99000000D+00
 DISTRIBUTION TYPE:  CONSTANT
  VALUE:  0.10000000D+01

RECONFIGURATION COVERAGE EXIT:

  EXIT PROBABILITY:   0.85000000D-02
 DISTRIBUTION TYPE:  EXP
  RATE:   0.50000000D+00

SINGLE POINT FAILURE EXIT:

  EXIT PROBABILITY:   0.15000000D-02
 DISTRIBUTION TYPE:  UNIF
  LOWER LIMIT:  0.00000000D+00
  UPPER LIMIT:  0.10000000D+01
```

Table 3-3

Power Supply FEHM

```
PROBABILITIES AND MOMENTS

TRANSIENT RESTORATION EXIT:

  EXIT PROBABILITY:  .9800
  FIRST MOMENT OF TIME TO EXIT:  .3000
  SECOND MOMENT OF TIME TO EXIT:  .5000
  THIRD MOMENT OF TIME TO EXIT:  .6000

RECONFIGURATION COVERAGE EXIT:

  EXIT PROBABILITY:  .1000E-01
  FIRST MOMENT OF TIME TO EXIT:  .4000
  SECOND MOMENT OF TIME TO EXIT:  .6000
  THIRD MOMENT OF TIME TO EXIT:  .7000

SINGLE POINT FAILURE EXIT:

  EXIT PROBABILITY:  .1000E-01
  FIRST MOMENT OF TIME TO EXIT:  .1000
  SECOND MOMENT OF TIME TO EXIT:  .2000
  THIRD MOMENT OF TIME TO EXIT:  .3000
```

Table 3-4

Remote Processor FEHM

```
ARIES. TRANSIENT. RECOVERY. MODEL

PROBABILITY THAT FAULT IS TRANSIENT 0. 80000000D+00

MEAN DURATION OF TRANSIENT FAULT 0. 10000000D-01

PROBABILITY THAT FAULT IS CATASTROPHIC 0. 10000000D-03

NUMBER OF TRANSIENT RECOVERY PHASES        3
  PHASE 1 DURATION:   0. 20000000D-02 EFFECTIVENESS:   0. 50000000D+00
  PHASE 2 DURATION:   0. 50000000D-02 EFFECTIVENESS:   0. 60000000D+00
  PHASE 3 DURATION:   0. 30000000D-01 EFFECTIVENESS:   0. 70000000D+00

COVERAGE OF PERMANENT FAULT:   0. 95000000D+00
```

Table 3-5

Modem FEHM

For the power supply, the Distribution and Probabilities FEHM was selected. The probability of a transient fault is .99, while the probability of successful reconfiguration given a permanent fault is .15. No FEHM was used for the antenna since this is a nonredundant unit. The remote processor was modelled with the Probabilities and Moments FEHM, which gives moments in time and exit probabilities for transient restoration, successful reconfiguration, and single point failure. The modems' coverage is represented by an Automated Reliability Interactive Estimation System (ARIES) transient fault recovery model. This model goes through a user defined number of transient recovery phases with varying probabilities of success. If all phases are completed without success, then a permanent recovery process is initiated with a user defined probability of success.

The HARP program was run on this system for a period of 240 hours, with a phase by phase analysis at 60 hour intervals. The failure rate of the commercial power supply is $.2 \times 10^{-4}$ failures/hr, for the reserve power supply $.35 \times 10^{-2}$ failures/hr, for the antenna $.1 \times 10^{-4}$ failures/hr for the remote processors $.24 \times 10^{-2}$ failures/hr and for the modem $.36 \times 10^{-2}$ failures/hr. Because the system does not include repair, the reliability declines with time which is easily observed from the phase-by-phase results which are shown in Table 3-6. These results are also placed in a file as x and y coordinates which can be used to plot reliability as a function of time. The state probabilities refer to the system failures according to component type, single point failure, and near coincident failure. For instance, "F3" is the probability of system failure due to exhaustion of component type 3, the antenna.

Included in Table 3-7 are the results of a HARP run on the same system using no fault coverage. Comparison reveals a significant different in system reliability. The reliability of the system with no fault coverage is lower than the system that includes FEHM's, due to the fact that the FEHM's allow for the possibility of transient faults which can be corrected prior to component failure.

```
Time:     0.600D+02
State Probabilities and Global Errors:
State name: F3                    0.57916732D-03     0.39D-17
State name: F2                    0.95041372D-06     0.15D-13
State name: F1                    0.10170925D-05     0.10D-13
State name: F4                    0.18805648D-03     0.26D-16
State name: F5                    0.65685076D-01     0.36D-14
State name: FSPF                  0.20407116D-01     0.18D-16
State name: FNCF                  0.91581048D-07     0.30D-21
-------------------------------------------------------

  Reliability =         0.91313853D+00
Unreliability =         0.86861475D-01

Total failure by redundancy exhaustion =          0.66454267D-01
Total single point failure probability =          0.20407116D-01
Near-coincident failure probability    =          0.91581048D-07

Maximum global error =  0.47D-14

Time:     0.120D+03
State Probabilities and Global Errors:
State name: F3                    0.10836240D-02     0.76D-16
State name: F2                    0.32338093D-05     0.16D-17
State name: F1                    0.36861364D-05     0.16D-17
State name: F4                    0.65271360D-03     0.30D-15
State name: F5                    0.20013548D+00     0.73D-13
State name: FSPF                  0.34366227D-01     0.47D-15
State name: FNCF                  0.16326067D-06     0.56D-20
-------------------------------------------------------

  Reliability =         0.76375487D+00
Unreliability =         0.23624513D+00

Total failure by redundancy exhaustion =          0.20187874D+00
Total single point failure probability =          0.34366227D-01
Near-coincident failure probability    =          0.16326067D-05

Maximum global error =  0.95D-13
```

Table 3-6

Results of HARP Analysis on the Remote Radar System

with Imperfect Fault Coverage

```
Time:     0.180D+03
State Probabilities and Global Errors:
State name: F3                    0.14942358D-02     0.31D-15
State name: F2                    0.60551409D-05     0.26D-17
State name: F1                    0.73109633D-05     0.47D-17
State name: F4                    0.12440793D-02     0.68D-15
State name: F5                    0.34690604D+00     0.31D-12
State name: FSPF                  0.43884838D-01     0.25D-14
State name: FNCF                  0.21767800D-06     0.21D-19
-------------------------------------------------------------
  Reliability =          0.60645719D+00
Unreliability =          0.39354282D+00

Total failure by redundancy exhaustion =          0.34965777D+00
Total single point failure probability =          0.43884838D-01
Near-coincident failure probability    =          0.21767800D-06

Maximum global error =  0.40D-12

Time:     0.240D+03
State Probabilities and Global Errors:
State name: F3                    0.18146931D-02     0.64D-15
State name: F2                    0.88648454D-05     0.16D-17
State name: F1                    0.11270696D-04     0.56D-17
State name: F4                    0.18499699D-02     0.17D-15
State name: F5                    0.48069980D-00     0.67D-12
State name: FSPF                  0.50357461D-01     0.70D-14
State name: FNCF                  0.25808907D-06     0.40D-19
-------------------------------------------------------------
  Reliability =          0.46525769D+00
Unreliability =          0.53474231D+00

Total failure by redundancy exhaustion =          0.48438459D+00
Total single point failure probability =          0.50357461D-01
Near-coincident failure probability    =          0.25808907D-06
```

Table 3-6 Continued

```
Time:     0.600D+02
State Probabilities and Global Errors:
State name: F3                    0.57475377D-03     0.49D-17
State name: F2                    0.10302731D-03     0.27D-16
State name: F1                    0.11035351D-03     0.23D-16
State name: F4                    0.17052045D-01     0.37D-14
State name: F5                    0.97658968D-01     0.35D-14
-------------------------------------------------

  Reliability =         0.88450085D+00
Unreliability =         0.11549915D+00

Total failure by redundancy exhaustion =          0.11549915D+00
Total single point failure probability =          0.00000000D+00

Maximum global error =  0.17D-12

Time:     0.120D+03
State Probabilities and Global Errors:
State name: F3                    0.10426632D-02     0.90D-16
State name: F2                    0.31368745D-03     0.12D-15
State name: F1                    0.35833827D-03     0.16D-15
State name: F4                    0.52905706D-01     0.14D-13
State name: F5                    0.27447627D+00     0.86D-13
-------------------------------------------------

  Reliability =         0.67090333D+00
Unreliability =         0.32909667D+00

Total failure by redundancy exhaustion =          0.32909667D+00
Total single point failure probability =          0.00000000D+00

Maximum global error =  0.11D-11

Time:     0.180D+03
State Probabilities and Global Errors:
State name: F3                    0.13822862D-02     0.15D-15
State name: F2                    0.52227750D-03     0.90D-16
State name: F1                    0.63200917D-03     0.30D-16
State name: F4                    0.90337399D-01     0.10D-13
State name: F5                    0.43992960D+00     0.16D-12
-------------------------------------------------

  Reliability =         0.46719643D+00
Unreliability =         0.53280357D+00

Total failure by redundancy exhaustion =          0.53280357D+00
Total single point failure probability =          0.00000000D+00

Maximum global error =  0.90D-12

Time:     0.240D+03
State Probabilities and Global Errors:
State name: F3                    0.16124295D-02     0.12D-15
State name: F2                    0.68356758D-03     0.21D-15
State name: F1                    0.86909859D-03     0.12D-15
State name  F4                    0.12144121D+00     0.25D-13
State name: F5                    0.56726175D+00     0.14D-12
-------------------------------------------------

  Reliability =         0.30813195D+00
Unreliability =         0.69186805D+00
Total failure by redundancy exhaustion =          0.69186805D+00
Total single point failure probability =          0.00000000D+00
```

Table 3-7

Results of HARP Analysis on the Remote RADAR System

With Perfect Fault Coverage

3.3 MODELING A MULTIFUNCTION SYSTEM UNDER DIFFERENT REPAIR PHILOSOPHIES USING REST

The following example demonstrates how REST can be used to compare the effects of two different maintenance philosophies for the same system. The system in question consists of five identical sets, each consisting of an active component and a redundant component in standby. Figure 3-7 shows the system configuration and Figure 3-8 shows the configuration of each set.

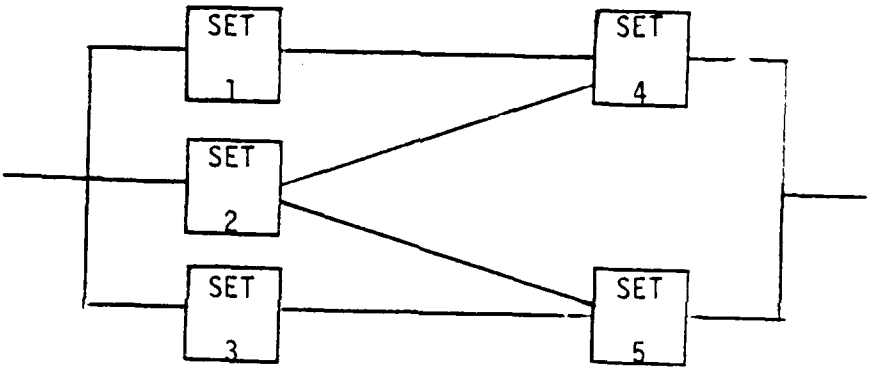
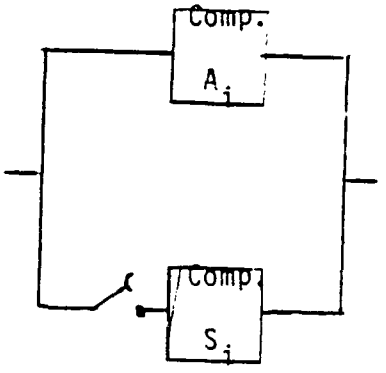


Figure 3-7 - System Configuration



Comp. A_i = Active Comp. of Set_i

Comp. S_i = Standby Comp. of Set_i

Figure 3-8 - Set Configuration

REST is used to compare the system availabilities when the following two maintenance philosophies are employed:

Maintenance Philosophy 1: The system is shut down and all failed components are repaired or replaced whenever a system critical failure occurs.

Maintenance Philosophy 2: The system is shut down and all failed components are repaired or replaced whenever any one of the five redundant sets fails.

This is accomplished by describing the system to REST as a multi-function system. "Function 1" is described by the system's operational configuration as shown in Figure 3-7. "Function 2" is represented by arranging the five redundant sets in a series configuration, as shown in Figure 3-9.

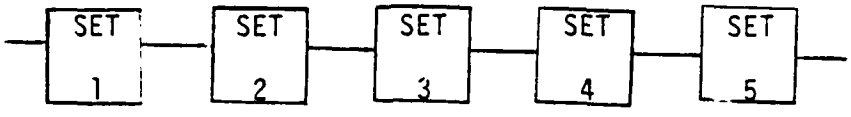


Figure 3-9 - "Function 2" Configuration

A separate simulation is run for each maintenance philosophy. To compute the system availability under maintenance philosophy #1, REST is instructed to simulate repair after every failure of "Function 1" (system failure). To compute the system availability under maintenance philosophy

#2, REST is instructed to simulate repair after every failure of "Function 2" (failure of any set).

This is just one example of the many types of tradeoff analyses which will be possible in the very near future when the development of REST is complete.

3.4 MODELING A LOCAL AREA NETWORK (LAN) SYSTEM USING R&MA^{2}T^2

The system presented here to demonstrate the use of R&MA^{2}T^2 is a Local Area Network (LAN) which is part of a world wide Command and Control System (CCS). The system is repairable and includes an approximate Administrative and Logistics Downtime (ALDT) for each component. This ALDT was added to the Mean-Time-To-Repair (MTTR) to determine the overall repair rate. The functional flow diagram is shown in Figure 3-10 and the system reliability block diagram and other related data in Figure 3-11.

The following acronyms are used in the Figures.

AMPE - Automated Message Processing Equipment

CUP - Common User Processor

IV - Interface Unit

IMP - Interface Message Processor

DDN - Defense Data Network

CC/SM - Control Center Security Monitor

WS IU - Work Station Interface Unit

TG - Transmission Group

CP - Cable Plant

TS - Terminal Server

ECP - External Communications Processor

DC - Dislay Consoles

ICP - Internal Communications Processor

SC - Storage Controller

DASD - Direct Access Storage Disk

MTSS - Magnetic Tape Subsystem

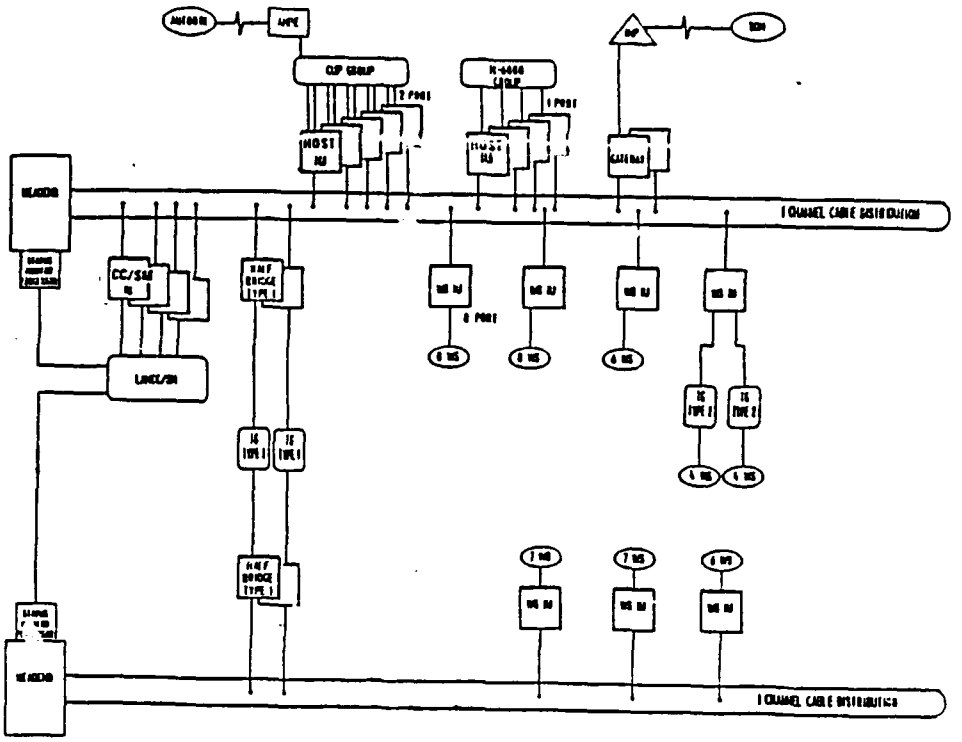


Figure 3-10

LAN Functional Flow Diagram

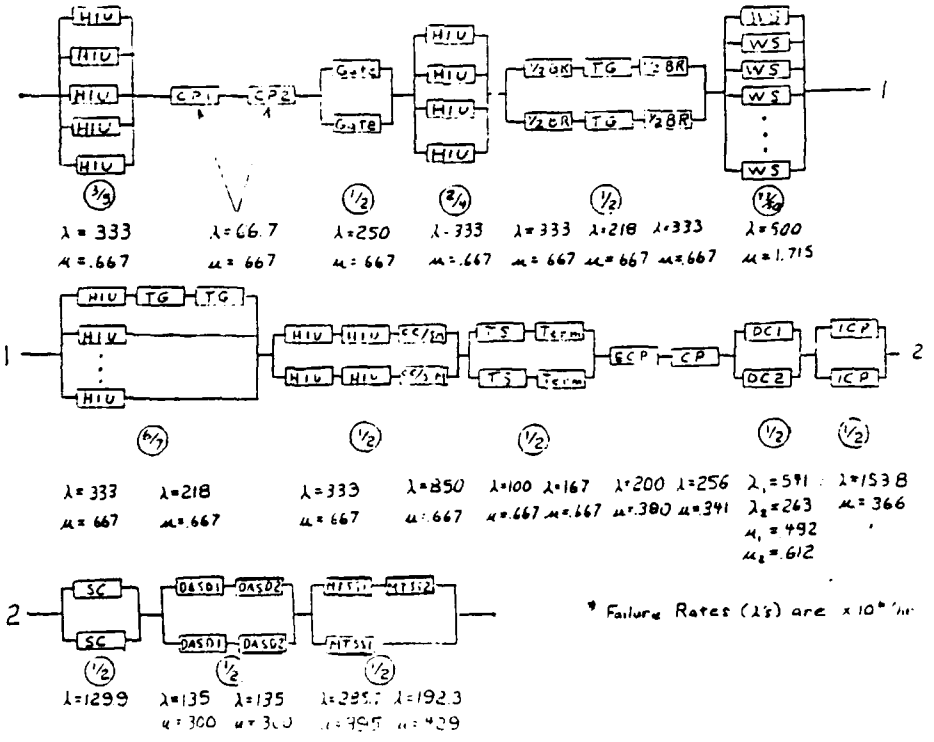


Figure 3-11

LAN Reliability Block Diagram

Each site of the CCS consists of a LAN and a common user processor (CUP) group. The LAN provides both intrasite communications and access to intersite communications. As part of the LAN, there are workstations which are used for local processing as well as accessing information from the CUP. There are 50 workstations of which 43 are necessary at all times. The CUP accesses information from remote sites which are very similar to this system. The system also contains a control center/security monitor which is employed for control functions and security policy. The last 7 sets of blocks on the reliability block diagram correspond to the "CUP Group" block on the functional flow diagram.

An advantage of R&MA^{2}T^2 is its ability to model large systems with repair such as this example problem. This system also includes dissimilar redundancy, as shown in the final block of Figure 3-11, which is easily modeled with this program. The Fortran code used to model this block is shown in Figure 2-12.

```
BLOCK (1) = 'Q1"
FA1 = .0002857
FA2 = .0001923
RA1 = .395
RA2 = .429
Q1F = SERIES (2)
Q1R = SREP
Q1M = SMTF
BLOCK (1) = 'Q'
FA1 = Q1F
FA2 = .0002857
RA1 = Q1R
RA2 = .395
QF = PARL (2,1)
QR = SREP
QM = SMTF
```

Figure 3-12

Sample R&MA^{2}T^2 Coding

Block "Q1" is defined as the top section of the block with the 2 components MTSS1 and MTSS2. The failure and repair rates FA1, FA2, RA1, and RA2 are then defined by the first 4 Fortran statements. Q1F is the failure rate for the series combination of the two components, while Q1R and Q1M are the repair rate and mean time to failure of the combination. Block "Q" is the entire parallel block, including the series combination and the single component MTSS1. The failure rate for the top section is defined as Q1F from the previous section, and the repair rate for the top section is defined as Q1R. The failure rate for the parallel block uses the PARL function for a 1 of 2 configuration with dissimilar redundancy.

Figure 3-13 is an example of the output provided by R&MA^{2}T^2. This figures shows the results for block "Q" and for the CCS system "S". R&MA^{2}T^2 provides results in the same form for every block in the series system. The 3 states provided for block "Q" refer to the times when the bottom half is operational, the top half is operational, or the whole block is operational. This is reflected in the state configurations shown.

```
        REL. PRED. OF BLOCK          Q

                        OPERATIONAL STATE INFO.

                    REQUIRING     1 OPERATIONAL BRANCHES

  STATE    STATE FAILURE RATE      STATE AVAILABILITY     STATE CONFIG.

    1          0.285700E-03       0.116964940E-02       0 1
    2          0.478000E-03       0.721922370E-03       1 0
    3          0.000000E+00       0.998107582E+00       1 1

    BLOCK STEADY STATE FR.   ***   1 OPERATIONAL BRANCHES=0.679248283E-06

    BLOCK STEADY STATE MTBF.  =0.147221572E+07

    BLOCK AVAILABILITY = 0.999999154E+00

    BLOCK MEAN REPAIR RATE = 0.802896072E+00


    BLOCK MEAN TIME TO FIRST FAILURE IS EQUAL TO OR GREATER THAN ==
                                                    0.147352514E+07

**********************************************************************

        REL. PRED. OF BLOCK          S

    BLOCK STEADY STATE FR.     = 0.672146459E-03

    BLOCK STEADY STATE MTBF.  =0.148777099E+04

    BLOCK AVAILABILITY = 0.998461396E+00

    BLOCK MEAN REPAIR RATE = 0.436182739E+00

    BLOCK MEAN TIME TO FIRST FAILURE IS EQUAL TO OR GREATER THAN ==
                                                    0.148931846E+04
```

Figure 3-13

LAN R&MA^{2}A^{2}T Output

4.0 OTHER RELIABILITY MODELING TOOLS FOR FAULT TOLERANT SYSTEMS/TOOL SURVEY

Research was conducted to survey other existing tools developed for computing the reliability characteristics of fault tolerant systems. There are numerous tools scattered around in Government, industry and academia for this purpose. This section outlines our initial efforts of surveying and searching for those existent in the field. Some of the tools listed below will be explored further for inclusion in the RADC Fault Tolerant System Reliability Evaluation Facility. This section is also aimed at providing a "dictionary" of fault tolerant system reliability modeling tools. Listed below in alphabetical order are short summaries of many of the existent tools:

1. ARIES - Automated Reliability Interactive Estimation Systems

Developer/Sponsor: Ng and Arizienis of UCLA

Model/Solution Type: Homogoneous Markov

Abstract: Written in "C" language. Models repairable and non-repairable systems. Accounts for transient fault recovery and scheduled preventive maintenance. Outputs include R(t), MTFF, system failure rate, subsystem reliability contributions, reliability improvement factors resulting from redundancy and various state probabilities.

2. ARM - Automated Reliability Modeling

Developer/Sponsor: Liceaga and Siewiorek of Carnegee Mellon University

Model/Solution Type: Markov Model Generator

Abstract: This tool is presently under development. ARM will generate reliability and availability Markov models for arbitrary interconnection structures at the processor-memory-switch (pms) level. The ARM output will be a file containing state transition matricies of which can be used to serve as input to other evaluation programs.

3. ASSIST - Abstract Semi-Markov Specification Interface to the SURE Tool

Developer/Sponsor: R. Butler of NASA Langley

Model/Solution Type: Markov Model Generator

Abstract: ASSIST allows the user to specify the behavior rules of a model and then generates the semi-Markov model automatically. The semi-Markov model output of ASSIST is formatted so that it can be used for input to the SURE program (see below). For programs requiring a different form of input than SURE, a simple program could be written to modify the model description file.

4. CAME - Computer Aided Markov Evaluator

Developer/Sponsor: Charles Stark Draper Laboratory

Model/Solution Type: Markov Analysis

Abstract: CAME is a relatively new computer-aided engineering tool that takes as input a graphical representation of the system and its operating requirements and automatically generates the Markov reliability model for analysis.

5. CARE III - Computer-Aided Reliability Estimation

Developer/Sponsor: Codeveloped by NASA Langley and Raytheon Company

Model/Solution Type: Semi-Markov Analysis

Abstract: Written in FORTRAN under VMS. Models nonrepairable systems only. Accounts for imperfect fault handling, i.e. probabilities associated with fault detection, fault generation, single point failures, permanent, transient and intermittent faults. Outputs include R(t), various other state probabilities.

6. CRAFTS - Computer-Aided Reliability and Availability Analysis of Fault Tolerant Systems

Developer/Sponsor: C.S. Raghavendra of Raghavendra and Associates

Model/Solution Type: Markov Analysis

Abstract: CRAFTS grew out of research done on the ARIES tool. Can model both repairable and nonrepairable systems. Models large systems, imperfect coverage, dependencies among subsystems and multilevel redundancy. Outputs include MTFF, MTBF, R(t), Availability and Safety. An interesting feature is that input specification can be by one of 3 ways: configuration and failure rate parameters; symbolic reliability block diagrams; or Markov state diagrams with transition rate matrices.

7. DEEP - Duke Evaluator for Extended Stochastic Petri Nets

Developer/Sponsor: Duke University

Model/Solution Type: Extended Stochastic Petri Nets

Abstract: No data

8. FASTER - Fault Tolerant Architecture Simulation Tool for Evaluating Reliability

Developer/Sponsor: Sanders Associates for RADC contract

Model/Solution Type: Monte Carlo Simulation

Abstract: Written in FORTRAN 77. Models repairable and non-repairable systems. Uses a constraint directed assistant concept for system specification. Flexibility allows for modeling of most traits and scenarios inherent to fault tolerance. Outputs include R(t), MTFF, Avail, MTBCF or others which can be computed by tailoring program to needs. Program is brand new.

9. FTC - Fault Tree Compiler

Developer/Sponsor: NASA Langley and NASA Washington

Model/Solution Type:

Abstract: FTC was designed to provide the user with a tool which can readily describe even the largest fault tree and then calculate the probability of the top event in the tree. The motivation of FTC began when it was observed that CARE III was being used to solve fault trees. FTC is a newer and faster method of solving fault trees.

10. GRAMP - Generalized Reliability and Maintainability Program

Developer/Sponsor: Systems Control Technology

Model/Solution Type: Markov Analysis

Abstract: Written in FORTRAN 77. Models repairable and non-repairable systems. Accounts for imperfect coverage, preventive maintenance, acquisition cost, operations cost and support cost. Outputs include a cost evaluator model, R(t), MTTF, MTBF, MTTR, and others.

11. GRAMS - Generalized Reliability and Maintainability Simulator

Developer/Sponsor: System Control Technology

Model/Solution Type: Monte Carlo Simulation

Abstract: Written in FORTRAN 77. Computes reliability, maintainability, and life cycle cost for the same fault tolerant systems as GRAMP. The judicious use of a Markov (GRAMP) and a simulation model (GRAMS) for the same system takes advantage of both methodologies.

12. HARP - Hybrid Automated Reliability Predictor

Developer/Sponsor: Duke University and NASA Langley

Model/Solution Type: Markov Analysis

Abstract: Written in FORTRAN 77. HARP has seven fault error handling models which allows for flexible modeling of coverage. Models both repairable and nonrepairable systems. Outputs include R(t), probability of near coincident fault, sensitivity analysis, failure probabilities.

13. MARK 1 - Markov Modeling Package

Developer/Sponsor:

Model/Solution Type: Markov Analysis

Abstract: Written in PL/1. MARK 1 models nonrepairable systems whose characteristics can be modeled using Markov chains. This program is a pure Markov analysis tool. The user specifies the number of states in the model, gives a description of each state and the occupancy probabilities and transition rates between states. Outputs include various plots, i.e. state probabilities as a function of time and plots of MTBF.

14. METASAN - Michigan Evaluation Tool for the Analysis of Stochastic Activity Networks

Developer/Sponsor: Industrial Technology Institute

Model/Solution Type: Stochastic Activity Network, which is an extension of stochastic Petri Nets.

Abstract: This tool was designed to treat reliability and performance in an integrated fashion - termed "performability". Models both repairable and nonrepairable systems.

15. METFAC

Developer/Sponsor: No Data

Model/Solution Type: Markov Analysis

Abstract: Written in FORTRAN 77 and Pascal. Models both repairable and nonrepairable systems. Outputs from METFAC include many reliability, performance and cost related figures.

16. MIREM - MIssion REliability Model

Developer/Sponsor: Developed by The Analytic Sciences Corporation for an Air Force Human Resources Lab (AFHRL) contract.

Model/Solution Type: Equations and Algorithms

Abstract: Written in FORTRAN 77. MIREM can model both repairable and nonrepairable systems. Models for imperfect testability and switching. Outputs include R(t), phase-by-phase reliability, Avail, MTFF, MTBMA, reliability bounds and MTBCF under various repair scenarios.

17. OPT - Optimization

Developer/Sponsor: No Data

Model/Solution Type: No Data

Abstract: No Data

18. RELCOMP - No Data

Developer/Sponsor: No Data

Model/Solution Type: No Data

Abstract: No Data

19. REST - REliability Simulation Tool

Developer/Sponsor: RADC

Model/Solution Type: Monte Carlo Simulation

Abstract: Written in FORTRAN 77. REST models both repairable and nonrepairable systems. It can account for failure rates and modes associated with switching and diagnostic circuitry. Outputs of REST include R(t), MTBCF, Avail, and MTTR. An interesting feature is the ability to reassess the dec' risks of MIL-STD-781 "Reliability Design Qualification and Production Acceptance Tests" as they apply to fault tolerant/redundant systems.

20. RMC-1 - Monte Carlo Multipurpose Code Package

Developer/Sponsor: Malchi Science Ltd. Tel Aviv, Isreal

Model/Solution Type: Monte Carlo Simulation

Abstract: RMC-1 is a package which contains three independent modules. The three modules are named AMIR, SPAR and ANAVA. AMIR is intended for the calculation of time dependent reliability, availability, sensitivities and mission success probability. SPAR is intended to calculate and analyze logistics requirements for a complex multisystem field. ANAVA is designed to conduct a statistical analysis of complex distributed communication lines.

21. R&MA^{2}T^2 - Reliabilaity, Maintainability and Availability Analysis and Tradeoff Tool

Developer/Sponsor: RADC

Model/Solution Type: Equations & Algorithms

Abstract: Written in FORTRAN IV. R&MA^{2}T^2 is an old program used to calculate the steady state availiability, MTBF, MTTTR, and MTFF for straight series-parallel systems. Multiple levels of redundancy can be modeled.

22. SAVE - System Availability Estimator

Developer/Sponsor:

Model/Solution Type: Both Markov and Monte Carlo Simulation

Abstract: Written in FORTRAN 77. SAVE models both repairable and nonrepairable systems. Typically used for high reliability and availability systems. Models can be solved both analytically and through Monte Carlo simulation. The Monte Carlo can be used either directly by generating random times to failure and repair or by simulating the Markov chain by generating state transitions randomly according to jump probabilities of the chain. Outputs include steady state availability, sensitivity analysis and system MTTF.

23. SHARPE - Symbolic Hierarchical Automated Reliability Performance Evaluator

Developer/Sponsor: K. Triredi of Duke University and R. Sahner of Gould Corporation

Model/Solution Type: 5 model types: (1) series parallel reliability block diagrams, (2) fault trees, (3) Markov chains, (4) semi-Markov chains, and (5) series-parallel directed (acyclic) graphics.

Abstract: SHARPE was developed for the purpose of analyzing complex reliability models which produce large state-space problems when analyzed using Markovian modeling techniques. SHARPE uses a hybrid, hierarchical modeling framework. Hybrid in the sense that it uses both combinatorial and Markov modeling and hierarchical in the sense that it can use different kinds of modeling techniques along different levels of a fault tree. Outputs include R(t), reliability of selected components and system steady state availability.

24. SIP - State Interpreter Program

Developer/Sponsor: No Data

Model/Solution Type: No Data

Abstract: No Data

25. SPADE - No Data

Developer/Sponsor: Duke University

Model/Solution Type: No Data

Abstract: No Data

26. SUPER - System Used for Prediction and Evaluation of Reliability

Developer/Sponsor: AT&T Bell Laboratories

Model/Solution Type: Markov Analysis

Abstract: SUPER can model both repairable and nonrepairable systems. Features which can be modeled include series, parallel and wheatstone bridge structures, k-out-of-n cold standby systems, and any hierarchical combination of these. Outputs include R(t), instantaneous failure rate, mean and standard deviation of the time to first failure, availability and other maintenance information.

27. SURE - Semi-Markov Unreliability Range Evaluator

Developer/Sponsor: PRC Kentron Inc. and later generalized by NASA Langley

Model/Solution Type: Markov Analysis

Abstract: Written in FORTRAN 77. The SURE program provides a rapid computational capability for semi-Markov models useful in describing the fault handling behavior of fault tolerant computer systems. The reliability analysis method utilizes a fast approximation theory developed by PRC to calculate the upper and lower bounds on system reliability. The upper and lower bounds are typically within 5% of each other. Since the computation method is extremely fast, large state spaces are not a problem. Therefore, state aggregation techniques are not utilized. Outputs include upper and lower bounds on reliability, probability bounds for each failure state in the model, list of every path in the model and its probability of traversal.

28. SURF -

Developer/Sponsor: Laboratoire d'Automatique et d'Analyse des Systemes du CNRS in France

Model/Solution Type: Markov Analysis

Abstract: Models both repairable and nonrepairable systems. SURF transforms non-Markov processes into Markov processes by using what is known as the Coxian method of stages which adds fictitous states to the model. SURF utilizes state merging and truncation of states in an attempt to avoid the problem of state explosion. Outputs include R(t), Avail, and some maintainability figures of merit.

Section 5 SUMMARY AND CONCLUSION

This report is concluded by reemphasizing the available services of the RADC Fault Tolerant System Reliability Evaluation Facility; and again, stating the need for standardization in the area of reliability modeling for fault tolerant systems.

As illustrated within this report, RADC has the tools and expertise necessary to provide reliability modeling support to other DoD agencies. If you have a modeling need or would like to learn more about any of the tools discussed, please contact Joseph A. Caroli, RADC/RBET, Griffiss AFB NY 13441-5700.

The tool survey presented in Section 4 of this report sends out a clear message. There are many reliability modeling tools available for fault tolerant system analysis. Before spending time and money developing a new tool, first search to see if there is one already available to suit your needs. This document could help you to get started. It is also strongly recommended that steps are taken in the future to standardize this area.

B.0 BIBLIOGRAPHY

1. Bavuso, Salvatore J., "A User's View of CARE III", 1984 Proceedings Annual Reliability and Maintainability Symposium, Inst of Electrical and Electronic Engineers, Inc., c. 1984, pp. 382-389.

2. Bavuso, S.J., and Petersen, P.L., "CARE III Model Overview and User's Guide (First Revision)", NASA TM-86404, 1985.

3. Bavuso, S.J., Dugan, J.B., Trivedi, K.S., Rothman, B., and Boyd, M., "Applications of the Hybrid Automated Reliability Predictor", NASA Technical Paper 2760, Dec 1987.

4. Bivens, G., Born, F., Caroli, J., Hyle, R., "Reliability Demonstration Technique For Fault Tolerant Systems", Proceeding of 1987 Annual Reliability and Maintainability Symposium, Jan 1987, pp. 316-320.

5. Bryant, L.A., and Stiffler, J.J., "CARE III Version 4 Enhancements", NASA CR-177963, 1985.

6. Butler, R.W., "The SURE Reliability Analysis Program", NASA TM 87593, Feb 1986.

7. Butler, R.W., and White, A.L., "SURE Reliability Analysis", NASA TP 2764, Mar 1988.

8. Conroe, D.J. and Murn, S.J., "R/M/T Design For Fault Tolerance", RADC-TR-88-69, Vol I, Mar 1988, pp. 5-8.

9. Duhi, A., and Goldfield, A., "Introduction to the RMC-1 Monte Carlo Multipurpose Code Package", Malchi Science Ltd Report, 1988.

10. Flemming, R.E., Dolby, L.J., Hoff, R.L., "Fault Tolerant Design-To-Specs With GRAMP & GRAMS", Proceeding of 1984 Annual Reliability and Maintainability Symposium, Jan 1984, pp. 403-498.

11. Johnson, A.M., and Malek, M., "Survey of Software Tools for Evaluating Reliability, Availability, and Serviceability", AMC Computing Surveys, Vol 20, No. 4, Dec 1988, pp. 247-269.

12. Johnson, S.C., "Assist Users Manual", NASA TM 87735, Aug 1986.

13. Klion, J., "A Redundancy Notebook", RADC-TR-77-287, Dec 1977.

14. Lyne, G., "Reliability Maintainability And Availability Analysis Tradeoff Tool", RADC-TR-75-149, Jun 1975.

15. Martensen, A.L., "CARE III User Friendly Interface User's Guide", NASA CR-178251, 1987.

16. Martensen, A.L., and Ravuso, S.J., "Tutorial and Hands-On Demonstration of a Fluent Interpreter for CARE III", NASA TM-4011, 1987.

17. Martensen, A.L., and Butler, R.W., "The Fault Tree Compiler", NASA TM 89098, Jan 1987.

18. Nelson, V.P., and Carroll, B.D., "Tutorial: Fault Tolerant Computing", IEEE Computer Society, IEEE Cat. No. EH0254-3, 1987.

19. Reghavendra, C.S., "Computer-Aided Reliability and Availability Evaluation of Fault Tolerant Systems", User Reference Manual, Jan 1988.

20. Sahner, R.A., and Trivedi, K.S., "Reliability Modeling Using SHARPE", IEEE Transactions On Reliability, Vol R-36, No. 2, Jun 1987, pp. 186-193.

21. Trivedi, K.S., Dugan, J.B., Geist, R.M., and Smotherman, M.K., "Hyrbrid Reliability Modeling of Fault Tolerant Computer Systems", Computer and Electrical Engineering, Vol II, No. 23, 1984, p. 87.

22. Trivedi, K.S., Dugan, J.B., Geist, R.M., Smotherman, M.K., Rothman, B., Boyd, M., and Bavuso, S., "HARP: The Hybrid Automated Reliability Predictor, Introduction and Guide for Users", Sep 1986.

23. Veatch, M.H., and Gates, R.K., "Mission Reliability Model Users Guide", AFHRL-TR-86-35, Nov 1986.

24. Veatch, M.H., and McManus, J.C., "Integrated Communication, Navigation, and Identification Avionics" Impact Analysis", AFHRL-TR-85-20, Oct 1985.

MISSION
of
Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control, Communications and Intelligence (C^3I) activities. Technical and engineering support within areas of competence is provided to ESD Program Offices (POs) and other ESD elements to perform effective acquisition of C^3I systems. The areas of technical competence include communications, command and control, battle management information processing, surveillance sensors, intelligence data collection and handling, solid state sciences, electromagnetics, and propagation, and electronic reliability/maintainability and compatibility.