Verification and Validation of the SAPHIRE Version 4.0 PRA Software Package

Prepared by
T. W. Bolander, M. B. Calley, E. L. Capps, W. J. Galyean
J. L. Jones, S. D. Novak, C. L. Smith, A. D. Trusty,
R. L. VanHorn, L. M. Wolfram, S. L. Zeigler

Idaho National Engineering Laboratory
EG&G Idaho, Inc.

Verification and Validation of the SAPHIRE Version 4.0 PRA Software Package

Prepared by
T. W. Bolander, M. B. Calley, E. L. Capps, W. J. Galyean
J. L. Jones, S. D. Novak, C. L. Smith, A. D. Trusty,
R. L. VanHorn, L. M. Wolfram, S. L. Zeigler

Idaho National Engineering Laboratory
Managed by the U.S. Department of Energy

EG&G Idaho, Inc.
Idaho Falls, Idaho 83415

MASTER

ABSTRACT

A verification and validation (V&V) process has been performed for the System Analysis Programs for Hands-on Integrated Reliability Evaluation (SAPHIRE). SAPHIRE is a set of four computer programs that the Nuclear Regulatory Commission (NRC) has developed for the performance of probabilistic risk assessments (PRAs). These programs allow an analyst to perform many of the functions necessary to create, quantify, and evaluate the risk associated with a facility or process being analyzed. The programs included in this set are Integrated Reliability and Risk Analysis System (IRRAS), System Analysis and Risk Assessment (SARA), Models and Results Database (MAR-D), and Fault Tree/Event Tree/Piping and Instrumentation Diagram (FEP) graphical editor.

There were several steps to the V&V. A V&V plan was prepared to describe the process and criteria by which the V&V would be performed. The software requirements documentation was reviewed to determine the correctness, completeness, and traceability of the requirements. A user survey was conducted to determine the usefulness of the user documentation. Vital and non-vital features were identified and tested. The testing that was performed and the results that were obtained were documented.

The results indicate improvements are needed in the software requirements documentation and in the user documentation. The majority of the results from the testing were acceptable; however, some discrepancies between expected code operation and actual code operation were identified. The discrepancies were provided to the code developers. Modifications that will be made to SAPHIRE are identified.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

correct results from SAPHIRE. The non-vital features are graphical fault tree construction, graphical event tree construction, or database manipulation. Lack of a non-vital feature can add significant time to a PRA since the work that would have been performed by that feature will have to be done manually or through an alternate method.

For SAPHIRE 4.0, 81 vital features and sub-features were identified for testing. However, it was recognized that performing all 81 tests would not be possible within the time available. Vital features and sub-features were ranked as high, medium, or low. Only those vital features ranked as high were tested. The testing covered 49 vital features and sub-features.

Software test records were prepared for all vital features tested. Anomaly reports were prepared for all discrepancies identified between the way SAPHIRE should operate and actual operation. Discrepancies were categorized as either critical or non-critical. A discrepancy was categorized as critical if the results were incorrect or the potential existed for the results to be misleading. A discrepancy was categorized as non-critical if the results were correct but the option was difficult to use.

The majority of the results were determined to be acceptable. However, some critical and non-critical discrepancies were identified for both vital and non-vital features. The discrepancies were provided to the code developers. The following paragraphs discuss each discrepancy and the actions that will be taken.

The critical discrepancies occurred in four areas: Link Editor (in both vital and non-vital features), Base Case Update, Uncertainty Analysis, and Process Flags. For the Link Editor option, it was found that SAPHIRE is not correctly processing event tree sequence linkage rules (under certain conditions). This discrepancy cannot be corrected in the SAPHIRE 4.0 series.

The quantification results were correct when the Base Case Update option was tested; however, the report function did not identify whether the printout applied to the base case or the alternate case. Even though the quantification results are correct, this discrepancy was categorized as critical since the user could inadvertently use the wrong case. This discrepancy will be corrected with the release of SAPHIRE 4.17.

There were several discrepancies identified for the uncertainty analysis. SAPHIRE will allow the user to input improper uncertainty distribution parameters. Even though in some cases SAPHIRE will display an error message, it will still attempt to perform the uncertainty analysis. For the Monte Carlo and Latin Hypercube sampling, it also appears that SAPHIRE has an internal arithmetic precision limitation when distribution sampled values are small. In another case, all tests that were defined as multiples of the chi-square distribution did not give sampling results as a multiple of the chi-squared distribution. Instead, the results gave only the chi-square distribution part [e.g., the results were very close to what would be expected for $\chi^2(1)$, not $2\chi^2(1)$]. The anomaly identified for the gamma distribution is that with a probability value of 1E–4 and an uncertainty value of 10,000, the Monte Carlo sampling seemed to return the same value, and consequently the sampling distribution results were in error. For certain Beta distributions, SAPHIRE actions ranged from giving runtime errors to giving incorrect results. These discrepancies will be corrected with the release of SAPHIRE 4.17.

Under the Process Flags option, the Sensitivity Process Flag was not tested since it is an undocumented feature, and as such no criteria could be developed by which to measure results. This option allows the analyst to vary the probability of an event or group of events over a specified range and plot the change to core damage frequency. Even though this is a menu option, it will not operate by simply selecting the option from the menu. An anomaly report was generated because the potential exists

EXECUTIVE SUMMARY

The System Analysis Programs for Hands-on Integrated Reliability Evaluation (SAPHIRE) is a set of four computer programs that the Nuclear Regulatory Commission (NRC) has developed for the performance of probabilistic risk assessments (PRAs). These programs allow an analyst to perform many of the functions necessary to create, quantify, and evaluate the risk associated with a facility or process being analyzed. The programs included in this set are Integrated Reliability and Risk Analysis System (IRRAS), System Analysis and Risk Assessment (SARA), Models and Results Database (MAR-D), and Fault Tree/Event Tree/Piping and Instrumentation Diagram (FEP) graphical editor.

This report documents the verification and validation (V&V) of SAPHIRE 4.0. The primary objective of this project was to determine if the results produced by SAPHIRE are correct. A related objective was to complete the necessary documentation of the V&V according to accepted standards.

The V&V of SAPHIRE consisted of five major steps:

1. Preparation of a V&V plan

2. Evaluation of the software requirements

3. Evaluation of the user documentation

4. V&V testing

5. Documentation of test results and recommendations

The above steps are consistent with IEEE Standard for Software Verification and Validation Plans (IEEE Std 1012-1986). The SAPHIRE V&V Plan was prepared for the purpose of describing the process and criteria by which the V&V would be performed. It was also prepared in accordance with the IEEE Standard for Software Verification and Validation Plans.

The evaluation of the software requirements found that the SAPHIRE Functional Requirements Document needs to be rewritten so that requirements and design constraints are more clearly and concisely stated. It was also found that the methodology employed to control both documentation and software changes needs to be more rigorous.

A survey of users was conducted across NRC, commercial industry, and national laboratories to determine how well the user documentation is meeting the needs of the users. It is important that the user documentation be complete, correct, and understandable so that the code will not be used incorrectly and so that the user can use the code to its fullest potential. The survey requested users to rate the user documentation in the categories of helpfulness, technical content, organization, completeness, and overall usefulness. The results of the survey indicated that improvements are needed to the user documentation. The majority of users commented that the user documentation needs a much better discussion of how the code works. Several users suggested that a road map or flow path of how to use the code from a PRA analyst's point of view be included in the user documentation.

SAPHIRE features were evaluated and designated either as vital or non-vital features. Vital features are those that (a) affect the results of a PRA (core damage frequency, top contributors, etc.) and (b) are essential for completing a PRA analysis (fault tree analysis, event tree analysis, uncertainty analysis, etc.). Non-vital features are those whose malfunction or lack of function will not preclude obtaining the final

correct results from SAPHIRE. The non-vital features are graphical fault tree construction, graphical event tree construction, or database manipulation. Lack of a non-vital feature can add significant time to a PRA since the work that would have been performed by that feature will have to be done manually or through an alternate method.

For SAPHIRE 4.0, 81 vital features and sub-features were identified for testing. However, it was recognized that performing all 81 tests would not be possible within the time available. Vital features and sub-features were ranked as high, medium, or low. Only those vital features ranked as high were tested. The testing covered 49 vital features and sub-features.

Software test records were prepared for all vital features tested. Anomaly reports were prepared for all discrepancies identified between the way SAPHIRE should operate and actual operation. Discrepancies were categorized as either critical or non-critical. A discrepancy was categorized as critical if the results were incorrect or the potential existed for the results to be misleading. A discrepancy was categorized as non-critical if the results were correct but the option was difficult to use.

The majority of the results were determined to be acceptable. However, some critical and non-critical discrepancies were identified for both vital and non-vital features. The discrepancies were provided to the code developers. The following paragraphs discuss each discrepancy and the actions that will be taken.

The critical discrepancies occurred in four areas: Link Editor (in both vital and non-vital features), Base Case Update, Uncertainty Analysis, and Process Flags. For the Link Editor option, it was found that SAPHIRE is not correctly processing event tree sequence linkage rules under infrequent conditions. This discrepancy cannot be corrected in the SAPHIRE 4.0 series, since insufficient data are retained. This problem is to be corrected in the SAPHIRE 5.0 series.

The quantification results were correct when the Base Case Update option was tested; however, the report function did not identify whether the printout applied to the base case or the alternate case. Even though the quantification results are correct, this discrepancy was categorized as critical since the user could inadvertently use the wrong case. This discrepancy will be corrected with the release of SAPHIRE 4.17.

There were several discrepancies identified for the Uncertainty Analysis option. SAPHIRE will allow the user to input improper uncertainty distribution parameters. Even though in some cases SAPHIRE will display an error message, it will still attempt to perform the uncertainty analysis. For the Monte Carlo and Latin Hypercube sampling, it also appears that SAPHIRE has an internal arithmetic precision limitation when distribution sampled values are small. In another case, all tests that were defined as multiples of the chi-square distribution did not give sampling results as a multiple of the chi-squared distribution. Instead, the results gave only the chi-square distribution part [e.g., the results were very close to what would be expected for $\chi^2(1)$, not $2\chi^2(1)$]. The anomaly identified for the gamma distribution is that with a probability value of 1E-4 and an uncertainty value of 10,000, the Monte Carlo sampling seemed to return the same value, and consequently the sampling distribution results were in error. For certain Beta distributions, SAPHIRE actions ranged from giving runtime errors to giving incorrect results. These discrepancies will be corrected with the release of SAPHIRE 4.17.

Under the Process Flags option, the Sensitivity Process Flag was not tested since it is an undocumented feature, and as such no criteria could be developed by which to measure results. This option allows the analyst to vary the probability of an event or group of events over a specified range and plot the change to core damage frequency. Even though this is a menu option, it will not operate by

simply selecting the option from the menu. An anomaly report was generated because the potential exists for a user to select this menu option and think that an analysis has been performed. To actually use this option, a hidden key must be used to gain access to the sensitivity menus and "run" the analysis. For the "I" process flag, SAPHIRE generated unacceptable results for the case of solving without fault trees. SAPHIRE complemented the existing system cut set incorrectly. These items will be corrected or the options will be removed with the release of SAPHIRE 4.17.

Software test records were not prepared for the non-vital features testing. Instead, a test specification and a test procedure were developed, and items were checked off the test procedure as they were completed. All discrepancies identified with the non-vital features were documented in anomaly reports. For the non-vital features, only one discrepancy was categorized as critical. As with the vital features, the sequence linkage rules are not being correctly processed. All other discrepancies were categorized as non-critical.

ACKNOWLEDGMENTS

1. INTRODUCTION

1.1 Background

The System Analysis Programs for Hands-on Integrated Reliability Evaluation (SAPHIRE) is used in a variety of regulatory applications. SAPHIRE was developed by the Nuclear Regulatory Commission (NRC) to support applications in the areas of

1. Analyzing the risk implications of plant designs, systems operations, and procedures

2. Assessing the effectiveness of existing and proposed regulations, including backfits

3. Evaluating the significance of operational occurrences

4. Prioritizing generic safety issues, research and licensing programs, and inspection activities

5. Assisting the Committee to Review Generic Requirements (CRGR) in tracking the progress that required plant modifications made toward improved safety levels.

Examples of these areas include the resolution of generic safety issues (GSIs), examining risk-based inspection strategies, analyzing a number of multiplant action (MPA) items, and performing a sensitivity study on the significance of motor-operated valve (MOV) failure rates. Because of the extensive use of SAPHIRE in regulatory applications, it was determined that the need existed to verify and validate (V&V) the operation of the code to ensure that it is adequate to perform accident frequency and risk calculations.

The primary objective of this project was to determine via a structured and systematic V&V process that the results produced by SAPHIRE are correct, and if not, to identify any areas requiring correction. In doing this, the V&V

process would be documented according to accepted industry standards.

This report documents the V&V of SAPHIRE Version 4.16. It is the intent of the NRC to V&V subsequent versions of SAPHIRE. Lessons learned from the V&V of SAPHIRE 4.16 will be incorporated into the V&V of subsequent SAPHIRE versions.

1.2 Description of the SAPHIRE Code Package

SAPHIRE is a set of four computer programs that the NRC has developed for the performance of probabilistic risk assessments (PRAs). These programs allow an analyst to perform many of the functions necessary to create, quantify, and evaluate the risk associated with a facility or process being analyzed. The programs included in this set are Integrated Reliability and Risk Analysis System (IRRAS), System Analysis and Risk Assessment (SARA), Models and Results Database (MAR-D), and Fault Tree/Event Tree/Piping and Instrumentation Diagram (FEP) graphical editor. These programs are "hands-on" in the sense that they execute on an IBM-compatible PC, and they are integrated in the sense that they can communicate with one another through the MAR-D database.

IRRAS is a program developed for the purpose of performing those functions necessary to create and analyze a complete PRA. It includes functions to allow the user to create event trees and fault trees, to define accident sequences and basic event failure data, to solve system and accident sequence fault trees, to quantify cut sets, and to perform uncertainty analysis on the results. Also included are features that allow the analyst to generate reports and displays that can be used to document the results of an analysis. Since this code is a very detailed technical tool,

the user should be familiar with PRA concepts and the methods used to perform these analyses.

SARA is a program that allows the user to review the results of a PRA and to perform limited sensitivity analysis on these results. It is limited primarily to the extent that changes in the plant model can be accommodated by using a cut set editor feature. If other than simple changes are being simulated, then IRRAS should be used so that new cut sets can be accurately generated. There are limitations with the use of truncated cut sets. For example, in performing a sensitivity study that increases basic event probabilities, it may be possible that cut sets previously truncated would now become dominant if a cut set generation was performed instead of a cut set update as in SARA.

MAR-D is a models and results database. This program defines a common relational database structure that is used by the entire set of programs. This structure allows all of the software to access and manipulate data created by other software in the system without performing a lengthy conversion. Hence, data created by IRRAS are immediately available to SARA for

sensitivity analysis. The MAR-D program also provides the facilities for loading and unloading PRA data from the relational database structure used to store the data. A simple ASCII data format is used for interchange with other PRA software not included in SAPHIRE. This feature allows for maximum data interchange. Elements of the MAR-D software are included with both IRRAS and SARA to allow these programs to load and unload data in the MAR-D format.

FEP is a program developed to provide a common access to the set of graphical tools developed for performing PRAs. These tools include the graphical event tree, fault tree, and piping and instrumentation diagram (P&ID) editors. The event tree and fault tree editors are available through IRRAS; however, the P&ID editor is only accessible through FEP. The event tree editor allows the analyst to construct and modify graphical, event trees. The fault tree editor allows the user to construct and modify graphical fault trees. The P&ID editor allows the user to construct and modify plant drawings. These drawings can then be used to document the modeling used in a PRA.

2. APPROACH

The V&V of SAPHIRE consisted of the following steps:

1. Preparation of a V&V plan

2. Evaluation of the software requirements specification

3. Evaluation of the user documentation

4. V&V testing

5. Documentation of test results and recommendations.

The above steps are consistent with IEEE Standard for Software Verification and Validation Plans (IEEE Std 1012-1986).[1] The SAPHIRE 4.0 Verification and Validation Plan was prepared to describe the process and criteria by which the V&V was to be performed. The software requirements specification was evaluated to determine the adequacy of the documentation. The third step was to evaluate the user documentation. To perform this step, a user survey was conducted to determine how well the user documentation is meeting the needs of the users. The fourth step, which was the major effort of the project, was to design and perform software tests.

SAPHIRE features were evaluated and either designated as vital or non-vital features. Vital features are those features essential for completing a PRA analysis such as fault tree analysis, event tree analysis, or uncertainty analysis. For vital features, test procedures were developed that specified input, execution conditions, and predicted results. Non-vital features are those features whose malfunction or lack of function will not preclude obtaining results from the SAPHIRE codes. Non-vital features include fault tree construction, event tree construction, and data base manipulation. Lack of a non-vital feature would increase the user time required since the work done by that feature would have to be done by other means. Non-vital features were tested for functionality and recorded in a checklist fashion. For both vital and non-vital features, anomaly reports were generated for discrepancies between expected and actual code operation.

This report documents the V&V of SAPHIRE Version 4.0. SAPHIRE 4.0 was originally released to the Energy Science and Technology Software Center (ESTSC) in March 1992. To correct software errors, interim versions of SAPHIRE 4.0 were released to ESTSC. SAPHIRE 4.15 was released in November 1992 and SAPHIRE 4.16 was released in July 1993. To determine the version of SAPHIRE 4.0 on a computer, the user can enter the PRATOOLS.B1 subdirectory in DOS, type dir *.exe, and press <Enter>. This will produce a list of files. The last column will contain the time of file creation (e.g., 4.15a).

This report documents the V&V of SAPHIRE 4.0. Because the V&V was performed over an extended time, different interim versions were tested. The non-vital features were tested on SAPHIRE 4.15. The vital features were tested on SAPHIRE 4.16. Even though the non-vital features were tested on an earlier version of SAPHIRE, the non-vital features were not re-tested on SAPHIRE 4.16 since non-vital features do not affect the accuracy of the numerical results. No additional important information would be gained from re-testing the non-vital features.

The following sections discuss in greater detail each of the above steps that were performed and the results that were obtained. As applicable, the version date used for performing the particular step in the V&V effort is specifically identified.

3. SAPHIRE V&V PLAN

The SAPHIRE V&V Plan was prepared for the purpose of describing the process and criteria by which the V&V would be performed. The SAPHIRE V&V Plan was prepared in accordance with IEEE Standard for Software Verification and Validation Plans (IEEE Std 1012-1986). This standard provides uniform and minimum requirements for the format and content of software verification and validation plans. Even though this standard applies to all phases of the software life cycle from the concept phase to the operation and maintenance phase, the SAPHIRE V&V Plan was tailored to apply to SAPHIRE that is in the operation and maintenance phase of the software life cycle.

The SAPHIRE 4.0 V&V Plan, provided in Appendix A, was prepared and completed in April 1992. The V&V plan provides the purpose, other binding compliance documents, and an overview of the V&V, including top level discussions of the V&V tasks that will be performed as well as those that will not be performed and of the software features to be tested. The V&V plan also addresses reporting and administrative procedures.

4. EVALUATION OF THE SOFTWARE REQUIREMENTS SPECIFICATION

As a part of the V&V activities performed on SAPHIRE 4.0, the documentation describing the functionality of and the requirements for the software were reviewed. Only one document, other than user manuals, describing the functionality of IRRAS was delivered to the V&V team. This document, "Integrated Reliability and Risk Analysis System (IRRAS), Version 4.0, Functional Requirements Document (DRAFT)," was reviewed for the purpose of evaluating the software requirements specification.

The IRRAS 4.0 Functional Requirements Document was reviewed by three individuals. The major review was performed by E. L. Capps, a senior engineer whose area of expertise is software engineering and D. L. Kelly, a senior engineering specialist in PRA. S. D. Matthews, a senior engineering specialist in the area of software quality/V&V, also provided comments on the IRRAS 4.0 Functional Requirements Document. Since E. L. Capps' review encompasses the findings from the other two reviews, it forms the basis for review of the functional requirements document. Section 4.1 is a synopsis of the findings and recommendations from that review. All three reviews are provided in Appendix B in their entirety and provide more details on specific changes that should be made.

In addition to the review of the IRRAS 4.0 Functional Requirements Document, a Software Metrics Analysis was also performed. The detailed metrics analysis is provided in Section 4.2.

4.1 Findings and Recommendations from the Review of the IRRAS 4.0 Functional Requirements Document

Five topics are addressed in the following synopsis of the review of the Functional Requirements Document: functional requirements documentation, system documentation, document and software configuration management, user documentation, and use of the functional description for V&V testing purposes.

4.1.1 Functional Requirements Documentation. The IRRAS 4.0 Functional Requirements Document is neither a functional description (FD) nor a system requirements specification (SRS). A functional description provides the reader an overview or a description of the functions to be performed by the system without providing low level details. An SRS provides detail of WHAT the system is to do without describing HOW it will be accomplished. How the system is to perform is addressed in the software design document (SDD). The IRRAS 4.0 Functional Requirements Document appears to be a combination of an FD, an SRS, an SDD, and a description of an earlier version of IRRAS.

The IRRAS 4.0 Functional Requirements Document definitely presents numerous requirements for IRRAS. However, the document is written as though it is describing changes or enhancements to an earlier version of IRRAS. Several sections contain wording such as "from now on" that clearly indicate that the document, or at the very least, portions of the document, is describing an existing system without clearly identifying new functionality, enhancements, or changes to the system.

It would be desirable if the IRRAS 4.0 Functional Requirements Document were rewritten in a requirements format. It is recommended that this be considered for future versions. The requirements should be clearly and concisely stated. Design constraints should be clearly identified. In many sections, simply rewriting the requirement as a statement of what must be accomplished will provide the necessary clarity.

Using the rewritten requirements documentation and existing code, a SDD should be prepared for future versions. IRRAS is a powerful and extremely useful system. IRRAS will continue to grow and will require that changes be implemented as technological advancements

are incorporated into more powerful computer hardware components. Preparing an SDD will greatly facilitate applying changes and enhancements to IRRAS in future versions. The lack of appropriate system documentation will be the source of deficiencies as the software matures.

4.1.2 System Documentation.

Documentation should be prepared for persons that will be required to maintain the system. This documentation is referred to as system documentation and programmer documentation. Currently, the individuals who are responsible for the maintenance of IRRAS have been intimately involved with the development of the system. Their knowledge and understanding of the system is extensive and quite thorough. However, there is absolutely no guarantee that these individuals, or individuals with equivalent knowledge and understanding of the system, will always be available. System documentation detailing system and performance requirements and restrictions should be prepared and maintained so that other individuals, when required, will be able to understand the system adequately and continue developing, enhancing, and maintaining the system.

The system documentation, when combined with an SRS and SDD, will provide a thorough view of IRRAS. The formal reporting and logging of discovered deficiencies and applied resolutions will provide a clear view of the growth and maturity of the system.

4.1.3 Document and Software Configuration Management.

It is clear that a number of software change requests have been submitted to the development team. However, it is not known if these requests were to resolve a discrepancy discovered in the software or to request an enhancement or change to the system.

The methodology employed to control both documentation and software changes needs to be more rigorous. Changes to software should be reflected in system and user documentation as new versions are released and deficiencies corrected.

4.1.4 User Documentation.

All user documentation should be updated to reflect the status of the most recently released version. User documentation updates should be provided as deficiencies are corrected and enhancements introduced. As new versions of the system are released, new versions of (or additions to) user documentation should also be released.

4.1.5 Use of the Functional Description for V&V Testing Purposes.

The review of the functional requirements document determined that it would be possible to extract considerable information from this document that would be useful in preparing test cases and test procedures. However, it was also determined that many implied requirements would be overlooked because they are not explicitly stated. As a result, some functionalities would not be tested if the testing relied only on this document.

The reason for this is that the document does not provide the detail necessary to provide the traceability necessary to perform thorough testing of a software system. The document provides considerable information, but too much detail must be inferred from improperly stated requirements. Such a subjective interpretation of requirements would cause the introduction of biases into test results.

Even though the Functional Requirements Document did not provide adequate information to develop a test procedure, it was used to the extent possible to define the criteria for such features as the maximum number of cut sets saved, minimal cut set upper bound, and split fraction.

4.2 Software Metrics Analysis of IRRAS 4.0

Software Metrics Analysis consists of the application of various techniques to measure certain aspects about a software program. As with anything produced through an applied process, programmers need to be able to provide the user, or customer, some measure that will indicate the level of confidence that can be placed in the product, how well it will function, and how complicated the program is.

Software metrics can provide the programmer useful feedback in identifying areas of the code that may be too complicated and may need to be rewritten, areas that should be considered for further refinement into sub-programs, and the complexity of the code. These same measures will also provide guidance for those analyzing or testing the program and the program's output. Software metrics can provide guidance as to which portions of the code are the most complicated and therefore require more attention during testing efforts. Performing software metrics to determine which portion of the code performs most of the work identifies those sections of the code that are most important to the successful operation of the program. Extra attention can then be focused on those sections of the code during V&V efforts.

The following sections provide a discussion of the application of software metrics to IRRAS. Section 4.2.1 provides an overview of the types of software metrics and their meaning. Section 4.2.2 provides an overview of the application of software metrics. Section 4.2.3 provides details of the methodology employed on IRRAS, the data obtained from the analysis, and a detailed discussion of the findings. Section 4.2.4 provides an interpretation of the analysis results and draws conclusions based on the analysis data obtained. Section 4.2.5 presents recommendations based on the conclusions.

4.2.1 Types of Measures. Software metrics are measures extracted from program source code and executable code to provide information about the code and the program. These measures can be static or dynamic.

Static metrics are measurements obtained without executing or running the program. The source code is examined and analyzed to extract certain measurements. The most familiar static measurement is the lines of code, or how long the source code is. This metric provides a 'feel' for how big the code is. Other metrics identify the number of unique operators and operands employed in the program source code and the variables that are defined but never used. Other tools are employed to determine if portions of the code are unreachable or if there have been

subprograms developed that are never referenced or called by the main program.

Dynamic metrics are extracted through processes that involve executing the program. As the program is running, the metric analysis tools obtain information about the program and how it is executed. One type of dynamic measure counts the number of times a certain portion of the program is actually executed. Other dynamic analysis tools can identify which portions of the code perform the most work, if certain logic branches are never executed, or if all statements have been executed through the application of test data sets.

Early attempts to identify measures, or software metrics, included the number of lines of code (LOC). This metric identifies how big a program is, but does little to indicate how difficult or complex the program is. The following metrics have been defined and implemented to provide more information and direction.

Halstead's Metrics. The first set of data to be reviewed includes a set of metrics developed by Halstead.[2] Halstead maintained that a simple count of lines of code would not provide an accurate measure of the difficulty or complexity of the code. All programs are composed of operators and operands, things that perform actions and things that have actions performed on them. Halstead proposed that measuring the number of operators and operands would provide a better indication of the complexity of the program.

Halstead defined four parameters, upon which he based other metrics:

> n1 - the number of unique operators
> n2 - the number of unique operands
> N1 - the total number of operators
> N2 - the total number of operands.

Using these measures, he defined the following additional metrics: program vocabulary, program length, predicted length, purity ratio, and program volume, each of which is discussed below.

Program Vocabulary—The vocabulary of a program, n, is defined as the total number of unique operators and operands and is computed by the equation:

$$n = n1 + n2 .$$

Program Length—The length of the program, N, is the total number of operators and operands and is computed by the equation:

$$N = N1 + N2 .$$

Predicted Length—Predicted length, $\hat{N}$, is computed by the equation:

$$\hat{N} = [n1 * \log_2(n1)] + [n2 * \log_2(n2)] .$$

Purity Ratio—Halstead suggested that programs where the program length is not the same as the predicted length must contain impurities, with the purity ratio being:

$$PR = \hat{N}/N .$$

This is a measure of the degree to which impurities exist in a piece of code. A purity ratio of 1 indicates the existence of few impurities. The reported correlation seldom drops below 0.8 in studies involving traditional programming languages. When the purity ratio drops below this value, the reasons should be determined, if possible.

Program Volume—Program volume is computed by:

$$V = N * \log_2(n) .$$

If a program has n unique operators and operands, it will take $\log_2(n)$ "bits" to uniquely represent each. In this manner, two programs with an equal total number of uses of operators and operands can have a measure applied indicating that the program using more different operators and operands is more complex. Thus, volume can be used to indicate the complexity of the program.

Number of Errors Predicted—Using the metrics described above, software scientists

have developed formulas to predict the number of errors likely to exist in the code upon completion of the coding phase. The formula,

$$\hat{B} = [N * \log_2(n)] / E0,$$

is based on the tendency to make mistakes, on the average, every E0 mental comparisons. Work by psychologists indicates an appropriate value for E0 is approximately 3,000–3,200 comparisons.

It is obvious that the number of errors in a program is a factor of more than the number of operators and operands employed by the program. Other items, such as the programmer's familiarity with the programming language, the system in use, the ability of the programmer, and available support, obviously influence the actual number of errors. However, $\hat{B}$ provides a reasonable starting point.

McCabe's Cyclomatic Complexity. The cyclomatic complexity measure was developed by McCabe.[3] McCabe used an alternate approach to program complexity that considers the flow of control. McCabe believes that the cyclomatic number of a program's control flow graph provides an accurate measure of how complex the program is.

A control flow graph contains only nodes for each *basic block* within the code. A basic block is a segment of code that is entered only at one point (the top), exited at one point (the bottom), and has no transfers of control within it. Basic blocks often begin with decision-making statements and end immediately before another decision-making statement.

The cyclomatic number can be calculated as

$$V(g) = e - n + 2 ,$$

where n is the number of nodes in the graph and e is the number of edges or lines connecting each node.

The flow graph does not need to be created in order to compute the cyclomatic complexity. V(g) can also be computed by counting the

number of decision making statements and adding one (n+1). This process has been automated for many programming languages that have programs to "count" the decision statements or decision points within them.

Time of Development. The time required to develop a system may be connected to the complexity of the program(s) under development. However, this metric is introduced for other purposes at this time.

When considering system analysis and design, coding, and V&V activities, the time and resources should be divided 40%, 20%, and 40%, respectively.[4] This indicates that the amount of time, effort, and resources devoted to V&V activities should equal that dedicated to analysis and design. This also indicates the importance of analysis and V&V during the life cycle of software development.

4.2.2 Applying Software Metrics.

After identifying areas and items where metrics can provide insight into program complexity, how are these metrics to be used? This section describes how some of the metrics are applied to software and how they are interpreted.

Path Testing. McCabe's Cyclomatic Complexity measure is often used to judge how many path tests might be needed. Path testing involves identifying a path along the flow graph, determining the data necessary to execute each step in the path, and then executing the program to determine if the path is followed and the correct results achieved at each node, as well as the overall product as the result of following the defined path. Path testing strategies include

1. Path testing - where all possible control paths through the program are executed. Although this is the strongest criterion for path testing, it is generally impossible to achieve.

2. Statement testing - where all statements in the program are executed at least once under some test. This is the weakest criterion for path testing and should be considered as nothing more than the minimum testing activity.

3. Branch testing - where enough tests are executed to ensure that every branch alternative has been exercised at least once. Branch testing at 100% branch coverage will also provide 100% statement coverage.

McCabe's complexity measure indicates the number of branches to be tested. The Cyclomatic Complexity measure, therefore, indicates the minimum number of test cases to be considered to provide 100% branch and 100% statement coverage.

Module Testing. Modules with a higher complexity should receive more attention than those with lower complexity values. The more complex the code, the more likely it is to contain errors. The complexity measure is used to identify those areas that should receive more attention.

The complexity measure also assists in allocating resources during V&V and testing efforts.

Bug Prediction. Using the predicted number of bugs will certainly provide direction to those who will be testing the program! Modules and subprograms with a high number of bugs predicted should be exercised rigorously. The value should also be used by the programmer to examine the code before others begin testing it.

Halstead's Metrics & Code Analysis. Programs with very high counts in the number of unique operands, total number of operands, number of unique operators, and total number of operators should be examined more closely than those with smaller values. Obviously, the higher the language value, the higher the probability for errors to exist in the code simply because there are more operators and operands for the programmer to remember and use in the appropriate manner.

IRRAS Development Effort. Determining the level of effort required to bring IRRAS to its current state of development will assist in estimating the effort required to perform detailed and thorough V&V activities on the code.

Software Requirements Specification

4.2.3 IRRAS Software Metrics Preliminary Summary.
This section identifies the values of the metrics obtained when a software metric program was executed to obtain measures of the IRRAS code. The next section provides a brief discussion of the implications of these findings.

Methodology. The software metric tool, PC-Metric from Set Laboratories, Inc., was executed on only the Modula 2 code of IRRAS Version 4.14. Only the Modula 2 code developed specifically for IRRAS was evaluated by software metric analysis. It was determined that the number of modules written in Modula-2 specifically for IRRAS is 349. This does not include code to generate screens, display menus, nor the code of subprograms called from the Sage libraries. It was originally believed that IRRAS could be broken down into collections of modules and that each of the smaller collections could be analyzed separately. However, due to the numerous calls to and from procedures, it was determined that all 349 modules would have to be treated as a single program.

Data Produced. The metric analyzer generated two reports. The first was an exception report that identified, on a procedure by procedure basis within each module, which of the metrics exceeded accepted limits in the Modula-2 module. No summary by module or entire program was produced.

The second report generated contained Halstead's and McCabe's metrics. A summary was not provided for each module, but was provided for the entire collection of modules. The results are discussed in the following section.

Module Data—Regarding the overall size of the IRRAS Modula-2 files that were analyzed, the 349 files, consist of 4,916 procedures/functions.

Lines of Code—The total LOC is 228,333. Discussions with the programmers indicate that this is probably one-half of the total when the Sage libraries, menus, and screens are considered. Data from each individual procedure or module are not included in this analysis.

Purity Ratio—The purity ratio is a measure of the degree to which impurities exist in a piece of code, and it seldom drops below 0.8 for traditional programming languages. For IRRAS the purity ratio of the first set of 122 files is 0.37. The ratio for the second set of files is 0.33. These two numbers indicate that the combined purity ratio will be less than 0.40. This indicates that IRRAS should be investigated further since a low purity ratio indicates that the code is not written as concisely as possible. A low purity ratio indicates the potential for errors in the code.

Number of Procedures/Functions—The number of procedures and functions, 4,916, indicates the size and complex nature of the code. Continued analysis of the results will identify which of these procedures or functions warrant further examination and testing.

Complexity—The Cyclomatic Complexity of the files is 15,230, which can be used to approximate the number of decision points within the IRRAS Modula-2 code. This represents the minimum number of test cases required to exercise each decision point, which also results in exercising each statement.

A preliminary review of the procedures and functions indicates that some are of a very high complexity value. These procedures or functions will require additional examination since there is a correlation between complexity and the potential for errors in the code.

Development Effort—The IRRAS development team estimated that 25 person-years have been required to bring IRRAS to the current level of development. The development effort to reach the first release is obviously only a portion of the total. The current release is Version 4.14. If we use the very conservative estimate that an equal amount of time was spent for each release, then each release required 6.25 person-years.

4.2.4 Conclusions.
The data presented above, although not entirely complete, provide adequate information to form conclusions as to the effort and types of activities to be performed

during the V&V of the IRRAS and to which activities available resources should be allocated.

Cyclomatic Complexity Metric. The critical factor considered in these conclusions is the Cyclomatic Complexity values derived from the metrics. The combined value for all 349 files examined indicates that approximately 15,000 test cases must be developed to ensure that every decision point be exercised. As a result of exercising each decision point, each line of code would also be exercised. The 15,000 test cases would exercise roughly only one-half of the code that makes up IRRAS. Even ignoring the menus, and the screens, and the Sage libraries, or assuming that their prolonged use and initial testing have provided adequate testing to provide the necessary confidence that they do indeed function as required, 15,000 test cases would be needed to ensure 100% path and statement coverage.

The impact of developing such a high volume of test cases is adversely impacted by the lack of documentation that identifies the decision points and the paths that would be followed as a result of evaluating the decision mechanisms. Without such documentation, it is virtually impossible to conduct path testing that will provide results from which general conclusions can be derived. Without the documentation, test case developers and test executors will be unable to determine accurately which decision points are being exercised and which path is taken from a particular decision point.

The complexity and difficulty of preparing test cases is further compounded when the effect of applying a fault tree with more than a few nodes to the IRRAS is considered. The number of logic paths through a given set of decision points will increase with the number of nodes in the fault tree. The issue becomes even more complicated in considering that a decision point may be reached from multiple paths.

The data presented indicate that path testing may not provide adequate results. The lack of documentation and the high number of test cases required indicate that testing efforts and resourc-es would be better used if directed to other areas of testing and V&V activities.

Development Time. The development team indicated that development required approximately 25 person years. Since the system is currently in its fourth version, it can be safely estimated that at least 6 person-years were necessary to develop the first version. In performing V&V activities during the development cycle of a software system, the amount of time and resources expended on V&V activities should equal the effort and resources expended on system development. This practice provides an indicator of the effort required to perform thorough V&V activities, where testing is only one of many activities performed.

Therefore, a tremendous effort would have to be expended to perform thorough testing of IRRAS. According to the 40-20-40 rule, at least 6 person-years should have been devoted to system V&V for each version. Although the V&V of each version has not been performed to the suggested level of effort pre-release Beta testing has been performed. This testing in combination with the present V&V effort are steps in the right direction.

4.2.5 Recommendations. Based on the data and conclusion as presented above, the following recommendations are presented for consideration.

Path and Statement Testing. Instead of expending resources on attempts to perform thorough path and statement testing, resources should be applied to thorough acceptance testing, to continued identification of potential problem areas through software metric analysis, to the creation and execution of test cases to exercise code with a higher probability of errors, and to analyze the processes in place to provide configuration management and to ensure the correct and proper application of error detection, reporting, and correction.

Acceptance Testing. Acceptance testing should also be expanded to provide testers the opportunity to test areas and functions that, through application of their experience in testing

programs, are vulnerable to user errors that may cause program failure.

The test results from tests performed at other locations should be obtained and analyzed. The results of those tests will identify either accurate results, the need for additional testing in certain areas, potential errors, or a combination of all three. The results of tests performed by others could preclude the necessity of performing the same set of tests.

Level of Effort. The pre-release Beta testing and the current V&V effort are steps in the right direction for improving the level of effort on V&V activities. V&V activities are important and should focus on determining that the results produced by SAPHIRE are correct. The Beta testing needs to be more structured and documented. The V&V level of effort should be increased to ensure thorough testing of IRRAS.

4.2.6 Actions Taken for SAPHIRE 4.0 V&V.

Based on the above results and recommendations from the metric analysis, the SAPHIRE 4.0 V&V effort focused on acceptance testing. The acceptance testing was developed on the basis of defining vital and non-vital features, as is discussed in Section 6.0. Given the complexity of the code and the large number of test cases that would need to be developed, it was decided that path and statement testing was not feasible.

The above results from the metrics analysis may be due to the process that was used to develop SAPHIRE. From discussions with the code developers, SAPHIRE was not developed in the traditional "waterfall" process. Instead, it was developed in a "cyclic" process in which requirements and coding were developed as the overall code progressed.

5. EVALUATION OF USER DOCUMENTATION

It is important that the user documentation be complete, correct, and understandable so that the code will be correctly used to its fullest potential. Evaluation of the user documentation is recommended in IEEE Std 1012-1986. To evaluate the user documentation, a survey was conducted for the purpose of determining how well the user documentation is meeting the needs of the user. The survey questionnaire is shown in Figure 1. The first three questions of the survey focus on the PRA codes that are used, how frequently the codes are used, and which manual is used the most. The fourth question asks the user to rank the SAPHIRE documentation in five categories: helpfulness, technical content, organization, completeness, and overall usefulness. The fourth question asks the user to rank the documentation for the hypothetical "average" PRA code, and the fifth question asks the user to rank the SAPHIRE documentation against the documentation for the hypothetical "average" PRA code. The final question requests input on improvements the user would like to see made to the user documentation.

Users that participated in the survey represent such organizations as the NRC Advisory Committee on Reactor Safeguards (NRC-ACRS), NRC Office of Nuclear Reactor Regulation (NRC-NRR), Science Application International Corporation (SAIC), Nuclear Utilities Services (NUS), TENERA, TGR, Los Alamos National Laboratory (LANL), and Idaho National Engineering Laboratory (INEL). Of 25 users called, 14 participated in the survey.

In terms of type of organization, two users from the NRC participated, four users from the commercial industry, and eight users from the national laboratories with two from LANL and six from INEL.

Since the purpose of this survey was to determine how well the SAPHIRE documentation was meeting the needs of the users, all 14 participants were SAPHIRE users. This was closely followed by users of CAFTA and SETS, both with eight participants each.

Under frequency of use, SAPHIRE is more heavily used by the national laboratories, whereas CAFTA and SETS are more heavily used by the commercial industry. Of the 14 users, nine use SAPHIRE every couple of weeks to daily, but for five participants, it had been awhile since they had used it.

The full list of user's manuals is located in Table 1. The manuals that the participants cited for this survey are as follows:

IRRAS 2.5 (NUREG/CR-5300)
    4 participants
IRRAS 2.6 (Draft Report)
    1 participant
IRRAS 4.0 (NUREG/CR-5313)
    9 participants
SAPHIRE 4.0 (NUREG/CR-5964)
    1 participant
SARA 4.0 (NUREG/CR-5303)
    1 participant

One user rated the IRRAS 4.0 manual and the SAPHIRE 4.0 manual separately. Another user gave a combined rating for the IRRAS 4.0 and the SARA 4.0 manuals.

On a scale of 1 to 5, where 1 represents very poor and 5 represents very good, the IRRAS/SAPHIRE user documentation was rated as follows:

Helpfulness	2.1
Technical Content	3.1
Organization	2.9
Completeness	2.4
Overall Usefulness	2.3

If it is assumed that all categories are of equal weight, the results yield an average rating of 2.6. On the same scale, the hypothetical "average" PRA code documentation (excluding SAPHIRE) was rated at 3.0. When specifically asked to rank the SAPHIRE user documentation against the hypothetical "average" PRA code documentation, the SAPHIRE documentation was considered to be slightly less than equivalent to the

"average" PRA code documentation with the ranking of 2.5.

The last question dealt with what improvements the user's would like made to the documentation. The majority of users (93%) commented that the documentation needs a much better discussion of how the code works. This includes a better discussion of the hierarchy of the code (which part of the code has precedence over another part), and of the features and options (what is the code actually doing under event tree analysis, when flags are applied, or when an uncertainty analysis is performed). Most users also commented that the documentation needs to discuss how to use the code when performing a PRA. Several users suggested that the user documentation include a road map or flow path of how to use SAPHIRE from a PRA analyst's point of view.

Table 1. User's documentation for SAPHIRE.

K. D. Russell et al., *Integrated Reliability and Risk Analysis System (IRRAS) Version 2.5 Reference Manual*, NUREG/CR-5300, EGG-2613, February 1991.

K. D. Russell et al., *Integrated Reliability and Risk Analysis System (IRRAS) Version 2.6 Reference Manual*, Draft, December 1991.

K. D. Russell et al., *Integrated Reliability and Risk Analysis System (IRRAS) Version 4.0, Volume 1, Reference Manual*, NUREG/CR-5313, EGG-2664, January 1992.

K. D. Russell et al., *SAPHIRE Technical Reference Manual: IRRAS/SARA Version 4.0*, NUREG/CR-5964, EGG-2694, December 1992.

K. D. Russell et al., *System Analysis and Risk Assessment System (SARA) Version 4.0, Volume 1, Reference Manual*, NUREG/CR-5303, EGG-2628, February 1992.

USER DOCUMENTATION QUESTIONNAIRE

Date _____

User Name _____

Company _____

NRC _____ Commercial _____ Internal _____ Other Lab _____

1. What PRA codes are you using or have you used in the past?
    (circle)

IRRAS	SARA	MAR-D	FEP	P&ID ED
CAFTA	SETS	NUPRA	REBECCA	RISKMAN
GRAFTER	T-MAC	SANET	C-TREE	AT RISK
UNIRAM	LHS	EVNTRE	PSTEVNT	F-TAP
WAM				

2. Order codes used from least used to most used:

    _____ 1 2 3 4 5 New User

    _____ 1 2 3 4 5 New User

    _____ 1 2 3 4 5 New User

    _____ 1 2 3 4 5 New User

    _____ 1 2 3 4 5 New User

    1 - use very little or has been awhile
    2 - use occasionally such as every couple of months
    3 - use at least once a month
    4 - use every couple of weeks
    5 - use daily or weekly

Figure 1. User documentation questionnaire.

3.   Which SAPHIRE user documentation do you use the most? (circle)

        IRRAS User Guide 1.0 (NUREG/CR-4844)
        P&ID Editor 1.0 (Draft)
        Event Tree Editor User's Guide March 1989
        MAR-D User's Guide 2.0 (EGG-CATT-8249)
        Procedures Guide for Loading PRA Data into MAR-D (NUREG/CR-5520)
        IRRAS User's Guide 2.0 (NUREG/CR-5111)
        IRRAS Reference Manual 2.5 (NUREG/CR-5300)
        IRRAS Reference Manual 2.6 (Draft)
        IRRAS Reference Manual 4.0 (NUREG/CR-5313)
        IRRAS 4.0 Tutorial (NUREG/CR-5313)
        SARA Reference Manual 4.0 (NUREG/CR-5303)
        SARA 4.0 Tutorial (NUREG/CR-5303)
        MAR-D Reference Manual 4.0 (NUREG/CR-5301)
        FEP Reference Manual 4.0 (NUREG/CR-5866)
        SAPHIRE Reference Manual 4.0 (NUREG/CR-5964)

4.   On a scale of 1 to 5, with 1 being very poor and 5 being very good, how would you rank SAPHIRE documentation as far as:

Helpfulness	1	2	3	4	5
Technical Content	1	2	3	4	5
Organization	1	2	3	4	5
Completeness	1	2	3	4	5
Overall Usefulness	1	2	3	4	5

Figure 1. (continued).

5.   On a scale of 1 to 5, with 1 being very poor and 5 being very good, how
     would you rate user documentation for the hypothetical "average" PRA code
     excluding SAPHIRE from the "average?"   (circle)

     1   2   3   4   5 New User

6.   In comparison to the user documentation for the hypothetical "average" PRA code,
     do you think that SAPHIRE user documentation is: (circle)

     Far worse   1
     Worse   2
     Equivalent   3
     Better   4
     Far better   5

7.   What improvement would you like to see in the SAPHIRE user
     documentation?

Figure 1. (continued).

6. SOFTWARE TESTING OF VITAL FEATURES

SAPHIRE features were evaluated and designated either as vital or non-vital features. Vital features are those features that (a) affect the results of a PRA (core damage frequency, top contributors, etc.) and (b) are essential for completing a PRA analysis, such as fault tree analysis, event tree analysis, uncertainty analysis, change set features, and importance measures. The overall purpose of the vital features testing was to determine if SAPHIRE 4.0 will produce the correct results—results that if not correct could affect the findings and interpretation of a PRA.

The testing of SAPHIRE 4.0 was limited to within the normal conditions of a PRA. The testing did not include such items as entry of negative or bogus numbers, typing in words where numerical data are expected, or of the ability of SAPHIRE to recover from power failures or similar system failures.

Section 6.1 discuss how the vital features were established and Section 6.2 provides a summary of the vital features testing that was performed.

6.1 Establishment of Vital Features

Table 2 presents the original list of vital features. The list was developed by outlining the major functions that are generally performed in a PRA. These include fault tree analysis, event tree analysis, uncertainty analysis, change set features, cut set editor, and importance measures. These functions are shown in the first column of Table 2. Vital features were then determined for each of these functions. The vital features are shown in the second column of Table 2 and include such items as cut set generation, quantification, event tree sequence generation, and applying data change sets.

Not all of the documentation for the vital features testing was performed consistently with the IEEE Standard for Software Test Documentation (IEEE Std 829-1983).[5] For example, a test

design specification and a test procedure were not specifically developed. The purpose of a test design specification is to identify and summarize the features to be tested. The purpose of a test procedure is to identify the success criteria that the feature must meet. Due to state of the software requirements documentation, as has been discussed in Section 4.1.1, it was difficult to identify the functional requirements that would normally form the basis for the success criteria. To meet the intent of a test design specification and test procedure, sub-features for each vital feature were developed to identify what aspect of the vital feature was to be tested. For each sub-feature, a description/criteria was developed to define how the sub-feature was to be tested. Columns three and four in Table 2 present this information.

With this process, 81 tests were identified. It should be recognized that for any one test, several cases must be tested in order to complete the test. For example, under uncertainty analysis, various sample sizes had to be tested in order to complete the testing of the Monte Carlo and Latin Hypercube sampling techniques used in SAPHIRE.

It was recognized that performing 81 tests would not be possible within the time available for this project; therefore, it was necessary to reduce the original list of vital features. This was accomplished by having frequent SAPHIRE users at the INEL rank each sub-feature as high, medium, or low. Sub-features that were ranked as medium or low were removed from testing. The only major group that was completely removed from testing was the cut set editor group since all sub-features in this group were either ranked as medium or low. Table 3 is the list of SAPHIRE 4.0 vital features that were tested.

Once the features and sub-features to be tested had been identified, a software test record was developed for each test. The software test record specifies the input, execution conditions, and expected results for the test. As each test

Table 2. SAPHIRE 4.0 vital features.

Major function	Vital feature	Sub-features	Description	Test number
Fault Tree Analysis	Cut set generation process	Cut set probability cutoff	Keep only cut sets whose probability is above the cut set probability cutoff.	1
		Event probability cutoff	Keep cut sets where each event in the cut set is above the probability cutoff.	2
		Starting gate	Generate cut sets from the logic below the designated starting gate.	3
		Maximum number of cut sets	The maximum number of cut sets saved in the data is $$i\sum_1^m (1+n_i)<32{,}700$$ where n = number of events in the ith cut set m = number of minimal cut sets in the sequence.	4
		Correct cut set generation	When a fault tree is reduced to cut sets the correct number of cut sets that produce the top event exists.	5
		Size truncation	Truncates the cut sets based on the number of events in the cut set.	6
		Zone truncation	With the zone flag set, cut sets will be truncated based on the number of zones in an event.	7
	Cut set quantification process	Generate minimal cut set upper bound for alternate cut sets of the selected systems	This is the bounding number for the sum of all the cut sets.	8
Fault Tree Analysis	Cut set quantification process	Equation for min-cut upper bound is $$MC=1-i\prod_1^m (1-C_i)$$	• MC = minimal cut set upper bound. • C = probability of the ith cut set. • m = number of minimal cut sets in the sequence.	9
	Analysis of single fault tree	A fault tree created with less than 100 gates and reduced by hand.	This effort is to test the ability of SAPHIRE to solve a single fault tree.	10

Table 2. (continued).

Major function	Vital feature	Sub-features	Description	Test number
Fault Tree Analysis	Analysis of multiple fault trees	Analyze large fault trees (fault tree with greater than 1500 gates). Include complemented events in the fault tree.	Maximum number of gates for a fault is 2500. Fault trees with greater than 1500 gates exercise all fault tree reduction functions.	11
		Analyze medium fault trees (fault tree with greater than 100 and less than 1500 gates). Include complemented events in the fault tree.	The break between medium and large is arbitrary. However, the break between medium and small fault is to allow a possible reduction by hand.	12
		Analyze small fault trees where the total number of gates for each fault tree is less than 100.	Small fault tree reduction should be accomplished by hand if possible.	13
Fault Tree Analysis	Editing and modifying of fault tree cut sets.	Cut set editor - List all systems and allow user to select a system to edit its base cases or alternate cut sets.	Allows the user to • Add a cut set - Add a brand new cut set • View an existing cut set • Modify cut set - Change the events that exist in a cut set. • Delete cut set - Delete the entire cut set. • Restore cut set - Restore the last deleted cut set. • Add event to cut set - Add event to existing cut set. • Modify event in cut set - Modify highlighted event name. • Delete event from cut set - Remove the highlighted event from the cut set. • Restore event - Restore the last deleted event. • Speed search - User can enter a name and the first occurrence of that name is highlighted. The "NEXT" option goes to and highlights the next matching event. The "PREVIOUS" option goes to and highlights the previous matching event.	14

Table 2. (continued).

Major function	Vital feature	Sub-features	Description	Test number
			• View events with mask - User enters a set of attributes as a mask and views any matching events with those attributes. The "NEXT" option looks at the next matching event. The "PREVIOUS" option looks at the previous matching event.	14
			• Find - Finds the cut set(s) that contains a given list of events and performs the following functions: **Insert** - Add the specified event to the cut set **Replace** - Replace a list of events with the specified event. **Delete** - Delete the found cut set. **Copy/Add** - Create a copy of the found cut set and in that copy, replace the list of events with the specified event. **Delete** - Delete the found cut set. **Copy/Add** - Create a copy of the found cut set and in that copy, replace the list of events with the specified event.	
Fault Tree Analysis	Editing and modifying of fault tree cut sets.	Options	• Number of cut sets to look through on find options. • Toggle to ask for modify confirmation or modify without confirmation. • Toggle to ask for delete confirmation or delete without confirmation.	15
Event Tree Analysis	Sequence Process Generation	Generate sequences	• Generate the possible sequences for an event tree or list of event trees. • Generate all logic for each sequence. • All sequences have logic generated except (1) those with "OK," "SUCCESS," or "IGNORE" as end states and (2) those whose end states begin with the character "@". • Add mutually exclusive top (if non-blank top exist) as a successful top to each sequence logic. • Name of list file - Output the logic generated in a report form to the screen, printer, or user named file. • Numbers - Use sequence numbers as the name of each generated sequence. • Names - Sequence name contained in the event tree as the name of each generated.	16

Table 2. (continued).

Major function	Vital feature	Sub-features	Description	Test number

Event Tree Analysis	Sequence Process Generation	Sequence editor - Allow user to change the end state header and fields for a given event tree.	• Frequency - Look up sequence minimum cut set upper bound for marked/highlighted sequences (0.0 is default)
			   - Add frequency to the event tree drawing.
			   - Changed header for the third column to be frequency.
			• '@' status - Change the end state name to be generated (without '@' in the first character) or not generated (with '@' in the first character) for each highlighted/marked sequence.
			• Line edit - Change any highlighted line.
			• Follow transfer - If transfer event tree exists, edit its logic.
			• Global replace - Given the column number and a search string, replace any occurrences of the search string in the given column number with the replace string.
			• Header edit - Change the header for the drawing.
			• Change transfer - Toggle logic between being a transfer path or a termination path.
				17

Event Tree Analysis	Sequence Process Generation	Link Editor - Allows user to enter links (exceptions) that are to be applied during sequence logic generation	• Multiple branches-If more than one failed branch exists and an exception applies to at least one of them, then all branches must have a replaced top and replacement top.
			• Definitions
			   - Rule - A rule of replacement contains a number of conditional tops, tops to be replaced, and tops to replace.
			   - Entry - A cell in a rule (conditional top, replaced top, or replacement top).
			   - Add RULE - Complete rule.
			     ENTRY - Add an entry at the highlighted cell.
			   - Modify RULE - Complete rule.
			     ENTRY - Add an entry at the highlighted cell.
			   - Delete RULE - Complete rule.
			     ENTRY - Highlighted entry.
			   - Restore RULE - Last deleted complete rule.
			     Entry - Last deleted entry.
			   - Copy RULE - Make a copy of an existing rule at user specified location.
				18

Table 2. (continued).

Major function	Vital feature	Sub-features	Description	Test number
	Sequence Cut Set Generation	Cut set probability cutoff	• Discard any cut sets whose probability is less than this value. • Cut set probability of failure is equal to the product of the failure probability of each event.	19

Event Tree Analysis	Sequence Cut Set Generation	Event probability cutoff	• Do not discard cut set if
			  - Cut set's failure probability is greater than the cutoff.
			  - Failure probability for each event in the cut set is greater than or equal to the event probability cutoff.
				20

		Size truncation	• Throw away any cut sets whose size exceeds a specified size.	21
		Zone truncation	• Throw away any cut sets whose number of zone events exceeds the specified zone size.	22
Event Tree Analysis	Sequence Cut Set Generation	Solve sequence with fault trees - Combine the fault tree logic for all referenced successful and failed systems, generate cut sets, and solve using fault tree cut set generation algorithm.	• Maximum of **10,000** events and gates per combined **failed** logic. • Maximum of **10,000** events and gates per combined **successful** logic. • Maximum of **2,500** gates per combined **failed** logic. • Maximum of **2,500** gates per combined **successful** logic.	23
		Solve sequence with system cut sets	• Combine the cut sets for all referenced systems to generate the sequence cut sets. • Referenced system cut sets must exist at this point.	24
		Maximum number of cut sets that can be saved in the database for one sequence is calculated as follows:	• n = number of events in the ith cut set. • m = number of minimal cut sets in the sequence.	25

$$i\sum_{1}^{m} (1+n_i) < 32,700$$

Table 2. (continued).

Major function	Vital feature	Sub-features	Description	Test number
Event Tree Analysis	Sequence Cut Set Generation	Default flag (change Set)	• Each sequence can have a change set that applies only to it during the cut set generation. • The default change set listed here will overwrite any change sets that are specifically tied to the sequence. • Keyword "NONE" will cause cut sets to be generated without any sequence change set.	26

Event Tree Analysis	Sequence Cut Set Generation	Update cut sets	• Reevaluate existing cut sets based on the current data and these parameters:
			  - Cut set probability cutoff
			  - Discard any cut sets whose probability of failure is less than the cutoff value.
			  - The probability of failure is equal to the product of the probability of each event in the cut set.
			  - Size truncation - Throw away any cut sets whose size exceeds a specified size.
			  - Zone truncation - Throw away any cut sets whose number of zone events exceeds the specified zone size.
			• Use base case or alternate case - Use base case cut sets or alternate case cut sets as the basis for the evaluation.
				27

Event Tree Analysis	Sequence Cut Set Generation	Quantify cut sets	• Generate minimal cut set upper bound for the alternate cut sets of the selected sequence or sequences. • Equation for min-cut upper bound is $$MC = 1 - i \prod_{1}^{m} (1 - C_i)$$ where MC = minimal cut set upper bound C = probability of the ith cut set m = number of minimal cut sets in the sequence.	28

Table 2. (continued).

Major function	Vital feature	Sub-features	Description	Test number
		Cut set uncertainty - Perform uncertainty analysis on the sequence(s) based on the following:	• Sampling techniques - Monte Carlo - Latin Hypercube • Sample size • Random number seed • Probability distribution of each event in the cut set • Level - Single - Perform uncertainty on each highlighted/marked sequence individually. - Group - Perform uncertainty analysis on a group of highlighted/marked sequences. - End State - Perform uncertainty analysis on the group of sequences that belong to the same highlighted end state. - Family - Perform uncertainty analysis on all sequences as a group in the family.	29
Event Tree Analysis	Sequence Cut Set Generation	Split fraction	• Calculate the split fraction for the highlighted/marked sequences. • Split fraction is an approximation of the minimum cut set upper bound for a sequence. • The equation is $$SF = (i\prod_{1}^{n} (1 - sMC_i)) \times (j\prod_{1}^{m} fMC_j)$$ where SF = split fraction sMC = minimal cut set upper bound of the ith successful system n = number of successful systems fMC = minimal cut set upper bound of the ith failed system m = number of failed systems	30
		Base case update	• Copy the alternative cut set information to the base case. • Information includes the cut sets, quantified values, and the uncertainty values.	31

Table 2. (continued).

Major function	Vital feature	Sub-features	Description	Test number
	Quantifying Process	• Generate minimal cut set upper bound for the alternate cut sets of the selected sequence or sequences. • Equation for min-cut upper bound is $$MC = 1 - i \prod_{1}^{m} (1 - C_i)$$	• MC = minimal cut set upper bound • C = probability of the ith cut set • m = number of minimal cut sets in the sequence	32
Event Tree Analysis	Capability of performing the analysis on a single event tree	Perform analysis on event tree using fault trees.	• Maximum of **10,000** events and gates per combined **failed** logic. • Maximum of **10,000** events and gates per combined **successful** logic. • Maximum of **2,500** gates per combined **failed** logic. • Maximum of **2,500** gates per combined **successful** logic.	33
		Perform analysis on event tree using cut sets.	• Combine the cut sets for all referenced systems to generate the sequence cut sets. • Referenced system cut sets must exist at this point.	34
	Capability of performing the analysis on multiple event trees.	Perform analysis on two event trees using fault tree method	• Maximum of **10,000** events and gates per combined **failed** logic. • Maximum of **10,000** events and gates per combined **successful** logic. • Maximum of **2,500** gates per combined **failed** logic. • Maximum of **2,500** gates per combined **successful** logic.	35
Event Tree Analysis	Capability of performing the analysis on multiple event trees.	Perform analysis on five event trees using fault tree method	• Maximum of **10,000** events and gates per combined **failed** logic. • Maximum of **10,000** events and gates per combined **successful** logic. • Maximum of **2,500** gates per combined **failed** logic. • Maximum of **2,500** gates per combined **successful** logic.	36

Table 2. (continued).

Major function	Vital feature	Sub-features	Description	Test number
		Perform analysis on fifty event trees using fault tree method	• Maximum of **10,000** events and gates per combined **failed** logic. • Maximum of **10,000** events and gates per combined **successful** logic. • Maximum of **2,500** gates per combined **failed** logic. • Maximum of **2,500** gates per combined **successful** logic.	37
		Perform analysis on two event trees using system cut set method	• Combine the cut sets for all referenced systems to generate the sequence cut sets. • Referenced system cut sets must exist at this point	38
		Perform analysis on five event trees using system cut set method	• Combine the cut sets for all referenced systems to generate the sequence cut sets. • Referenced system cut sets must exist at this point	39
		Perform analysis on fifty event trees using system cut set method	• Combine the cut sets for all referenced systems to generate the sequence cut sets. • Referenced system cut sets must exist at this point	40

Event Tree Analysis	Event tree rules	Multiple branches- If more than one failed branch exists and an exception applies to at least one of them, then all branches must have a replaced top and replacement top.	• Definitions
			  - Rule - A rule of replacement contains a number of conditional tops, tops to be replaced, and tops to replace.
			  - Entry - A cell in a rule (conditional top, replaced top, or replacement top).
			  - Add RULE - Complete rule.
			    ENTRY - Add an entry at the highlighted cell.
			  - Modify RULE - Complete rule.
			    ENTRY - Add an entry at the highlighted cell.
			  - Delete RULE - Complete rule.
			    ENTRY - Highlighted entry.
			  - Restore RULE - Last deleted complete rule.
			    Entry - Last deleted entry.
			  - Copy RULE - Make a copy of an existing rule at user specified location.
				41

Table 2. (continued).

Major function	Vital feature	Sub-features	Description	Test number
Event Tree Analysis	Editing and modifying of event tree sequence cut sets	Cut set editor - list all sequences and allow user to select a sequence to edit its base case or alternate cut sets.	• View existing cut sets. • Add cut set - Add a brand new cut set. • Modify cut set - change the events that exist in a cut set. • Delete cut set - Delete the entire cut set. • Restore cut set - Restore the last deleted cut set. • Add event to cut set - Add event to existing cut set. • Modify event in cut set - modify highlighted event. • Delete event from cut set - Remove the highlighted event from the cut set. • Restore event - Restore the last deleted event. • Speed search - Allow user to enter a name and highlight the first event that matches that name. 　- Next - Highlight the next matching event. 　- Previous - Look at the previous event that matches that name • View events with mask 　- Enter a set of event attributes as a mask and view any matching events along with their attributes. 　- Next - Look at the next event that matches. 　- Previous - Look at the previous event that matches. • Find - Find cut set(s) that contains a given list of events and perform the following functions: 　- Insert - Add the specified event to the found cut set. 　- Replace - Replace the list of events with the specified event. 　- Delete - Delete the found cut set. 　- Copy/Add - Create a copy of the found cut set and in that copy, replace the list of events with the specified event. • Options 　- Number of cut sets to look through on find operations. 　- Toggle to ask for modify confirmation or modify without confirmation. 　- Toggle to ask for delete confirmation or delete without confirmation. • Exit without saving changes. • Exit while saving changes.	42 42

Table 2. (continued).

29

Major function	Vital feature	Sub-features	Description	Test number
Uncertainty Analysis	Single Event sampling - Take single event with known distribution and sample the event to determine if the distribution is reproduced.	Monte Carlo Sampling	• Lognormal Distribution • Normal Distribution • Beta Distribution • Chi-Squared Distribution • Exponential Distribution • Uniform Distribution • Histogram	43
Uncertainty Analysis	Single Event sampling - Take single event with known distribution and sample the event to determine if the distribution is reproduced.	Latin Hypercube Sampling	• Lognormal Distribution • Normal Distribution • Beta Distribution • Chi-Squared Distribution • Exponential Distribution • Uniform Distribution • Histogram	44
Uncertainty Analysis	Uncertainty analysis of sequences.	Monte Carlo Sampling	Perform uncertainty on sequences with approximately 100, 1000, and 5000 cut sets. Check the following levels for the sequences: - Single - Perform uncertainty on each highlighted/marked sequence individually. - Group - Perform uncertainty analysis on a group of highlighted/marked sequences. - End State - Perform uncertainty analysis on the group of sequences that belong to the same highlighted end state. - Family - Perform uncertainty analysis on all sequences as a group in the family.	45

Table 2. (continued).

Major function	Vital feature	Sub-features	Description	Test number
		Latin Hypercube Sampling	Perform uncertainty on sequences with approximately 100, 1000, and 5000 cut sets. Check the following levels for the sequences: - Single - Perform uncertainty on each highlighted/marked sequence individually. - Group - Perform uncertainty analysis on a group of highlighted/marked sequences. - End State - Perform uncertainty analysis on the group of sequences that belong to the same highlighted end state. - Family - Perform uncertainty analysis on all sequences as a group in the family.	46
Change Set Features	Creating Change Sets	Individual Probability Method - Direct assignment of new probability.	This method is most appropriate for when on-of-a-kind events must be changed.	47
Change Set Features	Creating Change Sets	Tagging Method - Manual designation of a group of events that need the same change definition.	This method is appropriate when search criteria for a group of events is difficult to establish or when more than one group of events must reside in a change set.	48
		Class Method -Establishing search criteria for a class of events, inputting the desired change definition, and allowing the software to search the database for all events that meet the search criteria and automatically applying the change definition.	This method is appropriate when a large number of like events must have changes defined.	49
Change Set Features	Applying Changes to Models	Enter the change set menu - In IRRAS it's under "Generate Event Data" menu and in SARA it's under "Modify Event Data" under the Systems Analysis Menu.	• Mark the change set to be modified. • Generate changes.	50

Table 2. (continued).

Major function	Vital feature	Sub-features	Description	Test number
Change Set Features	Applying Changes to Models	Apply changes to system data - Enter the "Analyze Sequence" menu in IRRAS or the "Analyze Systems" menu or the "Analyze Event Tree" menu in SARA.	• Select the quantification option. • Select the system or sequence. • Press enter.	51
	Flags to be applied in change set events	Sensitivity Analysis, "S" flag	Special flag put in for a one-of-a-kind study.	52
		Do not expand transfers, "X" flag	"X" tells SAPHIRE that the basic event is to be used for failure references, but success references are to be treated the same as if the flag was blank.	53
		Never expand transfers, "Y" flag	"Y" indicates that a transfer is to be replaced with its basic event for failed references and the complement of the event is to be used for success references.	54
		Always expand transfers, "I" flag	"I" causes SAPHIRE to treat the transfer as independent. Logic below this transfer is expanded for failure references and for success references, the complement of the logic is used.	55
		Blank flag field	When the process field is blank, the transfer associated with this event is expanded; however, the cut sets generated are removed from the cut sets using cut set matching.	56
		Key event, "K" flag	Fails all locks in the same group.	57
		Lock event, "L" flag	Failed only by its key event.	58
		Zoned flagged event, "Z" flag.	Used to designate that the event has location information attached that can be used as a criteria on for truncation during cut set generation.	59
Cut Set Editor		Add a cut set	Add a brand new cut set View an existing cut set.	60
		Modify cut set	Change the events that exist in a cut set.	61
		Delete cut set	Delete the entire cut set.	62
		Restore cut set	Restore the last deleted cut set.	63

31

Table 2. (continued).

Major function	Vital feature	Sub-features	Description	Test number
Cut Set Editor		Add event to cut set	Add event to existing cut set.	64
		Modify event in cut set	Modify highlighted event name.	65
		Delete event from cut set	Remove the highlighted event form the cut set.	66
		Restore event	Restore the last deleted event.	67
Cut Set Editor	List all sequences and allow user to select a sequence to edit its base case or alternate cut sets.	Speed search	User can enter a name and the first occurrence of that name is highlighted. The "NEXT" option goes to and highlights the next matching event. The "PREVIOUS" option goes to and highlights the previous matching event.	68
		View events with mask	User enters a set of attributes as a mask and views any matching events with those attributes. The "NEXT" option looks at the next matching event. The "PREVIOUS" option looks at the previous matching event.	69
		Find	Finds the cut set(s) that contains a given list of events and performs the following functions: **Insert** - Add the specified event to the cut set **Replace** - Replace a list of events with the specified event. **Delete** - Delete the found cut set. **Copy/Add** - Create a copy of the found cut set and in that copy, replace the list of events with the specified event.	70
Importance Measures	Calculate importance measures with: **P(top)** - Probability that the top event occurs. **P(A)** - Probability of event A (the event of interest.	Fussell-Vessely	$FV = \dfrac{P(top) - P(top/A=0)}{P(top)}$	71
		Risk Reduction Ratio	$RDR = \dfrac{P(top)}{P(top/A=0)}$	72
		Risk Increase Ratio	$RIR = \dfrac{P(top/A=1)}{P(top)}$	73

Table 2. (continued).

Major function	Vital feature	Sub-features	Description	Test number
	P(top/A=1) - Probability that the top event occurs given A always occurs (Probability of A=1). **P(top/A=0** - Probability that top event occurs given event A never occurs (Probability of A=0).	Birnbaum Importance	$BB = P(top/A=1) - P(top/A=0)$	74
Importance Measures	(Same as above)	Risk Reduction Interval	$RRI = P(top) - P(top/A=0)$	75
		Risk Increase Interval	$RII = P(top/A=1) - P(top)$	76
	Select basic events to show their importance. This selection is accomplished through various selection techniques.	Matching basic events with various criteria.	Matching criteria follows: **Primary name** **Group Name** **System** **Train** **Type** **Failure mode** **Location** **Initiating event** **Attributes**	77

Table 2. (continued).

Major function	Vital feature	Sub-features	Description	Test number
		Exclude basic events that match various criteria.	Matching criteria follows: **Primary name** **Group Name** **System** **Train** **Type** **Failure mode** **Location** **Initiating event** **Attributes**	78
Importance Measures	Select basic events to show their importance. This selection is accomplished through various selection techniques.	Complement	Basic events that were not included or marked are now the selected events.	79
		Reset	Make all events selected.	80
		View events	Basic events that are marked ('*') will be selected.	81

was performed, the actual results were also documented on the software test record. The software test record numbers are shown in the sixth column of Table 3. The fifth column of Table 3 provides the cross-reference to Table 2.

Anomaly reports were prepared for all discrepancies identified between the way SAPHIRE should operate and actual operation. Discrepancies were categorized as either critical or non-critical. A discrepancy was categorized as critical if the results were incorrect or the potential existed for the results to be misleading. A discrepancy was categorized as non-critical if the results were correct but the option was difficult to use.

During testing it was recognized that some tests would be redundant in that they would be bounded or covered by other tests. Specifically, the tests that fall into this category are software test records 8, 26, and 28; software test records 14 and 15; and software test record 20.

Analyze a Medium Fault Tree, software test record 8, is bounded by software test record 7, Analyze a Large Fault Tree, and software test record 9, Analyze a Small Fault Tree. Perform Analysis on Two Event Trees Using Fault Tree Method, software test record 26, is bounded by software test record 24, Perform Analysis on Event Tree Using Fault Tree Method, and by software test record 27, Perform Analysis on Five Event Trees Using Fault Tree Method. Perform Analysis on Two Event Trees Using System Cut Set Method, software test record 28, is bounded by software test record 25, Perform Analysis on Event Tree Using Cut Sets, and by software test record 29, Perform Analysis on Five Event Trees Using System Cut Set Method. As a result, Tests 8, 26, and 28 were not explicitly conducted.

Software test records 14 and 15 involve testing if SAPHIRE will generate the correct cut sets for an event tree sequence using the fault tree method in one case (software test record 14) and the cut set method in the second case (software test record 15). These tests are also covered by software test records 24 and 25, respectively. Software test records 24 and 25 involve

testing if SAPHIRE will generate the correct cut sets for all event tree sequences respectively using the fault tree method and the cut set method.

Software test record 20 involves evaluating uncertainty analysis results for event tree sequences. Software test record 20 is covered by software test records 32 and 33.

6.2 Summary of Vital Features Testing

All vital features were tested on SAPHIRE 4.16. The following sections summarize by major group and vital feature the testing that was performed and if the results were as expected or not. The discrepancies that were identified between the expected operation and the actual operation of SAPHIRE are discussed in greater detail in these sections. Software test records are provided in Appendix C. These software test records describe the input files, expected results, and if the output compared to the expected results or not. Software test records with all supporting documentation (actual input and output files) have been retained in the V&V project files. Anomaly reports that were generated for identified discrepancies are provided in Appendix D.

6.2.1 Fault Tree Analysis. The vital features for the major group of Fault Tree Analysis are

1. Cut set generation process

2. Cut set quantification process

3. Cut set quantification process—analysis of single and multiple fault trees.

Each of these features and their associated sub-features were tested. The majority of the results were determined to be acceptable except as discussed below. The implementation for two points should be discussed. First is the message given when the cutoff (either size or probability) is set above any cut set in the fault tree. The message given is, "The top event cannot occur (FALSE)!" A better message might be "No cut sets qualify based upon cutoff." Second, a cut

Table 3. SAPHIRE 4.0 vital features tested.

Major function	Vital feature	Sub-features	Description	Cross-Ref No. to Table 1	Test Record No.
Fault Tree Analysis	Cut set Generation Process	Cut set probability cutoff	Keep only cut sets whose probability is above the cut set probability cutoff.	1	1
		Maximum number of cut sets	The maximum number of cut sets saved in the data is $$i\sum_1^m (1+n_i)<32,700$$ where n = number of events in the ith cut set. m = number of minimal cut sets in the sequence.	4	2
		Correct cut set generation	When a fault tree is reduced to cut sets, the correct number of cut sets exist that produce the top event exists.	5	3
	Cut set quantification process	Generate minimal cut set upper bound for alternate cut sets of the selected systems	This is the bounding number for the sum of all the cut sets.	8	4
Fault Tree Analysis	Cut set quantification process Analysis of single fault tree	Equation for min-cut upper bound is $$MC=1-i\prod_1^m (1-C_i)$$	• MC = minimal cut set upper bound. • C = probability of the ith cut set. • m = number of minimal cut sets in the sequence.	9	5
	Analysis of multiple fault trees	A fault tree created with less than 100 gates and reduced by hand.	This effort is to test the ability of SAPHIRE to solve a single fault tree.	10	6
		Analyze large fault trees (fault tree with greater than 1,500 gates). Include complemented events in the fault tree.	Maximum number of gates for a fault is 2,500. Fault trees with greater than 1,500 gates exercise all fault tree reduction functions.	11	7

Table 3. (continued).

Major function	Vital feature	Sub-features	Description	Cross-Ref No. to Table 1	Test Record No.
Fault Tree Analysis	Analysis of multiple fault trees	Analyze medium fault trees (fault tree with greater than 100 and less than 1,500 gates). Include complemented events in the fault tree.	The break between medium and large is arbitrary. However, the break between medium and small fault three is to allow a possible reduction by hand.	12	8
		Analyze small fault trees where the total number of gates for each fault tree is less than 100.	Small fault tree reduction should be accomplished by hand if possible.	13	9
Event Tree Analysis	Sequence Process Generation	Generate sequences	• Generate the possible sequences for an event tree or list of event trees. • Generate all logic for each sequence • All sequences have logic generated except (1) those with "OK," "SUCCESS," or "IGNORE" as end states and (2) those whose end states begin with the character "@". • Add mutually exclusive top (if non-blank top exist) as a successful top to each sequence logic. • Name of list file - Output the logic generated in a report form to the screen, printer, or user named file. • Numbers - Use sequence numbers as the name of each generated sequence. • Names - sequence name contained in the event tree as the name of each generated	16	10
Event Tree Analysis	Sequence Process Generation	Sequence editor - Allow user to change the end state header and fields for a given event tree.	• Frequency - Look up sequence minimum cut set upper bound for marked/highlighted sequences (0.0 is default) - Add frequency to the event tree drawing. - Changed header for the third column to be frequency.	17	11

Table 3. (continued).

Major function	Vital feature	Sub-features	Description	Cross-Ref No. to Table 1	Test Record No.
			• '@' status - Change the end state name to be generated (without '@' in the first character) or not generated (with '@' in the first character) for each highlighted/marked sequence.		
			• Line edit - Change any highlighted line.		
			• Follow transfer - If transfer event tree exists, edit its logic.		
			• Global replace - Given the column number and a search string, replace any occurrences of the search string in the given column number with the replace string.		
			• Header edit - Change the header for the drawing.		
			• Change transfer - Toggle logic between being a transfer path or a termination path.		
Event Tree Analysis	Sequence Process Generation	Link Editor - Allows user to enter links (exceptions) that are to be applied during sequence logic generation	• Multiple branches - If more than one failed branch exists and an exception applies to at least one of them, then all branches must have a replaced top and replacement top.	18	12
			• Definitions		
			- Rule - A rule of replacement contains a number of conditional tops, tops to be replaced, and tops to replace.		
			- Entry - A cell in a rule (conditional top, replaced top, or replacement top).		
			- Add RULE - Complete rule. ENTRY - Add an entry at the highlighted cell.		
			- Modify RULE - Complete rule. ENTRY - Add an entry at the highlighted cell.		
			- Delete RULE - Complete rule. ENTRY - Highlighted entry.		
			- Restore RULE - Last deleted complete rule. Entry - Last deleted entry.		
			- Copy RULE - Make a copy of an existing rule at user specified location.		
			• Discard any cut sets whose probability is less than this value.		

Table 3. (continued).

Major function	Vital feature	Sub-features	Description	Cross-Ref No. to Table 1	Test Record No.
	Sequence Cut Set Generation	Cut set probability cutoff	• Cut set probability of failure is equal to the product of the failure probability of each event.	19	13
Event Tree Analysis		Solve sequence with fault trees - Combine the fault tree logic for all referenced successful and failed systems, generate cut sets, and solve using fault tree cut set generation algorithm.	• Maximum of **10,000** events and gates per combined **failed** logic. • Maximum of **10,000** events and gates per combined **successful** logic. • Maximum of **2,500** gates per combined **failed** logic. • Maximum of **2,500** gates per combined **successful** logic.	23	14
		Solve sequence with system cut sets	• Combine the cut sets for all referenced systems to generate the sequence cut sets. • Referenced system cut sets must exist at this point.	24	15
		Maximum number of cut sets that can be saved in the database for one sequence is calculated as follows: $$\sum_{1}^{m} i\ (1+n_i) < 32,700$$	• n = number of events in the ith cut set. • m = number of minimal cut sets in the sequence.	25	16
Event Tree Analysis	Sequence Cut Set Generation	Default flag (change Set)	• Each sequence can have a change set that applies only to it during the cut set generation. • The default change set listed here will overwrite any change sets that are specifically tied to the sequence. • Keyword "NONE" will cause cut sets to be generated without any sequence change set. • Reevaluate existing cut sets based on the current data and these parameters: – Cut set probability cutoff – Discard any cut sets whose probability of failure is less than the cutoff value. – The probability of failure is equal to the product of the probability of each event in the cut set.	26	17

Table 3. (continued).

Major function	Vital feature	Sub-features	Description	Cross-Ref No. to Table 1	Test Record No.
Event Tree Analysis	Sequence Cut Set Generation	Update cut sets	• Size truncation - Throw away any cut sets whose size exceeds a specified size. • Zone truncation - Throw away any cut sets whose number of zone events exceeds the specified zone size. • Use base case or alternate case - Use base case cut sets or alternate case cut sets as the basis for the evaluation.	27	18
		Quantify cut sets	• Generate minimal cut set upper bound for the alternate cut sets of the selected sequence or sequences. • Equation for min-cut upper bound is $$MC = 1 - i \prod_1^m (1 - C_i)$$ where MC = minimal cut set upper bound C = probability of the ith cut set m = number of minimal cut sets in the sequence.	28	19

Event Tree Analysis	Sequence Cut Set Generation	Cut set uncertainty - Perform uncertainty analysis on the sequence(s) based on the following:	• Sampling techniques
			  - Monte Carlo
			  - Latin Hypercube
			• Sample size
			• Random number seed
			• Probability distribution of each event in the cut set
			
			• Level
			  - Single - Perform uncertainty on each highlighted/marked sequence individually.
			  - Group - Perform uncertainty analysis on a group of highlighted/marked sequences.
			  - End State - Perform uncertainty analysis on the group of sequences that belong to the same highlighted end state.
			  - Family - Perform uncertainty analysis on all sequences as a group in the family.
				29	20

Table 3. (continued).

Major function	Vital feature	Sub-features	Description	Cross-Ref No. to Table 1	Test Record No.
Event Tree Analysis	Sequence Cut Set Generation	Split fraction	• Calculate the split fraction for the highlighted/marked sequences. • Split fraction is an approximation of the minimum cut set upper bound for a sequence. • The equation is	30	21

$$SF = (i\prod_1^n (1 - sMC_i)) \times (j\prod_1^m fMC_j)$$

where

SF	=	split fraction
sMC	=	minimal cut set upper bound of the ith successful system
n	=	number of successful systems
fMC	=	minimal cut set upper bound of the ith failed system
m	=	number of failed systems.

Major function	Vital feature	Sub-features	Description	Cross-Ref No. to Table 1	Test Record No.
Event Tree Analysis	Sequence Cut Set Generation	Base case update	• Copy the alternative cut set information to the base case. • Information includes the cut sets, quantified values, and the uncertainty values.	31	22
	Quantifying Process	• Generate minimal cut set upper bound for the alternate cut sets of the selected sequence or sequences. • Equation for min-cut upper bound is	• MC = minimal cut set upper bound • C = probability of the ith cut set • m = number of minimal cut sets in the sequence	32	23

$$MC = 1 - i\prod_1^m (1 - C_i)$$

Table 3. (continued).

Major function	Vital feature	Sub-features	Description	Cross-Ref No. to Table 1	Test Record No.
Event Tree Analysis	Capability of performing the analysis on a single event tree	Perform analysis on event tree using fault trees.	• Maximum of **10,000** events and gates per combined **failed** logic. • Maximum of **10,000** events and gates per combined **successful** logic. • Maximum of **2,500** gates per combined **failed** logic. • Maximum of **2,500** gates per combined **successful** logic.	33	24
		Perform analysis on event tree using cut sets.	• Combine the cut sets for all referenced systems to generate the sequence cut sets. • Referenced system cut sets must exist at this point.	34	25
Event Tree Analysis	Capability of performing the analysis on multiple event trees.	Perform analysis on two event trees using fault tree method	• Maximum of **10,000** events and gates per combined **failed** logic. • Maximum of **10,000** events and gates per combined **successful** logic. • Maximum of **2,500** gates per combined **failed** logic. • Maximum of **2,500** gates per combined **successful** logic.	35	26
Event Tree Analysis	Capability of performing the analysis on multiple event trees.	Perform analysis on five event trees using fault tree method	• Maximum of **10,000** events and gates per combined **failed** logic. • Maximum of **10,000** events and gates per combined **successful** logic. • Maximum of **2,500** gates per combined **failed** logic. • Maximum of **2,500** gates per combined **successful** logic.	36	27
	Capability of performing analysis on multiple event trees	Perform analysis on two event trees using system cut set method	• Combine the cut sets for all referenced systems to generate the sequence cut sets. • Referenced system cut sets must exist at this point	38	28
		Perform analysis on five event trees using system cut set method	• Combine the cut sets for all referenced systems to generate the sequence cut sets. • Referenced system cut sets must exist at this point	39	29

Table 3. (continued).

Major function	Vital feature	Sub-features	Description	Cross-Ref No. to Table 1	Test Record No.
Uncertainty Analysis	Single Event sampling - Take single event with known distribution and sample the event to determine if the distribution is reproduced.	Monte Carlo Sampling	• Lognormal Distribution • Normal Distribution • Beta Distribution • Chi-Squared Distribution • Exponential Distribution • Uniform Distribution • Histogram	43	30
		Latin Hypercube Sampling	• Lognormal Distribution • Normal Distribution • Beta Distribution • Chi-Squared Distribution • Exponential Distribution • Uniform Distribution • Histogram	44	31
Uncertainty Analysis	Uncertainty analysis of sequences.	Monte Carlo Sampling	Perform uncertainty on sequences with approximately 100, 1,000, and 5,000 cut sets. Check the following Levels for the sequences: - Single - Perform uncertainty on each highlighted/marked sequence individually. - Group - Perform uncertainty analysis on a group of highlighted/marked sequences. - End State - Perform uncertainty analysis on the group of sequences that belong to the same highlighted end state. - Family - Perform uncertainty analysis on all sequences as a group in the family.	45	32
Uncertainty Analysis	Uncertainty analysis of sequences.	Latin Hypercube Sampling	Perform uncertainty on sequences with approximately 100, 1,000, and 5,000 cut sets. Check the following Levels for the sequences: - Single - Perform uncertainty on each highlighted/marked sequence individually. - Group - Perform uncertainty analysis on a group of highlighted/marked sequences.	46	33

Table 3. (continued).

Major function	Vital feature	Sub-features	Description	Cross-Ref No. to Table 1	Test Record No.
			- End State - Perform uncertainty analysis on the group of sequences that belong to the same highlighted end state. - Family - Perform uncertainty analysis on all sequences as a group in the family.		
Change Set Features	Creating Change Sets	Individual Probability Method - Direct assignment of new probability.	This method is most appropriate for when on-of-a-kind events must be changed.	47	34
Change Set Features	Creating Change Sets	Tagging Method - Manual designation of a group of events that need the same change definition.	This method is appropriate when search criteria for a group of events is difficult to establish or when more than one group of events must reside in a change set.	48	35
		Class Method - Establishing search criteria for a class of events, inputting the desired change definition, and allowing the software to search the database for all events that meet the search criteria and automatically applying the change definition.	This method is appropriate when a large number of like events must have changes defined.	49	36
	Applying Changes to Models	Enter the change set menu - In IRRAS it's under "Generate Event Data" menu and in SARA it's under "Modify Event Data" under the Systems Analysis Menu.	• Mark the change set to be modified. • Generate changes.	50	37

Table 3. (continued).

Major function	Vital feature	Sub-features	Description	Cross-Ref No. to Table 1	Test Record No.
Change Set Features	Applying Changes to Models	Apply changes to system data - Enter the "Analyze Sequence" menu in IRRAS or the "Analyze Systems" menu or the "Analyze Event Tree" menu in SARA.	• Select the quantification option. • Select the system or sequence. • Press enter.	51	38
Change Set Features	Flags to be applied in change set events	Sensitivity Analysis, "S" flag	Special flag put in for a one-of-a-kind study.	52	39
		Do not expand transfers, "X" flag	"X" tells SAPHIRE that the basic event is to be used for failure references, but success references are to be treated the same as if the flag was blank.	53	40
		Never expand transfers, "Y" flag	"Y" indicates that a transfer is to be replaced with its basic event for failed references and the complement of the event is to be used for success references.	54	41
		Always expand transfers, "I" flag	"I" causes SAPHIRE to treat the transfer as independent. Logic below this transfer is expanded for failure references and for success references the complement of the logic is used.	55	42
		Blank flag field	When the process field is blank, the transfer associated with this event is expanded; however, the cut sets generated are removed from the cut sets using cut set matching.	56	43

Table 3. (continued).

Major function	Vital feature	Sub-features	Description	Cross-Ref No. to Table 1	Test Record No.
Importance Measures	Calculate various importance measures with the following definitions: **P(top)** - Probability that the top event occurs. **P(A)** - Probability of event A (the event of interest. **P(top/A=1)** - Probability that the top event occurs given A always occurs (Probability of A=1). **P(top/A=0)** - Probability that the top event occurs given event A never occurs (Probability of A=0).	Fussell-Vessely	$FV = \dfrac{P(top) - P(top/A=0)}{P(top)}$	71	44
		Risk Reduction Ratio	$RDR = \dfrac{P(top)}{P(top/A=0)}$	72	45
		Risk Increase Ratio	$RIR = \dfrac{P(top/A=1)}{P(top)}$	73	46
		Birnbaum Importance	$BB = P(top/A=1) - P(top/A=0)$	74	47
		Risk Reduction Interval	$RRI = P(top) - P(top/A=0)$	75	48
		Risk Increase Interval	$RII = P(top/A=1) - P(top)$	76	49

set report (empty except for the date, time, and page number) is produced from the Report Module even when there are no cut sets available. It might be better to produce the same message that appears under the Display Option, "No cut sets available." Software test records 1 through 9 in Appendix C address the testing performed for the vital features of Fault Tree Analysis.

Cut Set Generation Process—Maximum Number of Cut Sets Saved.

The formula provided in the IRRAS Version 4.0 Functional Requirements Document (Draft, dated December 1992) for the maximum number of cut sets saved was tested (software test record 2). The formula provided is incorrect. It states that the total number of events plus the number of cut sets saved will be one less than 32,700. The correct value was found to be one less than 32,000. The discrepancy appears to be a misinterpretation of the space provided for the storage of cut sets. The impact is considered to be non-critical; however, the user documentation should be changed to reflect the correct storage capacity of the code. The maximum number of cut sets saved was also a problem under Event Tree Analysis (software test record 16).

Cut Set Quantification Process—Large Fault Tree.

A test was performed (software test record 7) to determine if SAPHIRE would correctly quantify a large fault tree (between 1500 and 2500 gates). Even though the quantification and cut set results from SAPHIRE compared well with the results from CAFTA, SAPHIRE and CAFTA reported a different number of gates per tree. CAFTA recognized 1856 gates, while SAPHIRE recognized only 612 gates. By changing the amount of RAM memory, the size of the tree that could be analyzed would change. However, after maximizing memory to 627K out of 640K, it was not possible to analyze a tree past 612 gates without receiving the error message, "Too many gates in this tree!" Therefore, the maximum number of gates (2,500) was never reached and is misleading because the total number of gates is dependent on the amount of available RAM. Since the total number of gates that SAPHIRE calculated for this test was less than 3 times the number of

gates that CAFTA calculated, there is a problem with the gate counter. The impact was categorized as non-critical. The error message does lead the user to believe that the problem may be in a different area other than memory. In addition, SAPHIRE will corrupt the database and recovery actions will be necessary if SAPHIRE is unable to process the tree. The appropriate documentation should be changed.

6.2.2 Event Tree Analysis. The vital features for the major group of Event Tree Analysis are

1. Sequence generation process

2. Sequence cut set generation process

3. Quantifying process

4. Capability of performing analysis on a single event tree

5. Capability of performing analysis on multiple event trees.

Software test records 10 through 29 in Appendix C address the testing performed for the vital features of Event Tree Analysis. The majority of the results were acceptable. The following discrepancies were identified for Event Tree Analysis.

Sequence Generation Process—Link Editor.

The Link Editor sub-feature was tested (software test record 12) to determine if correct sequence logic would be generated when linkage rules were invoked. This sub-feature or option allows the user to define linkage rules and generate sequence logic.

To test the linkage rules, an event tree with multiple branches was developed. The user selects multiple branches to consider more than one possible failure path for an event tree top event. For example, a user can set one safety relief valve to reseat and a separate failure path for failure of more than one safety relief valve to reseat. Figure 2 shows the event tree used for this test. Table 4 reflects the linkage rules that were imposed (the input file). According to the

IE-1	A	B	C	SEQ #

IE-1
Tests Multiple Branches
on a Fault Tree

1
2
3
4
5
6
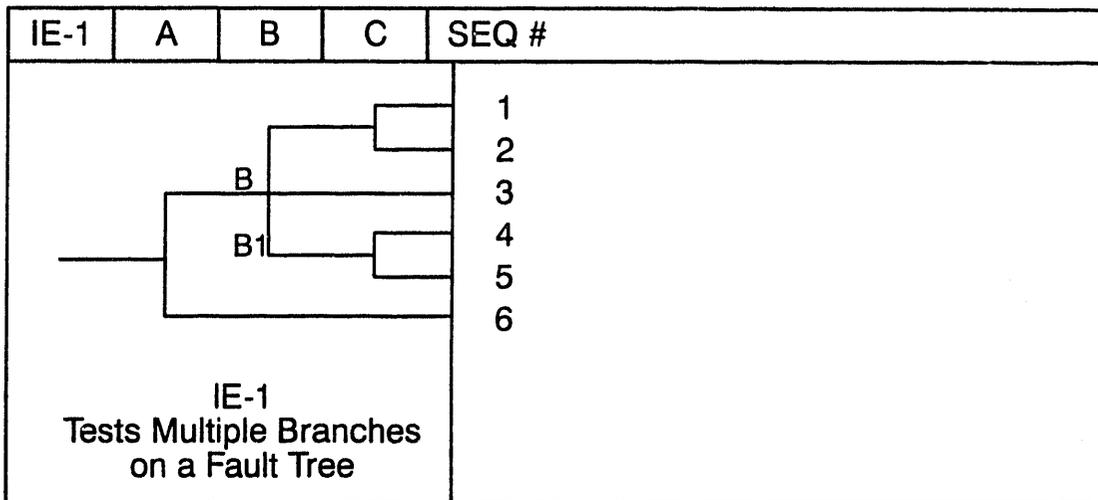
Z048-WHT-1093-01a

Figure 2. Link editor discrepancy.

Table 4. Linkage rules.

IF	AND	THEN	IS
1 /A		B	=B
		B	=B1
2 /A	B1	C	=C1

Table 5. Output file for link editor test.

Sequence name	System	System	System
1	/A	/B	/C
2	/A	/B	C
3	/A	B	
4 substitutes	/A	B B1	/C
5 substitutes	/A	B B1	C
6	A		

rules in Table 4, when A succeeds, B can fail in two ways: either as B or B1. Or, when A succeeds and B1 fails, C should be designated as C1 instead. Table 5 shows the output file after processing by SAPHIRE. All sequences are correct except for Sequence 5. This sequence should have been A succeeds, B fails and is then replaced by B1, and C fails and is then replaced by C1. SAPHIRE did not replace C with C1. Anomaly report AR-12-0893-001 was generated to document this problem.

As a further test, an "@" symbol was placed on the end state of Sequence 3. Therefore, no logic was generated for this sequence. Table 6 shows the output files after processing by SAPHIRE. All sequences are correct except for Sequences 4 and 5. Sequence 4 should have been A succeeds, B fails and is then replaced by

B1, and C succeeds and is then replaced by C1. SAPHIRE did not replace B with B1 and C with C1. Sequence 5 should have been A succeeds, B fails and is then replaced by B1, and C fails and is then replaced by C1. SAPHIRE did not replace B with B1 and C with C1.

From these two tests, SAPHIRE is not correctly processing event tree sequence linkage rules for event trees with multiple branches. Given the potential for some PRA analysts to

Table 6. Output file for link editor test with @ symbol.

Sequence name	System	System	System
1	/A	/B	/C
2	/A	/B	C
4	/A	B	/C
5	/A	B	C
6	A		

develop event trees with more than one possible failure path for an event tree top event, this discrepancy was categorized as critical

Sequence Cut Set Generation Process—Maximum Number of Cut Sets Saved. A separate anomaly report was not generated for the maximum number of cut sets saved under event tree analysis (software test record 16) since the same problems were encountered here as in the fault tree analysis section on the maximum number of cut sets saved (see software test record 2 and anomaly report AR-2-07-93-001).

Sequence Cut Set Generation Process—Split Fraction. From the test of the split fraction calculation (software test record 21), SAPHIRE is not rounding the results correctly. SAPHIRE results may be truncated after three decimal places. The problem appears to be due to incorrect or lack of rounding techniques. This discrepancy was categorized as non-critical; however, it could be an important factor if rounding is treated in the same manner in other portions of SAPHIRE, such as in the quantification modules. (The rounding problem was also noted in software test record 23 on the minimal cut set upper bound calculation.)

Sequence Cut Set Generation Process—Base Case Update. Even though SAPHIRE correctly performed the quantification when the base case update function was tested (software test record 22), the report function printed out the alternate change set values

instead of the base case probabilities. Since the report function does not identify whether the printout is for the base case or the alternate case, the user could inadvertently use the wrong case. This discrepancy was categorized as critical since the user could inadvertently use the wrong case.

6.2.3 Uncertainty Analysis. The vital features for the major group of Uncertainty Analysis are

1. Single event sampling

2. Uncertainty analysis of sequences.

Each of these features and their associated sub-features were tested. Software test records 30 through 33 in Appendix C address the testing performed for the vital features of Uncertainty Analysis. The uncertainty analysis was checked for both the Monte Carlo and Latin Hypercube sampling. For both of these sampling techniques, single event analysis (i.e., a single basic event) and multiple event analysis (i.e., a group of basic events, such as system or sequence cut sets) were investigated. Additionally, the random number generator used in SAPHIRE was checked for its suitability. The following discrepancies were identified for Uncertainty Analysis.

For the single event uncertainty analysis, all SAPHIRE available distributions were checked. These distributions are uniform, exponential, lognormal, chi-square, normal, gamma, beta, and histogram. Out of these eight distributions, only one passed all the acceptance criteria tests (the specific acceptance criteria is spelled out in the particular software test record). The predominant reason why seven of eight distributions failed some of the acceptance criteria tests was that SAPHIRE allows the user to input improper distribution parameters. In some cases, SAPHIRE will print an error message indicating that improper distribution parameters were used, but still attempt to perform the uncertainty analysis using the improper parameters. This could lead to incorrect results, and consequently, was categorized as a critical discrepancy for each appropriate distribution. Anomaly reports AR-30-0893-001 (exponential), AR-30-0893-005

49

(chi-square), AR-30-0893-008 (beta), AR-30-0893-010 (normal), AR-30-0893-011 (gamma), AR-30-0893-012 (lognormal), and AR-30-0893-013 (uniform) discuss the lack of parameter validation for the respective distributions.

For those test cases that did not use improper distribution parameters, another generic problem was found for both the Monte Carlo and Latin Hypercube Sampling. The problem is that SAPHIRE appears to have an internal arithmetic precision limitation when distribution sampled values are small (approximately 1E–14 to 1E–15 or smaller). When a small value is generated by SAPHIRE, the value is rounded down to zero. It is assumed that this problem appears because of the arithmetic operations in the min-cut upper-bound approximation. This discrepancy was categorized as critical for both Monte Carlo and Latin Hypercube Sampling. Anomaly reports AR-30-0893-002 and AR-30-0893-003 give additional detail on this generic problem.

The chi-square distribution has a discrepancy that could result in confusion. When using the chi-square distribution, the SAPHIRE Technical Reference Manual: IRRAS/SARA Version 4.0 (NUREG/CR-5964), states that the chi-square distribution that is sampled (Y) is given by

$$Y = aX \ ,$$

where X is distributed as $\chi^2(k)$, k is the uncertainty value, and a = (probability value)/(uncertainty value). Thus, if the probability value equals 2 and the uncertainty value equals 1, then the sample chi-square distribution results should equal $2\chi^2(1)$.

All the tests that were set up as a multiple of the chi-square distribution did not give sampling results as a multiple of the chi-square distribution. Instead, the results only seemed to give the chi-square distribution part [e.g., the results were very close to what would be expected for $\chi^2(1)$, not $2\chi^2(1)$]. Other than this discrepancy of not resulting in a multiple of the chi-square distribution, the chi-square sampling for n degrees of freedom seemed to be close to tabulated values of chi-square percentiles. Consequently, if the user ignores the probability value and uses only

the uncertainty value as the degrees of freedom for the chi-square distribution, the results should be as expected. This discrepancy was categorized as critical. See anomaly report AR-30-0893-004 for details on the test results.

The gamma distribution had two anomalies, one categorized as non-critical and one as critical. The non-critical anomaly is that when the uncertainty value equals n/2 (where n equals the degrees of freedom) and the probability field equals 2*(uncertainty value), then the gamma distribution will be chi-square distributed with n degrees of freedom. Consequently, it would be expected that the sampling for identical gamma and chi-square distributions should be exactly the same. The Monte Carlo sampling for the gamma distributions, which are $\chi^2(1)$ and $\chi^2(2)$, differs a little from the Monte Carlo sampling for the corresponding chi-square distributions. This inconsistency could cause some confusion. The Latin Hypercube Sampling for the gamma/chi-square distributions does not display this inconsistency. See anomaly report AR-30-0893-007 for details on the test results.

The critical anomaly for the gamma distribution is that with a probability value of 1E–4 and an uncertainty value of 10,000, the Monte Carlo sampling seemed to return the same value (7.06-8E–6) for each sample. Consequently, the sampling distribution results were in error. Also, no error messages were displayed alerting the user to the problem. Two other statistical packages (@RISK and SAS) were used to see if the choice of parameters for the gamma distribution is a generic problem. Both @RISK and SAS correctly handled the gamma distribution with these parameters. Thus, some parameter values for the gamma distribution could result in erroneous results. The users of SAPHIRE should exercise caution and test specific gamma distributions if the gamma distribution must be used. See anomaly report AR-30-0893-006 for details on the test results.

The exponential distribution had a discrepancy that was categorized as non-critical. When using the exponential distribution, the distribution results should be the same as those from the gamma distribution (when the

uncertainty value parameter equals 1 in the gamma distribution). For the Latin Hypercube Sampling analysis, the sampling for EXP(Θ) is not exactly the same as GAMMA(Θ,1), while for the Monte Carlo sampling the two distributions produce exactly the same sampling. This is not a critical error (since both sampling methods for the Latin Hypercube Sampling distributions seem to produce accurate results). However, to be consistent, the EXP(Θ) and GAMMA(Θ,1) sampling for Latin Hypercube Sampling should be the same. (This could also be labeled as another anomaly for the gamma distribution.) See anomaly report AR-31-0893-001 for details on the test results.

The beta distribution has a discrepancy that was categorized as critical. For the beta distribution, the parameters that were specified for several distributions [e.g., (1) prob. value = 0.001, uncertainty value = 0.2; (2) prob. value = 1E–6, uncertainty value = 2; and (3) prob. value = 1E–6, uncertainty value = 800000] should result in legitimate beta distributions. For these beta distributions, SAPHIRE actions ranged from giving runtime errors (causing the need to recover the database) to seemingly running correctly but giving nonsensical results. Thus, some parameter values for the beta distribution could result in erroneous results. The users of SAPHIRE should exercise caution and test specific beta distributions if the beta distribution must be used. See anomaly report AR-30-0893-009 for details on the test results.

No anomalies were found for the multiple event uncertainty analysis (note: only distributions and parameters that checked out correctly from the single event uncertainty analysis tests were used for the multiple event uncertainty analysis tests). Also, tests checking the random number generator did not reveal any anomalies.

6.2.4 Change Set Features. The vital features for the major group of Change Set Features are

1. Creating change sets

2. Applying changes to models

3. Flags to be applied in change set events.

Software test records 34 through 43 in Appendix C address the testing performed for the vital Change Set features. The majority of the results were acceptable. The following discrepancies were identified for Change Set Features.

Creating Change Sets—Class Method.
A test (software test record 36) was performed to evaluate the Class Method sub-feature. This sub-feature or option is used when a large number of events need to be changed. The user establishes a search criteria, and the software will search the database for all events that meet that criteria and automatically apply the change definition. The results obtained from this test were correct; however, an anomaly report (AR-36-0793-001) was generated since the menus and the user documentation do not explain how to mark a class of events. This discrepancy between how the code actually works and the documentation was categorized as non-critical.

Flags to be Applied to Change Set Events. The Sensitivity Process Flag ("S" Flag) was not tested since it is an undocumented feature and has no criteria by which to measure results. This flag allows the analyst to vary the probability of an event or group of events over a specified range and plot the change to core damage frequency. To use this option, the hidden key "P" must be used to gain access to the sensitivity menus. This feature could be very useful and needs to be documented. The discrepancy between how the code actually works and the documentation was categorized as critical. If users do not know about the hidden "P" key, they could mark the flag, run the option, and think that an analysis had been performed by the code when, in actuality, the analysis would not have been performed. Users could then interpret the results as the event or group of events having very little effect on the core damage frequency. Anomaly report AR-39-07-93-002 was generated to document this problem.

The "I" process flag uses the failure of system logic on the event tree failure path and the success of system logic on the event tree success path. This feature was tested under software test

record 42. A change set was created, and sequence logic and cut sets were generated both with and without fault trees. "With fault trees" means that the fault tree logic was used to generate the sequence logic. "Without fault trees" signifies that SAPHIRE used existing fault tree cut sets to generate the sequence logic. Unacceptable results were generated for the without fault trees case. The cause appears to be that SAPHIRE complemented the existing system cut sets incorrectly. SAPHIRE did not account for the event tree success logic in generating sequence cut sets. Anomaly Report AR-42-0793-001 was generated. This discrepancy was categorized as critical since it could provide the user with incorrect results.

6.2.5 Importance Measures. Six importance measures are available in SAPHIRE: Fussell-Vesely, Risk Reduction Ratio, Risk Increase Ratio, Birnbaum Importance, Risk Reduction Interval, and Risk Increase Interval. The importance measures were examined using three slightly different approaches. First, IRRAS calculations for the CSS system in the DEMO family were independently checked using a LOTUS 1-2-3 spreadsheet. Second, importance measures for an event in the LOSP event tree in the DEMO family were compared with importance measures calculated by hand. Last, IRRAS calculations using a SARA-level database of the LaSalle PRA[6] were compared against importance measures published in the LaSalle PRA documentation. All results were determined to be acceptable. Software test records 44 through 49 in Appendix C address the testing performed for importance measures.

7. SOFTWARE TESTING OF NON-VITAL FEATURES

SAPHIRE features were evaluated and either designated as vital or non-vital features. Non-vital features are those features whose malfunction or lack of function will not preclude obtaining the final correct results from SAPHIRE. The non-vital features are fault tree construction, event tree construction, database manipulation, histograms, results display, and general capabilities. Lack of function of a non-vital feature can add many additional hours to an analysis since the work that would have been performed by that feature will have to be performed manually or through an alternate method.

Non-vital features were tested for functionality. The first step was to prepare a test design specification to identify and summarize the non-vital features to be tested. The non-vital features test design specification is provided in Appendix E.

The next step was to prepare a test procedure to identify the success criteria that the feature must meet. The non-vital features test procedure is provided in Appendix F.

The final step was to apply the test procedures to SAPHIRE. Software test records were not generated for the non-vital features. Instead the test procedure was followed and items were checked off as they were completed. All discrepancies identified with the non-vital features were documented on anomaly reports.

The non-vital features testing was performed on SAPHIRE 4.15. Except for one discrepancy, all other discrepancies identified were categorized as non-critical. The one discrepancy categorized as critical was the link editor problem that was also identified during the vital features testing and discussed in Section 6.2.2. The anomaly reports for the non-vital features are provided in Appendix G.

8. CONCLUSIONS AND RECOMMENDATIONS

Section 8.1 discusses the V&V findings that are directly applicable to SAPHIRE and provides recommendations for these findings. Section 8.2 discusses the findings from the V&V process that was used and provides recommendations for improving this process.

8.1 Conclusions and Recommendations from the SAPHIRE V&V

A consistent comment that was heard during the V&V of SAPHIRE/IRRAS was that it is a very powerful and extremely useful PRA code. This comment was received from NRC, commercial industry, and national laboratory users. However, in almost all cases there was the caveat statement that the code is difficult to use and that the user documentation is not helpful. Correction of many of the following findings could improve this situation.

The following summarizes the major findings from each task and, if appropriate, recommends actions that need to be taken.

8.1.1 Software Requirements Documentation. As discussed in Section 4.1.1, the Functional Requirements Document is neither a functional description nor a system requirements specification. It will be desirable to rewrite the Functional Requirements Document so that requirements and design constraints are clearly and concisely stated. It is recommended that this be considered for future SAPHIRE versions. Using the rewritten Functional Requirements Document, a Software Design Description (SDD) should be prepared. Preparing an SDD would greatly facilitate applying changes and enhancements to future versions.

Overall, the methodology employed to control both documentation and software changes needs to be more rigorous. Changes to software should be reflected in system and user documentation as new versions are released and deficiencies corrected. System documentation detailing system

and performance requirements and restrictions should be prepared and maintained so that other code developers could continue developing, enhancing, and maintaining the code if the current code developers were not available.

8.1.2 User Documentation. It is important that the user documentation be complete, correct, and understandable so that the code will be used correctly and to its fullest potential. Evaluation of the user documentation is recommended in IEEE Std 1012-1986. As discussed in Section 5.0, to evaluate the user documentation, users were surveyed across NRC, commercial industry, and national laboratories to determine how well the user documentation meets the needs of the users. Users rated the SAPHIRE documentation for helpfulness, technical content, organization, completeness, and overall usefulness. This survey determined that the SAPHIRE user documentation needs improvement. The majority of users commented that the documentation needs a much better discussion of how the code works, which includes a better discussion of the code hierarchy and of the features and options. Most users also commented that the documentation needs to discuss how to use the code to perform a PRA. Several users suggested that the user documentation include a road map or flow path of how to use SAPHIRE from a PRA analyst's point of view.

8.1.3 SAPHIRE V&V Testing and Code Discrepancies. As discussed in Section 6.1, not all vital features were tested due to the time available to V&V SAPHIRE 4.0. To reduce the original list of vital features, frequent INEL users were asked to rank each sub-feature as high, medium, or low. Only those sub-features that were ranked as high were tested. For the V&V of SAPHIRE 5.0, it is planned that all vital features will be listed again, including any new vital features that result from changes or enhancements to SAPHIRE 5.0. It is expected that all vital features will be tested for SAPHIRE 5.0.

Discrepancies between the way SAPHIRE should operate and actually operates were

identified. All discrepancies identified were either considered to be of moderate or minor significance. Section 6.2 presented discrepancies that were identified during the V&V testing of vital features. The discrepancies were provided to the code developers for modifications to SAPHIRE 4.16. Briefly, the following discrepancies were categorized as critical:

Link Editor. SAPHIRE is not correctly processing event tree sequence linkage rules for event trees with multiple branches (see Section 6.2.2). This discrepancy cannot be corrected in the SAPHIRE 4.0 series.

Sequence Cut Set Generation Process —Base Case Update. Even though SAPHIRE correctly performed the quantification when the base case update function was tested, the report function did not identify whether the printout was for the base case or the alternate case (see Section 6.2.2). This discrepancy was categorized as critical since the user could inadvertently use the wrong case. This discrepancy will be corrected with the release of SAPHIRE 4.17.

Uncertainty Analysis. Several discrepancies were identified with the uncertainty analysis that are considered to be of moderate significance (see Section 6.2.3). The first is that SAPHIRE will allow the user to input improper distribution parameters. In some cases SAPHIRE will print an error message indicating that improper distribution parameters were used, but it will still attempt to perform the uncertainty analysis.

Another moderate significance discrepancy was the generic problem for both Monte Carlo and Latin Hypercube Sampling. It appears that SAPHIRE has an internal arithmetic precision limitation when distribution sampled values are small.

All the tests that had a multiple of the chi-square distribution did not give sampling results as a multiple of the chi-square distribution. Instead, the results seemed only to give the chi-square distribution part [e.g., the results were

very close to what would be expected for $\chi^2(1)$, not $2\chi^2(1)$].

The anomaly for the gamma distribution is that with probability value of 1E–4 and uncertainty value of 10,000, the Monte Carlo sampling seemed to return the same value (7.068E–6) for each sample. Consequently, the sampling distribution results were in error. Also, no error messages were displayed alerting the user to the problem.

For certain beta distributions, SAPHIRE actions ranged from giving runtime errors (causing the need to recover the database) to running seemingly correctly but giving nonsensical results.

All discrepancies associated with uncertainty analysis will be corrected with the release of SAPHIRE 4.17.

Process Flags. The Sensitivity Process Flag ("S" Flag) was not tested since it is an undocumented feature and has no criteria by which to measure results. This flag allows the analyst to vary the probability of an event or group of events over a specified range and plot the change to core damage frequency. Even though this is a menu option, it will not operate by simply selecting the option from the menu. An anomaly report was generated because the potential exists for a user to select this menu option and think that an analysis has been performed. To use this option, a hidden key must be used to gain access to the necessary menus and "run" the analysis (see Section 6.2.4). This feature could be very useful and needs to be documented.

For the "I" process flag, unacceptable results were generated for the "without fault trees" case because SAPHIRE complemented the existing system cut sets incorrectly. SAPHIRE did not take into account the event tree success logic in generating sequence cut sets. This discrepancy was categorized as critical since it could provide the user with incorrect results (see Section 6.2.4). All discrepancies associated with process flags will be corrected as the options will be removed with the release of SAPHIRE 4.17.

8.2 Conclusions from the V&V Process

The lessons learned during the V&V are discussed below along with any changes that should be incorporated into the V&V process for SAPHIRE 5.0.

The V&V Plan provided a definite structure to the V&V effort and ensured that all testing had a purpose and was documented. However, in one case the V&V plan wasn't followed to the letter, and this deviation was not documented. Initially, a SAPHIRE V&V committee of senior PRA analysts was formed according to the V&V Plan to supplement the experience and knowledge of the personnel performing the V&V. However, for the vital features testing, it was decided to use senior PRA analysts, thus negating the need for the V&V committee. The SAPHIRE 5.0 V&V Plan should allow for the V&V committee to be optional, based on the experience and knowledge of the personnel performing the V&V.

Reproducing the results is very important in any technical analysis, including a V&V. To ensure that results can be reproduced, the code version must remain the same throughout the V&V. Every effort should be made to use the same code version during the V&V of SAPHIRE 5.0.

The description/criteria column of the vital features list did not always provide well-defined criteria and test conditions for all features and sub-features. Some tests were also bounded by other tests or were imbedded within other tests. To eliminate any redundancies and inconsistencies for the V&V of SAPHIRE 5.0, the list of vital features tested should be revised to ensure that the best test of a feature has been determined and is properly described.

9. REFERENCES

1. The Institute of Electrical and Electronic Engineers, "IEEE Standard for Software Verification and Validation Plans," IEEE Std 1012-1986.

2. M. H. Halstead, *Elements of Software Science*, Amsterdam: North-Holland, 1977.

3. R. J. McCabe, *A Complexity Measure*, IEEE Transaction Software Engineering, SE-2, 308-20, 1976.

4. R. S. Pressman, *Software Engineering - A Practioner's Approach*, 3rd ed., New York: McGraw-Hill, 1990.

5. The Institute of Electrical and Electronic Engineers, "IEEE Standard for Software Test Documentation," ANSI/IEEE Std 829-1983.

6. A. C. Payne et al., *Analysis of the LaSalle Unit 2 Nuclear Power Plant: Risk Methods Integration and Evaluation Program (RMIEP), Internal Events Accident Sequence Quantification*, NUREG/CR-4832, SAND92-0537, August 1992.

Appendix A

Verification and Validation
Plan for SAPHIRE 4.0

Appendix A

Verification and Validation
Plan for SAPHIRE 4.0

This document defines the software verification and validation (V&V) plan to be used for the System Analysis Programs for Hands-on Integrated Reliability Evaluation (SAPHIRE) 4.0 code suite. This V&V plan is based on ANSI/IEEE Std 1012-1986. However, not all the sections outlined in this standard for a life-cycle V&V will be applied because SAPHIRE 4.0 is an existing code. All the sections that will not be applied will be so indicated.

SAPHIRE is a set of four computer programs that the NRC has developed for the performance of probabilistic risk assessments (PRAs). These programs allow an analyst to perform many of the functions necessary to create, quantify, and evaluate the risk associated with a facility or process being analyzed. The programs included in this set are Integrated Reliability and Risk Analysis System (IRRAS), System Analysis and Risk Assessment (SARA), Models and Results Database (MAR-D), and Fault Tree/Event Tree/Piping and Instrumentation Diagram (FEP) graphical editor. These programs are "hands-on" in the sense that they execute on an IBM-compatible PC, and they are integrated in the sense that they can communicate with one another through the MAR-D database.

To ensure an adequate review and completeness of tests, a team of expert users and code developers has been assembled. Henceforth, this assemblage will be known as the SAPHIRE V&V committee. The purpose of the SAPHIRE V&V committee is to ensure that this V&V Plan and the resulting test plans and test procedures are adequate to test the specified SAPHIRE capabilities, that the procedures are uniform and will actually verify the items being checked, and that data from the tests meet the acceptance criteria. This team is necessary to supplement the experience and knowledge of the performers of the V&V.

A-1. PURPOSE

A documented V&V of SAPHIRE 4.0 is necessary to demonstrate that the software is adequate for expected production applications. Although SAPHIRE 4.0 is an existing code and it has been extensively Beta tested, no formal V&V presently exists. This V&V plan is proposed for the purpose of completing a documented V&V effort that shows that SAPHIRE 4.0 is adequate to perform safety related frequency and risk calculations.

A-2. REFERENCED DOCUMENTS

This V&V plan shall comply with the following documents:

1. ANSI/IEEE Std 1012-1986, "IEEE Standard for Software Verification and Validation Plans"

2. ANSI/ASME NQA-1 Supplement 11S-2, "Supplementary Requirements for Computer Program Testing"

Documents referenced by this V&V plan are:

1. ANSI/ANS-10.4-1987, "Guidelines for the Verification and Validation of Scientific and Engineering Computer Programs for the Nuclear Industry"

2. ANSI/IEEE Std 829-1983, "Test Documentation"

Supporting documents required to supplement or implement this V&V plan are:

1. The software requirements specification for SAPHIRE 4.0.

2.   The system environment specification of the SAPHIRE 4.0 version. DOS versions 3.31 to 5.0 are compatible with SAPHIRE 4.0; any of these DOS versions are acceptable. Hardware requirements are an 8088-based through 80486-based or 100% compatible machine with 640K main memory. SAPHIRE 4.0 was written on a Stoneybrook compiler dated September 30, 1990, and Sage development environment version 3.1 dated November 30, 1990.

3.   The SAPHIRE 4.0 users manual.

A-3. DEFINITIONS

In the context of this V&V plan, the following definitions apply.

Anomaly. Anything observed in the documentation or operation of software that deviates from expectations based on previously verified software products or reference documents. A critical anomaly is a failed vital feature and must be resolved before the V&V effort can proceed.

Beta testing. Testing and notification of anomalies provided by software users.

Vital feature. A vital feature is a feature that is essential for completing the PRA analysis. If this feature failed, it would be impossible to get final PRA results from the SAPHIRE 4.0 PRA workstation.

Life-cycle phase. Any period of time during software development or operation that may be characterized by a primary type of activity (such as design and testing) that is being conducted.

Non-vital feature. Non-vital features are those features whose malfunction or lack of function will not preclude getting the final results from the SAPHIRE 4.0 workstation. Lack of function of a non-vital feature can add many additional hours, days, or months to an analysis since the work that has been done by that feature will have to be done by hand.

Software requirements specification. Documentation of the essential requirements (functions, performance, design constraints, and attributes) of the software and its external interfaces.

Test case specification. Documentation specifying inputs, predicted results, testing procedures, and a set of execution conditions for a test item.

Test design specification. Documentation specifying the details of the test approach for a software feature or combination of software features and identifying the associated tests.

Validation. The process of evaluating software at the *end* of the software development process to ensure compliance with software requirements obtained from the software requirements specification.

Verification. The process of determining whether or not the products of a *given phase of the software development cycle* fulfill the requirements established *during the previous phase.*

A-4. VERIFICATION AND VALIDATION OVERVIEW

A-4.1 Organization

To be completed by the principal investigator (PI).

A-4.2 Master Schedule

To be completed by the PI.

A-4.3 Resources Summary

The resources, such as staffing, facilities, tools, finances, and special procedural requirements, shall be defined by the PI. Emphasis will be placed on the most important software elements with the scope of the V&V being balanced with cost and schedule constraints.

A-4.4 Responsibilities

The responsibilities of the PI and the SAPHIRE V&V committee are defined in this section. The PI has responsibility for specifying and meeting budget, schedule and V&V plan requirements. The PI will direct and oversee performance of the V&V. The SAPHIRE V&V committee is responsible for the following items:

- Review and approval of the V&V plan

- Approval of features to be tested

- Review, approval, and recommendation of actions for anomaly reports

- Review and approval of the final V&V report.

A-4.5 Tools, Techniques, and Methodologies

Test cases will be developed to test major features, and the testing will focus on expected usage as predicted by the V&V performers. Where possible, test results will be compared with theoretically-based hand calculations. Where hand calculations are not practical, existing codes such as CAFTA and PC-SETS software will be used to provide computer-assisted results for comparison. Although neither CAFTA nor PC-SETS are V&Ved, it is deemed appropriate that correlation of results with SAPHIRE signifies correct operation of the software since the three codes use different coding techniques and methodologies.

A-5. LIFE-CYCLE VERIFICATION AND VALIDATION

A-5.1 Management of V&V

1. *Software Verification and Validation Plan Generation.* This document will serve as the software V&V plan, and can be revised by the PI and V&V committee as necessary.

2. *Baseline Change Assessment.* All V&V tests must be performed on one version of the code. The current V&V plan does not accommodate an iterative cycle of producing anomaly reports and updating the code. However, if a critical anomaly is found, the SAPHIRE V&V committee can recommend modifying the code and performing V&V on the new version. In this circumstance, the impact on the V&V schedule and budget would have to be assessed. Following the code modification, tests may have to be rerun.

3. *Management Review.* The PI will provide V&V progress reports, test design specification, test case specifications, anomaly reports, test summary reports, and the final V&V report to the SAPHIRE V&V committee.

    The SAPHIRE V&V committee will review these items and provide a written summary of actions to all committee members and the PI.

4. *Review Support.* The SAPHIRE V&V committee (or member(s) appointed by the committee) will review and evaluate V&V progress reports, test design specification, test case specifications, anomaly reports, test summary reports, and the final V&V report. Other reviews may be requested by the PI such

as independent reviews or external reviews. The PI should conduct reviews as required by the Quality Program Plan that the project is conducted under. Documentation of reviews will include as a minimum an internal correspondence summarizing what was reviewed, findings, and who performed the review.

5. *Applicable Life-cycle Phases*. Because SAPHIRE 4.0 is an existing code, not al' life-cycle phases of V&V stipulated by IEEE Std 1012-1986 are applicable to this plan. Therefore, the following sections of the IEEE Std 1012-1986 report format are *not* applicable to this plan:

A-5.2 Concept Phase V&V

Not applicable.

A-5.3 Requirements Phase V&V

Not applicable.

A-5.4 Design Phase V&V

Not applicable.

A-5.5 Implementation Phase V&V

Not applicable.

A-5.6 Test Phase V&V

Not applicable.

A-5.7 Installation and Checkout Phase V&V

Not applicable.

A-5.8 Operation and Maintenance Phase V&V

5.8.1 V&V Tasks

The following V&V tasks will be performed on SAPHIRE 4.0:

1. Evaluate completeness and correctness of the software requirements specification. Generate an anomaly report for any discrepancies found. Summarize completion of this task in an internal correspondence to the SAPHIRE V&V committee.

2. Throughout the V&V, evaluate completeness and correctness of user documentation, particularly with regard to preventing code misuse. Generate an anomaly report for any discrepancies found. (See item 4.)

3. Design and perform software tests. Software features that are designated as vital features are to be tested using the requirements stated in Section 5.X.4. Software features designated as non-vital features are to be tested for functionality. The functionality tests will be recorded as a checklist and any anomalies will be reported. Any untested features will be listed. The SAPHIRE V&V committee should approve the designation of vital, non-vital, and untested features.

4. Tasks that are not planned include the following:

(a) Comprehensive inspection of source code; however, source code listings should be available for inspection.

(b) Comprehensive review of user documentation; however, anomalies detected should be reported.

5.8.2 Software Features to be Tested

The following vital features will be tested:

- Fault Tree Analysis: Test cases will be designed to test the fault tree cut set generation process, the quantifying process, and the capability to perform the analysis on a single fault tree or on multiple fault trees. The editing and modifying of fault tree cut sets are also tested in this process.

- Event Tree analysis: Test cases will be designed to test the event tree sequence generation process, sequence cut set generation process, the quantifying process, and the capability of performing the analysis on a single event tree or on multiple trees. The event tree rules and editing and modifying of event tree sequence cut sets are also tested in this process.

- Uncertainly Analysis: Test cases will be designed to test the uncertainty analysis for the whole "family" (SAPHIRE designation for a entire PRA model) using both the Latin Hypercube and the Monte Carlo simulation processes. The effect of sample size and seed number on the results will also be tested.

- Change Sets Feature: Test cases will be designed to test the change sets feature and similar features used to perform sensitivity analyses.

- Cut set editor: Test cases will be designed to test the cut set editor.

Non-vital features to be tested are as follows:

- Fault Tree construction: Tests will be performed to test fault tree construction, including logic symbols and the functions in the Build Fault tree menu (e.g., alpha to graphics, pager, text fonts, scaling, and printing capability).

- Event Tree construction: Tests will be performed to test the event tree construction, including logic, sequences, plant damage states, and the functions in the Create Event tree menu (e.g., alpha to graphics, add/delete branch, add/delete top events, text fonts, scaling, and printing capability).

- Data Base Manipulation: The adding and deleting of basic events, event trees or fault trees, the base case update feature, and the change sets update feature will be tested.

- Histogram: The percentage and range format of the histograms will be tested.

- Results Display: The result displays and reports on screen and on hard copy will be tested.

- General Capabilities: The general capabilities of SAPHIRE, such as the on-line help features, the error messages, and the locate function (F5), will be tested.

5.8.3 Procedures for V&V

The V&V performers under the direction of the PI are expected to follow the basic outline below:

1. Evaluate the software requirements specification and add or clarify requirements if necessary to complete the V&V.

2. Develop test design specifications following the requirements of Section 5.8.4. For tasks that require a deviation from the V&V plan, create a deviation report. Develop and document test case specifications that define input, predicted results, testing procedures, and execution conditions.

3. Circulate the test design and test case specifications for the SAPHIRE V&V committee review and approval.

4. Execute the tests.

5. Report anomalies and progress.

6. Receive review of anomaly reports from the SAPHIRE V&V committee.

7. Complete documentation of V&V and retain records as required in Section 7.4.

5.8.4 Testing Requirements

Testing objectives will be documented in a test design specification report for each group of features to be tested. The test design specification describes the testing approach. The report will contain tables that document the coverage of test cases with regard to the requirements and features being tested. Test case specifications will be created that specify input, predicted results, testing procedures, and execution conditions. Documentation of predicted results will include a description of how the results were generated and appropriate references.

For each test case (or test segment) a software test record (STR) form will be completed and the following information will be provided:

- STR number for tracking
- test identification
- computer system and operating environment
- description of segment being tested
- input data files
- output data files
- batch or operator entries
- output listings
- date of test
- test performer.

A test summary report will be created for each test case that includes a summary of results and variances, summary of activities, comment on adequacy and coverage of the test and appropriate approvals.

A numbering system for the test cases, STRs, and anomaly reports, and a filing system for testing materials will be established. Configuration control of test cases, input and output files, and all project materials is required.

Subsequent revisions to SAPHIRE can use the test design specifications generated in this V&V effort to test features that have not changed. Such usage results in a cost saving on subsequent SAPHIRE V&Vs.

A-6. SOFTWARE VERIFICATION AND VALIDATION REPORTING

The documents to be created during the V&V effort are described below. Their distribution for review is described in Section 5.1.

1. *V&V Progress Reports.* EG&G, Idaho, Inc., will prepare a monthly letter status report, including recovery data, in accordance with the current NRC Manual Chapter 1102.

2. *Anomaly Report.* Anomaly reports shall be generated each time an anomaly is detected by the V&V effort. Each report shall include the control number for tracking purposes, description and location of the anomaly, its impact and cause (if known), its criticality (as described in Section 7.1 of the V&V plan), recommendations, and resolution decision. Following review by the SAPHIRE V&V committee, each anomaly report should be signed by the PI and SAPHIRE V&V committee representative.

3. *Test Design Specification.* The test design specifications will be documented in a format that is suitable for review and should be included in the final V&V report. Contents of the test design specification are described in Section 5.8.4.

4. *Test Case Specification.* The test case specification documentation specifying inputs, predicted results, testing procedures, and a set of execution conditions for a test item will be documented in a format suitable for review and use.

5. *Test Summary Reports.* Test summary reports should be provided in internal correspondence format. Contents of the test summary reports are described in Section 5.8.4

6. *V&V Final Report.* A V&V final report using the NUREG/CR format will be generated. This report will summarize all life-cycle V&V tasks performed, all task results, test design specification reports, all anomalies identified and resolutions, assessment of overall software quality, and appropriate recommendations.

A-7. VERIFICATION AND VALIDATION ADMINISTRATIVE PROCEDURES

A-7.1 Anomaly Reporting and Resolution

Anomaly reports will be created by the V&V performers, and shall be submitted to the SAPHIRE V&V committee. Anomalies shall be ranked into one of three categories to assess the significance from a user's viewpoint as to the ability to produce accurate results:

1. Critical

2. Moderate significance

3. Minor significance.

The significance category of the anomaly and correctness of the assessment of the anomaly shall be reviewed by the SAPHIRE V&V committee. Acceptance of the stated anomaly by the SAPHIRE V&V committee shall be shown by a committee representative signature on the anomaly report. Recommendations for resolution can be made by the SAPHIRE V&V committee.

A-7.2 Task Iteration Policy

An anomaly that is judged by the SAPHIRE V&V committee to be critical may require stoppage of the V&V to correct the code. (See Section 5.1) However, the current V&V plan does not accommodate an iterative process, and the V&V plan would have to be modified.

A-7.3 Deviation Policy

Deviations from the finalized V&V plan must be documented on a deviation form. Deviations shall be reviewed by the SAPHIRE V&V committee.

A-7.4 Control Procedures

All relevant materials (listings and files on magnetic media) including codes, reports, test plans, test case listings, and test input/output shall be retrievable and organized by a documented filing system. The records should be kept to aid future V&V needs as well as to provide quality records for the present project.

A-7.5 Standards, Practices, and Conventions

An appropriate Quality Program Plan (QPP) shall be created as designated by the PI.

SOFTWARE TEST RECORD

STR No.   :_____

Date   :_____

Originator   :_____

Test Number: _____

Test Description: _____

_____

_____

_____

_____

Computer system description (attached): _____

_____

_____

_____

_____

Batch job or operator commands (attached): _____

_____

_____

_____

_____

Expected values/acceptance criteria (attached): _____

_____

_____

_____

_____

Input data file names and listings (attached): _____

_____

_____

_____

_____

Output data file names and listings (attached): _____

_____

_____

_____

_____

Appendix A

ANOMALY REPORT

Anomaly Report No.   :_____
Test Record No.   :_____
Date   :_____
Originator   :_____

Source/Location:

_____

_____

Description:

_____

_____

Cause:

_____

_____

Impact:

_____

_____

Criticality Scale: 1   2   3
                         Critical   Moderate   Minor
                                        Significance   Significance
Recommendations:

_____

_____

_____

Requested Resolution Date: _____

Resolution:

_____

_____

_____

_____

_____

Principal Investigator   _____   _____
                                  Signature   Date

Committee Representative   _____   _____

DEVIATION REPORT

Deviation Report No.   :_____
Test Record No.   :_____
Date   :_____
Originator   :_____

Source/Location:

_____

_____

_____

Description:

_____

_____

_____

Justification:

_____

_____

_____

Anticipated Impact on V&V

_____

_____

_____

Principal Investigator _____   _____
                            Signature   Date

Committee Representative _____   _____
                            Signature   Date

Appendix B

Functional Requirements Document Review

CONTENTS

Appendix B

Functional Requirements Document Review

B-1. INTRODUCTION

As part of the verification and validation activities (V&V) performed on the Systems Analysis Program for Hands-on Integrated Reliability Evaluation (SAPHIRE), in particular the Integrated Reliability and Risk Analysis System (IRRAS), the documentation describing the functionality of and requirements for IRRAS were to be reviewed.

Only one document, other than user manuals, describing the functionality of IRRAS was delivered to the V&V team. This document, a draft of the "Integrated Reliability and Risk Analysis System (IRRAS) Version 4.0 Functional Requirements Document," was completed in December 1991 by S. Ted Wood and Kenneth D. Russell, two key people in the development of IRRAS.

B-2. PURPOSE AND SCOPE

A review of the document was conducted as a part of an independent verification and validation (IV&V) project conducted on IRRAS. The review of the Functional Requirements Document was conducted to determine correctness, completeness, and traceability of the requirements and to determine how useful the document will be in developing test cases and test procedures for use in executing tests on the IRRAS code. As the Functional Requirements Document for IRRAS was examined, the reviewer noted observations concerning the document that impact the ability to perform tests on the IRRAS as well as the completeness and correctness of the requirements, where possible.

The purpose of this document is to present the findings of the review and to provide reviewer observations of the content of the Functional Requirements Document. This document contains a summary of those observations as applied to the entire Functional Requirements Document. This document also contains suggestions and recommendations based on the observations made.

No attempt is made to specifically identify functional tests to be executed on the IRRAS. Additional document reviews must be conducted before providing any testing suggestions based on the Functional Requirements Document.

B-3. ACRONYMS

FD	Functional Description
INEL	Idaho National Engineering Laboratory
IRRAS	Integrated Risk and Reliability Analysis System
IV&V	Independent Verification and Validation
PRA	Probability Risk Assessment
SAPHIRE	System Analysis Program for Hands-on Integrated Reliability Evaluation
SDD	Software Design Document
SRS	Software Requirements Specification

B-4. METHODOLOGY

As the document was reviewed to identify functionalities as well as specific items for testing, the document was examined to determine the testability of the requirements. The requirements presented were also examined for completeness, correctness, and traceability.

NUREG/CR-6145

Appendix B

The original intent was to review each subsection in the same manner as the Functional Requirements document structure and provide observations and comments about each subsection. However, upon completion of the Fault Tree Graphical Editor, it was determined that such a detailed report was not necessary. The same observations, or the same comments are valid for the entire section. Therefore, from that point on, observations are presented that refer to an entire section with subsections specifically addressed only when clarification is achieved or required.

B-5. OBSERVATIONS

This section presents the observations of the reviewer as the Functional Requirements document was examined. The observations are grouped to correspond with the paragraph numbering system employed in the Functional Requirements. A summary of observations is presented below.

B-5.1 General

This section contains requirements that apparently are to be applied throughout IRRAS. Information regarding the scope of this section is missing.

B-5.1.1 General Capabilities

Section A describes the general capabilities to be present in IRRAS. The section describes help features, locate functions, marking functions, escape sequences, hot keys, error messages, and warning messages.

The section is written as a description of changes to be implemented on an earlier version of IRRAS without identifying that version or which items constitute changes or enhancements.

B-5.1.1.1 On-Line Help. On-line help is required on any form showing <F1> as an option. No specific forms are identified.

B-5.1.1.2 Field Specific Help. Field specific help is required in *designated places*. The designated places are not identified.

B-5.1.1.3 Locate Function. A locate function is required on any form that displays a list of names and shows <F5> as an option. Specific forms are not identified.

B-5.1.1.4 Marking Functions. Marking functions are required on any form displaying a list of names and showing the function keys <F2>, <F3>, and <F4>. No specific forms are identified.

B-5.1.1.5 Error Messages. Error messages are to be provided that inform the user of incorrect data, invalid option selection, of failure of operation. No format or other specifics are provided.

B-5.1.2 Data base

This section of the Functional Requirements addresses the data base associated with IRRAS.

This section apparently describes the information to be displayed when the user selects options displayed on menus relating to families or plants, event trees, systems, end states, or basic events.

The requirements are not clearly stated. Apparently the requirements have been written as modifications and enhancements to a previous version of IRRAS. No previous version is identified nor referenced in this section.

It is also apparent that very specific requirements are contained and/or referenced in this section. However, they are not clearly stated as such.

B-5.2 Disclaimer Notice

This section describes the contents of a disclaimer that must be displayed when IRRAS is initially invoked. Additional specifics could be provided, but probably would not enhance this requirement.

B-5.3 Select Family

Apparently the requirements have been written as modifications and enhancements to a previous version of IRRAS. Version 2.5 data apparently must be upgraded before it can be used with Version 4.0.

This section clearly contains numerous requirements for IRRAS. However, they are not stated in a clear and concise manner. Several subsections are worded in such a way as to imply a requirement without explicitly stating the requirement.

Numerous design issues are presented in this section.

B-5.4 Generate Current Event Data

B-5.4.1 Definitions

This section contains information indicating that all defined fields must have field specific help available.

The section presents definitions that are field descriptions. However, it is not readily apparent if the user must supply all of the data or if the system will process existing data or use data just entered to provide the data required in other fields. No description of processing activities is provided.

This section clearly contains numerous requirements for IRRAS. However, they are not stated in a clear and concise manner. Many are implied.

Numerous design issues are presented in this section.

B-5.4.2 List Existing Change Sets

This section contains no information, only a section heading.

B-5.4.3 Add Change Set

This section clearly contains numerous requirements for IRRAS. However, they are not stated in a clear and concise manner. Many are implied.

Design issues are presented in this section.

B-5.4.4 Modify Change Set

This section clearly contains numerous requirements for IRRAS. However, they are not stated in a clear and concise manner. Many are implied.

Design issues are presented in this section.

B-5.4.5 Delete Change Set

The requirement, as stated, appear to be understandable. However, design issues are presented.

B-5.4.6 Probability

This section clearly contains numerous requirements for IRRAS. However, they are not stated in a clear and concise manner. Many are implied.

Design issues are presented in this section.

B-5.4.7 Class

This section clearly contains numerous requirements for IRRAS. However, they are not stated in a clear and concise manner. Many are implied.

Design issues are presented in this section.

B-5.4.8 Generate

This section clearly contains numerous requirements for IRRAS. However, they are not stated in a clear and concise manner. Many are implied.

Design issues are presented in this section.

B-5.5 Reports

No report formats are specified. This is not consistent with the type of information presented in previous sections, specifically design details.

This section clearly contains numerous requirements for IRRAS. However, they are not stated in a clear and concise manner. Many are implied.

Design issues are presented in this section.

B-5.5.1 Base Case and Function Keys

The remaining sections present enough information for the sections to be understandable without any ambiguities. However, the language is such that some requirements are implied rather than explicitly stated. Design issues are also presented.

B-5.6 Build Fault Trees

This section describes the output devices to be supported and a limited number of functions to be performed, apparently when using the fault tree editor. However, several requirements are implied rather than explicitly stated.

B-5.7 Fault Tree Graphical Editor

The graphical fault tree editor requirements are presented in this section. However, in many cases the language used does not explicitly state the requirements. Design issues are interspersed throughout the section. No easily identifiable list of inputs, outputs, or processing is provided.

The section begins by identifying the need for mouse support as well as arrow key support.

B-5.7.1 Menus

Three requirements are listed, but numerous others are implied. Apparently when certain menu items are selected, an additional pop-up menu will be invoked that must satisfy the listed requirements as well. Design issues are raised but not resolved.

The section provides enough information to provide the basis for an understanding of the intended functions but does not provide the detail required for thorough evaluation of the functionalities presented.

B-5.7.2 Exit and Show Options

Without explicitly stating the requirement, these two sections provide enough information to imply the requirements to be implemented.

B-5.7.3 Build

The symbols to be provided and used by the editor are described. An illustration would be appropriate.

B-5.7.4 Edit

This section describes the options to be provided when the user selects the edit option. However, this is not explicitly stated but is implied.

Apparently there are multiple options that are to be made available to the user.

B-5.7.5 Text

This section provides information concerning editing operations on text and shape names. Most requirements are implied rather than explicitly listed and clearly defined.

B-5.7.6 View

This section presents information that apparently describes the options to be provided if a user selects the view option from a higher level menu. The requirements are presented as definitions rather than stated as requirements.

B-5.7.7 File

The options that are apparently to be made available as file operations are described. The section appears to be a description of an existing system. It is not clear if new functionality has been introduced.

B-5.7.8 Name

The options that are apparently to be made available as 'name' operations are described. The section appears to be a description of an existing system. It is not clear if new functionality has been introduced.

B-5.7.9 Level of Detail Changed

At this point is was determined to not provide comments and observations on each subsection because the comments were becoming redundant. The remaining sections would be described in the same fashion as the previous two. The information provided is descriptive in nature. Many requirements are implied rather than explicitly stated.

B-5.7.10 Section Summary

As with previous sections of the Functional Requirements, the Fault Tree Graphical Editor section clearly contains numerous requirements for IRRAS. However, they are not stated in a clear and concise manner. Many are implied; several are not complete. The information is presented more as a description of an existing system than as requirements for the development of a new system.

Several requirements are stated so that reference to an earlier version of IRRAS is definitely implied. For example, consider paragraph L, "Change all text justification for all written tests *from now on* to left, right or center." (Italics added.)

B-5.8 Fault Tree Graphical Pager

No information is provided to identify what is being described. Options, apparently available through a menu, are presented and briefly described.

The information provided in this section appears to be a description of an existing system. It is not clear if the items described are enhancements or existed in a previous version.

B-5.9 Fault Tree Analysis

No information is provided to describe the subsystem for which the requirements will be listed.

The section does provide some information about the processes to be performed and supplies some of the parameters. It is not clear if all the required parameters have been described in this section.

Requirements are implied without explicitly stating them. For example, Subsection A indicates that all operations are to be performed on highlighted/marked systems. This implies that the user must, in some way, be able to mark or highlight the systems on which all operations are to be performed. The section does not explicitly list what is to be included in the phrase "all operations."

Several subsections include wording such "Show the description" and "Show events" without providing details concerning what is to be "shown" or what "shown" means; does "shown" imply CRT output, printer output, or file output.

Section B states "Provide tools to obtain different views of cut sets" and then presents descriptions of cut sets, importance measures, and uncertainty. Three views of cut sets are apparently presented, but there is no discussion of any "tools" to obtain the views. Apparently the views described are to be prepared and presented to the user and the code to perform these functions are the "tools." The lack of an introduction with a high-level view of the intent of this section can cause confusion.

The section also provides information about editors that apparently must be available to the user to modify the data available for analysis.

B-5.9.1 Cut Set Editor

A brief description of functions apparently required to be performed by the editor is supplied. The descriptions contain many design issues and other requirements can be derived from the requirements as they are currently stated.

B-5.9.2 Logic Editor

The section provides a few definitions and then describes what apparently are to be operations for the user who invokes the logic editor. Few details are provided. The reader must infer or derive requirements from the descriptions provided.

Many design issues are addressed, giving the perception that an existing system is being described.

B-5.10 Create Event Trees

This section describes and/or lists the requirements for creating event trees. No introductory paragraph is provided to clearly state the intent or purpose of this section.

B-5.10.1 Link Event Trees

The requirements have a few design issues imposed in them. Requirements appear to be descriptive of an existing system, describing what functionality is desired when menu options are selected, but not explicitly stating the requirements.

B-5.10.2 Event Tree Plotter

The requirements identify the drawing output destination and required drawing formats.

B-5.10.3 Remaining Options

The remaining options, Load Event tree Graphics and Logic, Extract Event Trees, and Define Plotter Pens appear to be descriptions of existing menu options. However, the requirements listed, also described as existing menu options, are fairly clear and understandable.

B-5.11 Event Tree Graphical Editor

This section does not contain an introductory paragraph that explains the intent of the section or describe the function being discussed.

B-5.11.1 Definitions and Cursor Control

The first two sections contain definitions and identify cursor control requirements.

B-5.11.2 Menus

This section apparently discusses actions that the user may perform on menus. Only familiarity with the system provides meaning to the section. Several pop-up menus are displayed as certain options are selected from other menus. Each of the pop-up menus may be repositioned on the screen. The menus may be removed from the screen when desired.

B-5.11.3 Exit

This section identifies two ways to exit the screen without saving changes. The requirement is not explicitly stated, appears to describe existing functions, and presents design issues.

B-5.11.4 Additional Menu Options

The remaining subsections appear to be descriptions of options found on existing menus or to be placed on menus as options. Requirements are not explicitly stated, but many are worded in such a manner to make the meaning fairly clear. However, even some of the requirements that appear to be stated in a clear manner could have requirements derived from them. For example, section H, Ends, indicates that the user is to be able to edit the end states for any sequence. This implies that the user should be able to edit each field of the end state as well as each of the attributes associated with the end state or with each of the contained fields.

Appendix B

Section N, Text Justification, contains wording that makes it very apparent that an existing system will have changes applied to it.

B-5.12 Event Tree Analysis

This section, as implied by the title, discusses the requirements associated with event tree analysis. There is no introductory paragraph providing a high level overview or description of the section and the planned functionality.

B-5.12.1 Analyze Sequences

This section indicates that tools are to be provided to allow the user to create and manipulate cut sets. It is not clear if the tools mentioned are the code that will be produced to satisfy the requirements, but it is implied in the way the requirements are organized and stated.

The section is subdivided into eight subsections. The first is a definition. The second indicates that all operations are to be performed on highlighted/marked events. Apparently the operations to be performed are described in the following sections. It is implied that the user will be able to highlight or mark events in some manner.

The third section contains requirements for generating minimal cut sets using the parameters listed. Many of the parameters listed contain information that imply that certain activities have been or must be performed. Restrictions for the generation of cut sets are also provided.

The remaining sections are similar to the third in that requirements are provided. However, many contain enough information to imply that other activities have or will take place or that the user should have already performed some action to prepare parameters for the function to be performed.

B-5.12.2 Display Results

This section indicates that tools are to be provided to allow the user to obtain different views of the cut sets. Neither the tools nor the views are explicitly identified. Apparently the "tools" to be provided are the functions that are described in the subsections of Section B.

This section provides considerable information concerning the selection and displaying of results. Sections appear to be descriptions of existing functionalities and how they are performed instead of describing what is to be done.

B-5.13 Modify Data Base

There is no introductory statement identifying what is to be described in the following sections. Only familiarity with the system allows the reader to understand many of the functions described.

B-5.13.1 Family

The information required to identify a family is presented as field descriptions, providing considerable design information.

The second subsection identifies functions, apparently to be performed on families. The add function does not indicate if all of the previously defined fields must have data entered. The reader should assume so.

B-5.13.2 Event Tree

Similar to the previous section, this section identifies the data that is contained in an event tree and then the function section apparently describes the activities that the user is to be allowed to perform on event trees.

Some requirements are not explicitly stated. Some requirements imply that the user has already performed or will perform some action in order to implement the current function.

B-5.13.3 Remaining Sections

The remaining sections, Systems, End States, Basic Events, Attributes, Gates, and Graphics are similar to the two described. Each presents a list of data fields to be included followed by a list of functions that the user will be able to perform. Considerable detail is provided but, in some cases, detail is also missing.

B-5.14 Reports

This section presents the requirements associated with producing reports using the IRRAS. All reports are output to the screen printer, or a user named file.

Reports are to be produced on highlighted/marked families, basic events, systems, event trees, sequences, and end states. Therefore, it is implied that the user be provided a mechanism whereby these items can be marked or highlighted.

The items to be included in the various reports are itemized. No report formats are specified.

B-5.15 Utilities

This section lists and describes the utility options to be provided for the user. No introductory paragraph is provided to give a high level overview of the functions to be included in "utilities."

B-5.15.1 Load MAR-D Data

This section is concerned with loading data for and into the IRRAS system. Valid formats for various file types are provided in an appendix.

The information appears to be fairly clear, but the section does contain instances where it is implied that the user has already performed some action or will perform it in order to continue with or initiate the function being described. These actions are not explicitly stated.

B-5.15.2 Remaining Subsections

Similar to the preceding subsections, the remaining subsections present information that is fairly clear but requires the user, or permits the user, to imply other, additional requirements. Some requirements are not explicitly stated while others are.

B-6. SUMMARY

B-6.1 Functional Requirements

The Functional Requirements Document is neither a functional description (FD) nor a system requirements specification (SRS).

A functional description provides the reader an overview or a description of the functions to be performed by the system without providing low level details. An SRS provides detail of WHAT the system is to do without describing HOW it will be accomplished. How the system is to perform is addressed in the software design document (SDD).

The Functional Requirements Document appears to be a combination of an FD, an SRS, an SDD, and a description of an earlier version of IRRAS.

B-6.2 Functional Requirements or Enhancements

The Functional Requirements Document definitely presents numerous requirements for the IRRAS. However, the document is written as though it is describing changes and/or enhancements to an earlier version of IRRAS. Several sections contain wording such as 'from now on' that clearly indicate that the document, or at the very least, portions of the document, is describing an existing system without clearly identifying new functionality, enhancements, or changes to the system.

B-6.3 Descriptive Introductions

In most sections there is no introductory paragraph that presents an overview of the contents of the current section and what the overall purpose of the function is. This can cause confusion. The lack of an introductory paragraph makes it difficult for another, not familiar with the system, to fully understand what information is being presented.

B-6.4 Using the Functional Description for Testing

It is possible to extract considerable information from this document that will be useful in preparing test cases and test procedures. However, many implied requirements will be overlooked because they are not explicitly stated. As a result, some functionalities will not be tested.

Although very useful in the development of test cases and test procedures, the document does not provide the detail necessary to provide the traceability or audit trail necessary to perform thorough and detailed testing of a software system. The document provides considerable detail but too much detail must be derived from improperly stated requirements. Such a subjective interpretation of requirements will cause the introduction of biases into test results.

This does not mean that the Functional Requirements Document does not or will not provide adequate information to perform a variety of functional tests of the IRRAS. As stated, the document does provide sufficient data to perform considerable testing. The results cannot be interpreted to indicate that the IRRAS performs all indicated functions as required, even if all of the tests developed from the Functional Requirements Document are successful. Successful completion of all tests developed from the Functional Requirements will indicate that the IRRAS performs all of the functions identified by the test script writer. It is possible that the subjective interpretation of the requirements has overlooked critical and non critical functionalities.

The Functional Requirements document can and should be used for testing purposes and the results will be beneficial. Testing will be prolonged as implied requirements are extracted, requiring more time for test preparation and execution.

B-6.5 Requirements Language

Many sections present considerable information without explicitly stating requirements. The language used implies requirements without explicitly stating the requirements. As indicated earlier, many sections appear to describe an existing system without providing adequate detail.

Many of the deficiencies, perceived or otherwise, are caused by the language used in preparing the document. For example, Subsection B, Display Results, of Section XI, Event Tree Analysis, states "Provide tools to obtain different views of cut sets." The document does not indicate that the tools mentioned are to be developed as part of the system. Nor does it explicitly identify the "different views" that are to be provided.

B-6.6 Design Issues

Numerous design issues are presented within requirements without a clear and concise description. It is not readily apparent when design issues are presented as requirements or as design constraints.

If the document describes enhancements to an existing system, or even it is describing an existing system, that fact should be clearly stated and the version of software referenced identified.

B-6.7 General Capabilities

Function keys and functionalities described in the General Capabilities section are identified throughout the remaining sections of the Functional Description. However, the General Capabilities section does not identify the forms to display function key mappings and provide the described functions nor does it indicate that those forms requiring the capability will indicate that it is required. It is not possible, therefor, to determine if the requirement has been satisfied.

Those sections that will not implement the general capabilities of the system should explicitly state the omission of the capabilities just as those sections that will implement the general capabilities should explicitly identify their use.

B-6.8 Traceability

No other source of functional requirements has been provided. Neither is the reviewer knowledgeable enough in probabilistic risk assessment techniques and methodologies to provide a determination pertaining to tracing the functions identified to accepted standards and/or methodologies and practices.

It is clear that a number of software change requests have been submitted to the development team. However, it is not known if these requests were to resolve a discrepancy discovered in the software or to request an enhancement or change to the system.

Even though the Functional Requirements Document appears to describe existing system functionalities, the reviewer must assume that this functional requirements document is at the root of the document hierarchy for the IRRAS project. Other project documentation should be traced to the functional requirements if no other documentation exists.

B-6.9 Completeness

When the number of implied requirements, or the need to derive requirements from the information contained in the Functional Requirements Document, is considered, the Functional Requirements Document is not complete.

If the functions required to perform probabilistic risk assessments (PRAs) will be used as a measure to determine completeness, the reviewer has neither the training nor the expertise to perform that activity. No conclusions as to completeness of functionality for the IRRAS system for performing PRA activities will be attempted in this review.

B-6.10 Correctness

When the number of implied requirements, or the need to derive requirements from the information contained in the Functional Requirements Document, is considered, the Functional Requirements document cannot be considered 100% correct. Neither can the document be considered as completely incorrect.

If the functions required to perform PRAs will be used as a measure to determine correctness, the reviewer has neither the training nor the expertise to perform that activity. No conclusions as to correctness of functions described for the IRRAS system will be attempted in this review.

B-7. SUGGESTIONS AND RECOMMENDATIONS

B-7.1 Rewrite Requirements Document

The Functional Requirements Document should be rewritten in a requirements format. The requirements should be clearly and concisely stated. Design constraints should be clearly identified.

In many sections, simply rewriting the requirement as a statement of what must be accomplished will provide the necessary clarity.

B-7.2 Prepare Design Documentation

Using the rewritten requirements documentation and existing code, a Software Design Description (SDD) should be prepared. IRRAS is a powerful and extremely useful system. IRRAS will continue to grow and will require that changes be implemented as technological advancements are continually incorporated into more powerful computer hardware components. Preparing an SDD will greatly facilitate applying changes and enhancements to the IRRAS in future versions. The lack of appropriate system documentation will be the source of deficiencies as the software matures.

B-7.3 Document and Software Configuration Management

The methodology employed to control both documentation and software changes needs to be more rigorous. Changes to software should be reflected in system and user documentation as new versions are released and/or deficiencies corrected.

B-7.4 User Documentation

All user documentation should be updated to reflect the status of the most recently released version. User documentation updates should be provided as deficiencies are corrected and enhancements introduced. As new versions of the system are released, new versions of user documentation should also be released.

B-7.5 System Documentation

Documentation should be prepared for those that will be required to maintain the system. This documentation is referred to as system documentation and programmer documentation.

Currently the individuals who are responsible for the maintenance of IRRAS have been intimately involved with the development of the system. Their knowledge and understanding of the system is extensive and quite thorough. However, there is absolutely no guarantee that these individuals, or individuals with equivalent knowledge and understanding of the system, will always be available. System documentation detailing system and performance requirements and restrictions should be prepared and maintained so that other individuals, when required, will be able to acquire an understanding of the system adequate to allow them to continue developing, enhancing, and maintaining the system.

The system documentation, when combined with an SRS and SDD will provide a thorough and complete view of the IRRAS. The formal reporting and logging of discovered deficiencies and applied resolutions will provide a clear view of the growth and maturity of the system.

EG&G Idaho

"Providing research and development services to the government"

INTEROFFICE CORRESPONDENCE

Date: August 4, 1992

To: J. L. Jones, MS 2406

From: D. L. Kelly, MS 2405

Subject: COMMENTS ON IRRAS 4.0 DRAFT FUNCTIONAL REQUIREMENTS DOCUMENTS
DLK-20-92

I have reviewed the draft IRRAS 4.0 functional requirements document
transmitted in your letter of July 22 (JLJ-13-92). I reviewed selected
portions of the document, but became rather discouraged and did not
completely review all sections because I found little discussion of how
IRRAS goes about its business. I feel that this is more of an outline of a
requirements document. However, it is a much belated step in the right
direction. Unfortunately, it appears to have not kept up with the IRRAS
(SAPHIRE) code development.

Overall, I think the document may be too sparse in detail to help much in
the development of vital feature tests. Also, the limitations of the code
are not entirely clear to me after reading this document. To write an
intelligent test plan, one needs to understand exactly what is being done by
the code. As an example, in uncertainty analysis, what are the differences
between single, group, and endstate analyses? In generating sequence
cutsets, are there circumstances in which IRRAS will truncate the results?
If so, what are the indicators that such has occurred? I could list many
more similar examples.

I close with a number of specific comments tied to particular sections of
the document. I have avoided being repetitious. For example, comments made
about importance analysis in one section apply generally to importance
analysis in other sections.

Specific Comments

1)   References to IRRAS 2.6 should be changed wherever they occur.

2)   The limit of 10,000 sequences per family stated under B.3 on p.4 is
     overly restrictive, particularly for a large event tree/small fault
     tree PRA. What happens if this limit is exceeded?

3)   B.11 on p.5 contradicts B.4.b on p.4.

J. L. Jones
August 4, 1992
DLK-20-92
Page 2

4)   In the definitions on pp. 7 and 8, there are several errors:

   a)   "p" is not a failure probability in calculation type 9,
   b)   failure rates and times do not have to be input in hours,
   c)   tau is also used as a surveillance test interval in calculation
         types 6 and 7,
   d)   the equation for calculation type 5 is incorrect,
   e)   calculation type S is not discussed,
   f)   process flags W, K, and L are not discussed.

5)   On p. 11, the discussion of cascaded events is too vague to tell what
      the actual requirements are.

6)   Under VI.E on p.14, are there any limits to the number of gate inputs?

7)   On p. 15, numbers 11 and 12, "undefined" should be "undeveloped."

8)   On p. 20 the definition of "cut set" is actually the definition of a
      minimal cutset.

9)   Under VIII.A.3 on pp. 20 and 21:

   a)   For "Event Probability Cut Off," cutsets are not discarded if
         conditions (a) and (b) are met,
   b)   What is a "zone event?"
   c)   The equation under (f) is a "constraint," not a "calculation."
         This is a very severe constraint. Is it a result of using small
         integer arrays in the programming? If so, how difficult would
         it be to fix by going to huge integer arrays? What does the
         code do if this limit is exceeded?

10)   Under VIII.A.6.a on p.22, "sample size" is not a sampling technique.

11)   Under VIII.B.2 on p. 23, the definitions of the importance measures
      are incorrect in some cases and vague in all cases. The requirements
      document should, I think, specify the equations used to calculate each
      importance measure. Also, where is the discussion of uncertainty
      importance, one of the options available in the "Define Constants"
      menu (it is mentioned in passing on p.54)?

12)   Under VIII.B.3 on p.24, what sample estimators are used for the
      uncertainty measures reported by IRRAS? How are the confidence
      intervals on the quantiles estimated? Are the same estimators used
      for both Monte Carlo and Latin hypercube sampling?

J. L. Jones
August 4, 1992
DLK-20-92
Page 3

13)   Under IX.B on p. 27:

       a)   What is the limit on the number of sequences in each event tree?
             How are transfer event trees counted in this limit? What does
             the code do if the limit is exceeded?
       b)   The description of the link editor is incorrect for the editor
             in IRRAS 4.11.

14)   Under XI.A.7 on p. 37, the equation for calculating sequence frequency
       using split fractions is incorrect. It is correct if "MC" is the
       minimal cutset upper bound of the <u>system</u> instead of the <u>cutset</u>. Also,
       the terminology is non-standard; "split fraction" normally refers to
       the probability of an event tree branch, not to the frequency of a
       sequence.

15)   Under XII.J.2 on p. 47, the discussion of histograms does not match
       what is currently in IRRAS 4.11. In IRRAS, the percentage format
       gives a discrete probability distribution, not a histogram. The range
       format gives a histogram, but one of the inputs is the height, not the
       range probability. The area format also gives a histogram; it is not
       mentioned in the requirements document.

cc: J. H. Bickel, MS 2405 𝒥𝓌𝓱𝐵
       S. A. Eide, MS 2406
       T. A. Thatcher, MS 2406
       Central Files, MS 1651
       D. L. Kelly File

Appendix B

EGｓG Idaho

"Providing research and development services to the government"

INTEROFFICE CORRESPONDENCE

Date: November 2, 1992

To: T. A. Thatcher, MS 2406

From: S. D. Matthews, MS 2408

Subject: REVIEW OF IRRAS 4.0 FUNCTIONAL REQUIREMENTS DOCUMENT - SDM-11-92

After reviewing the IRRAS 4.0 Functional Requirements Document (FRS),
December 1991, I have the questions listed below.

1. Is the FRS conformable to any standard? If so, what? If not,
   what is being used and why? (I would recommend ANSI/IEEE Std 830-
   1984.)

2. What quality level is required for IRRAS 4.0 and the IRRAS 4.0
   FRS? (Is the desired quality level documented?) If IRRAS 4.0 must
   have greater than quality level C, is a requirements traceability
   matrix being developed to provide traceability and auditability?

3. What is the purpose of the FRS and who is the intended audience
   for the FRS?

4. What are the user characteristics? (Are legitimate user
   complaints documented and considered in this FRS?) How much
   training will be required for the users?

5. Are there any related software packages?

6. Are there any reference documents?

7. What are the general constraints, e.g., hardware, regulatory
   policies, and interfaces to other applications?

8. What are the design constraints?

9. Are there any requirements for maintainability, portability,
   training, or testability?

10. Are there any external interface requirements, i.e., software,
    hardware, and/or user? For example, what are the required report
    contents? The required screen formats? Are there any required
    software packages such as a DBMS?

NUREG/CR-6145   B-22

T. A. Thatcher
November 2, 1992
SDM-11-92
Page 2

11.   What are IRRAS' performance requirements, e.g., measurable values for table and file sizes, and response times?

12.   What are the relationships between the functional requirements? (A context diagram and a data flow diagram or object model would be useful to show relationships.)

13.   How are the functional requirements delineated? Each functional requirement should contain a description of the requirement, the inputs to initiate the processing, a description of the processing including the algorithms, the resulting output from processing, and any required error handling for invalid data. Functional requirements should specify "what" has to happen, not "how" it will happen, which is a design issue. In addition, all values should show the range and units. For example, what are the acceptable ranges for "probability of failure" and "random number seed"?

14.   Are the functional requirements for the uncertainty analysis (Monte Carlo analysis and Latin Hypercube), DBMS, and graphical, provided in another reference? If so, what? If it is not provided anywhere else, why not?

15.   Is a data dictionary available to define all input, output and processing values?

16.   What are the algorithm references, e.g., Fussell-Vesely Importance, and Birnbaum Importance?

Also, I believe including a summary of IRRAS functionality and operating environment would benefit the FRS readers.

Please call me at extension 6-9575 if you have additional questions regarding this review.

pjc

cc: G. V. Miller, MS 2408
       B. J. Stacey, MS 2408
       Central Files, MS 1651
       S. D. Matthews File

Appendix C

Vital Features Verification and Validation
Software Test Records

Appendix C

Vital Features Verification and Validation
Software Test Records

C-1. INTRODUCTION

The following sections describe the tests that were performed for each vital features and associated sub-feature. The software test record number has the following format:

STR-VFN-XXX-MMYY

where

ST = Three-letter acronym for Software Test Report.

VFN = Two-digit vital feature number (i.e. 01 to 49)

XXX = Three-digit sequential test number assigned by the tester that represents the set of data being used to test the feature (features may be tested with multiple sets of data)

MM = Two-digit code for the month of the test

YY = Two-digit code for the year of the test.

Each software test record provides a description of the test, the computer system used, the expected values or acceptance criteria, and the input and output file names. Software test records with all supporting documentation (actual input and output files) can be obtained from EG&G Idaho, Inc. Anomaly reports that were generated for identified discrepancies are located in Appendix D.

C-2. FAULT TREE ANALYSIS

The vital features for the major group of Fault Tree Analysis are:

1. Cut set generation process

2. Cut set quantification process

3. Cut set quantification process—analysis of single and multiple fault trees.

Software test records 1, 3, 4, 5, 6, and 9 were tested using the DEMO Family. The DEMO Family has simple fault trees to test the analysis features and most of the results can be examined by inspection. The results from all of these tests were found acceptable. There were two points found that should have some discussion about their implementation. First is the message given when the cutoff (either size or probability) is set above any cut set in the fault tree. The message that is given is, "The top event cannot occur (FALSE)!" A better message might be "No cut sets qualify based upon cutoff." Second, a cut set report (empty except for the date, time, and page number) is produced from the Report Module even when there are no cut sets available. It might be better to produce the same message that appears under the Display Option, "No cut sets available."

Appendix C

C-2.1 Cut Set Generation Process

To test the cut set generation process the following three sub-features were tested: cut set probability cutoff (software test record 1), the maximum number of cut sets saved (software test record 2), and that the reduction of the fault tree results in the correct cut sets both in number and type (software test record 3).

The DEMO Family was used to test the cut set probability cutoff (software test record 1). Cut sets were generated in the analysis fault tree module with the probability cutoff at 0.00, 1E-20, 1E-8, 1E-5, 1E-3, 1E-1, and 1.00. The results were verified against the actual boolean reduction and lotus spread sheet calculations. The results were acceptable.

To test the maximum number of cut sets saved cutsets were created from the Peach Bottom IAS fault tree with a cutoff truncation value of 1E-19. Analysis of this fault tree generated more cut sets than the code could save. The IAS cut sets were saved in an ASCII report using the IRRAS report function and then loaded into MS EXCEL 4.0. The total number of events and cutsets were added to test the formula for the maximum number of cut sets saved. The results did not agree with the formula. It was found that the total number of cut sets saved is <32,000 not <32,700. Anomaly report AR-02-07-93-001 was generated.

The DEMO Family was used to test that SAPHIRE generated the correct cut sets (software test record 3). Cut sets were generated in the analysis fault tree module with the probability cutoff at 0.00, 1E-20, 1E-8, 1E-5, 1E-3, 1E-1, and 1.00. The results were verified against the actual boolean reduction and lotus spread sheet calculations. The results were acceptable.

C-2.2 Cut Set Quantification Process

The DEMO Family was used to test the cut set quantification process (software test record 4). Cut sets were generated in the analysis fault tree module with the probability cutoff at 0.00, 1E-20, 1E-8, 1E-5, 1E-3, 1E-1, and 1.00. The results were verified against the actual boolean reduction and lotus spread sheet calculations. The results were acceptable.

C-2.3 Cut Set Quantification Process—Analysis of Single and Multiple Fault Trees

To test the cut set quantification process through the analysis of single and multiple fault trees, the following four sub-features were tested: that the equation for the min-cut upper bound is correct (software test record 5), the ability of SAPHIRE to solve a small fault tree of less than 100 gates (software test record 6), the ability of SAPHIRE to solve large fault trees of greater than 1500 gates and with complemented events in the fault tree (software test record 7), and the ability of SAPHIRE to solve small fault trees with complemented events in the fault tree (software test record 9). The ability of SAPHIRE to solve medium fault trees of greater than 100 gates but less than 1500 gates and with complemented events, software test record 8, was not performed since it is bounded by software test records 7 and 9.

The DEMO Family was used to determine if the equation for the min-cut upper bound is correct (software test record 5). Cut sets were generated in the analysis fault tree module with the probability cutoff at 0.00, 1E-20, 1E-8, 1E-5, 1E-3, 1E-1, and 1.00. The results were verified against the actual boolean reduction and lotus spread sheet calculations. The results were acceptable.

The DEMO Family was used to evaluate the ability of SAPHIRE to solve a small fault tree of less than 100 gates (software test record 6). Cut sets were generated in the analysis fault tree module with the probability cutoff at 0.00, 1E-20, 1E-8, 1E-5, 1E-3, 1E-1, and 1.00. The results were verified against the actual boolean reduction and lotus spread sheet calculations. Both the cut sets and the fault tree quantification were correctly generated by SAPHIRE.

To evaluate the ability of SAPHIRE to solve large fault trees of greater than 1500 gates with complemented events in the fault tree (software test record 7), five fault trees from the LaSalle database were combined into a new tree. This new tree was also evaluated in CAFTA. The results were considered acceptable since they were comparable between the two codes. Even though the quantification and cut set results from SAPHIRE compared to the results from CAFTA, it was found that SAPHIRE and CAFTA reported a different number of gates per tree. CAFTA recognized 1856 gates while SAPHIRE recognized only 612 gates. It was found that by changing the amount of RAM memory, the size of the tree that could be analyzed would change. However, after maximizing the memory to 627K out of 640K, it was not possible to analyze a tree past 612 gates without receiving the error message, "Too many gates in this tree!" Therefore, the maximum number of gates (2500) was never reached and is misleading since the total number of gates is dependent on the amount of available RAM. How SAPHIRE calculated the total number of gates is also questionable since the total number of gates that SAPHIRE calculated for this test was less than 3 times the amount CAFTA calculated. The impact is minor but leads the user to believe that the problem may be in a different area other than memory. Also, it is annoying since SAPHIRE will corrupt the database and require recovery actions if it is unable to process the tree. The appropriate documentation should be changed.

The DEMO Family was used to evaluate the ability of SAPHIRE to solve small fault trees with complemented events in the fault tree (software test record 9). Cut sets were generated in analysis fault tree module with the probability cutoff at 0.00, 1E–8, 1E–5, 1E–3, 1E–1, and 1.00. The results in the report produced were verified against the actual boolean reduction and lotus spread sheet calculations. The results were acceptable.

C-3. EVENT TREE ANALYSIS

The vital features for the major group of Event Tree Analysis are:

1. Sequence generation process

2. Sequence cut set generation process

3. Quantifying process

4. Capability of performing analysis on a single event tree

5. Capability of performing analysis on multiple event trees.

C-3.1 Sequence Generation Process

To test the sequence generation process, three sub-features were tested: the generate sequences process, the sequence editor, and the link editor.

The VV Family was used to test the generate sequences portion of the event tree analysis. The purpose of software test record 10 was to determine if SAPHIRE would generate the correct sequence logic. Seven items were evaluated for software test record 10. The first item was to determine if the code correctly generated the sequence logic. The second item was to determine if all sequences have correct logic generated except for those sequences with end states that are designated as "ok," "success," or "ignore," and those sequences that are not to be included in a particular analysis by marking the end state with an "@" symbol. The third item was to determine if the code would generate correct sequence logic if a mutually exclusive top was added to each sequence logic. To test this item a fault tree was created for use as a mutually exclusive top event substitution during the sequence logic generation process. The fourth item was to verify that the code would correctly output the sequence logic to the screen, printer, and to a file in the family directory. The fifth item was to determine if the code would automatically assign sequence numbers to each sequence if this menu option was chosen. The sixth item was to determine if the code would automatically assign an alpha character to each sequence or allow the user to manually name each sequence if this menu option was chosen. For all of the above items the SAPHIRE generated the correct sequence logic and correctly made the requested changes.

The VV Family was used to test the generate sequence editor portion of the event tree analysis. The sequence editor allows the user to edit sequence names, end states, and assign frequencies after an event tree has been created. The purpose of software test record 11 was to determine if correct sequence logic would be generated using the sequence editor to modify the event tree. Seven items were evaluated for software test record 11. The first item was to determine if modification of the frequency would produce the correct changes to the sequences. The second item was to determine that sequence logic will not be generated for those sequences marked with an "@" symbol. The third item was to perform various changes by using the line edit option. The fourth item was to check the transfer function and determine if the user can edit the transfer event tree end state information in the same manner as that for the initiating event tree. The fifth item was to test the global replace option. This option allows a specific sequence name to be searched for and replaced with a new sequence name. The sixth item was to test the header edit option. The final item was to determine if the change transfer option will correctly work as a toggle switch to turn on and off transfer points. For all of the above items the SAPHIRE generated the correct sequence logic and correctly made the requested changes.

The VV Family was used to test the link editor portion of the event tree analysis. This option allows the user to define linkage rules and generate sequence logic. The purpose of software test record 12 was to determine if correct sequence logic would be generated when linkage rules were invoked.

The first item was to test multiple branches. Multiple branches are used when the user wants to consider more than one possible failure path for an event tree top event. For example, a user may want to show a failure path for failure of one safety relief valve to reseat and a separate failure path for failure of more than one safety relief valve to reseat. All sequences are correct except for Sequence 5. This sequence should have been A succeeds, B fails and is replaced by B1, and C fails and is replaced by C1. SAPHIRE did not replace C with C1. Anomaly report AR-12-0893-001 was generated.

As a further test an "@" symbol was placed on the end state of Sequence 3, therefore no logic was generated. All sequences are correct except for Sequences 4 and 5. Sequence 4 should have been A succeeds, B fails and is replaced by B1, and C succeeds and is replaced by C1. SAPHIRE did not replace B with B1 and C with C1. Sequence 5 should have been A succeeds, B fails and is replaced by B1, and C fails and is replaced by C1. SAPHIRE did not replace B with B1 and C with C1. Anomaly report AR-12-0893-002 was generated.

From these two tests SAPHIRE is not correctly processing event tree sequence linkage rules for event trees with multiple branches. Given the potential for PRA analysts to develop event trees with more than one possible failure path for an event tree top event, this is considered to be a significant discrepancy between the way SAPHRIE should work and its actual operation.

Other options that were evaluated to test the link editor included the Add, Modify, Delete, Restore, and Copy options. The results from exercising each of these options were acceptable.

C-3.2 Sequence Cut Set Generation Process

To test the cut set probability cutoff (software test record 13), the Peach Bottom sequence A17 was used and cut sets were generated for 4 different probability cutoff values. This test was repeated for cut sets using fault trees and then sequences to generate the cut sets. The results matched the appropriate cutoff probabilities in all cases.

Software test records 14 and 15 involve testing if SAPHIRE will generate the correct cut sets for an event tree sequence using the fault tree method in one case (software test record 14) and the cut set method in the second case (software test record 15). These tests are also covered by software test records 24 and 25, respectively. Software test records 24 and 25 involve testing if SAPHIRE will generate the correct cut sets for all event tree sequences respectively using the fault tree method and the cut set method.

The test that was performed to determine the maximum number of cut set that can be saved for one sequence (software test record 16) was essentially the same as that for software test record 2. The results were also the same.

Default flag was the next sub-feature exercised to test the sequence cut set generation process. A default flag designates that a change set has been previously established. For example, a change set may have been previously

established that says a point estimate should be used for a particular event tree top. The purpose of software test record 17 was to over-write previous change sets and determine if the sequence logic incorporated this new change set. Various combinations of new changes sets were tried using this option and in all cases the sequence logic was correct.

The update cut sets feature (software test record 18) was examined using the DEMO Family. All the sub-features tested were found acceptable. It is important to remember that for the alternate (or current) case, the results are dependent upon those cut sets available. If, for example, the most stringent probability cutoff (1.00) had been used first it would have eliminated all cut sets in the current case for subsequent analyses. For all testing, both the base and current case cut sets were generated without cutoffs, and the current case was always "renewed" to contain all sequence cut sets for each test cutoff.

To test the cut set quantification process (software test record 19), sequence A17 from the Peach Bottom database was used to generate a new cut set report that included the cut set members and the calculated probability. The files were loaded into MS EXCEL 4.0. The results did not match the IRRAS file. The EXCEL values were off by a factor of 1E–4, probably due to the initiating event which is not included in the cut sets, and therefore not calculated by the EXCEL macro. No anomaly report was generated.

Software test record 20 involves evaluating uncertainty analysis results for event tree sequences. Software test record 20 is covered by software test records 32 and 33.

To test the split fraction calculation (software test record 21), values from two systems were generated using IRRAS. These were then combined using the equation provided in the vital features list to calculate a sequence split fraction by hand. The values were compared to the IRRAS values. The results were consistent with the hand calculated values; however, it appears that SAPHIRE may not be rounding the results correctly. SAPHIRE results may be truncated after three decimal places. The problem appears to be due to incorrect or lack of rounding techniques. The impact is considered to be minor; however, this could be an important factor if rounding is treated in the same manner in other portions of SAPHIRE such as in the quantification modules (the rounding problem was also noted in software test record 23 on the minimal cut set upper bound calculation).

To test the base case update feature (software test record 22), an alternate case was developed that changed three basic event probabilities. Cut sets for the alternate case and the base case were determined. Even though SAPHIRE correctly performed the quantification when the base case update function was tested, the report function printed out the alternate change set values instead of the base case probabilities. Since the report function does not identify whether the print-out is for the base case or the alternate case, the user could inadvertently use the wrong case.

C-3.3 Quantifying Process

To test the minimal cut set upper bound sub-feature (software test record 23), the cut sets and their probabilities from sequences 2 and 3 of the DEMO Family were downloaded into a text tile. The text file was uploaded into MS EXCEL and manipulated. The results were acceptable, except that there still appears to be a problem with rounding.

C-3.4 Capability of Performing Analysis on a Single Event Tree

Software test records 24 and 25 involve testing if SAPHIRE will generate the correct cut sets for all event tree sequences using the fault tree method and the cut set method, respectively. To test this two systems were extracted from the LaSalle-4 database and a small event tree was created that included these systems. Sequence cut sets were generated in IRRAS and CAFTA using both the fault tree method and the cut set method. The results were acceptable.

C-3.5 Capability of Performing Analysis on Multiple Event Trees

Perform Analysis on Two Event Trees Using Fault Tree Method, software test record 26, is bounded by software test record 24, Perform Analysis on Event Tree Using Fault Tree Method, and by software test record 27, Perform Analysis on Five Event Trees Using Fault Tree Method.

Software test records 27 and 29 address testing the capability of performing analysis on multiple event trees. Software test record 27 covers the case of using the fault tree method. Software test record 29 covers the case of using the cut set method. For both tests the results were acceptable.

Perform Analysis on Two Event Trees Using System Cut Set Method, software test record 28, is bounded by software test record 25, Perform Analysis on Event Tree Using Cut Sets, and by software test record 29, Perform Analysis on Five Event Trees Using System Cut Set Method.

C-4. UNCERTAINTY ANALYSIS

The vital features for the major group of Uncertainty Analysis are:

1.   Single event sampling

2.   Uncertainty analysis of sequences.

The uncertainty analysis was checked for both the Monte Carlo (MC) and Latin Hypercube sampling (LHS). For both of these sampling techniques, single event analysis (i.e., a single basic event) and multiple event analysis (i.e., a group of basic events, such as system or sequence cutsets) were investigated. Additionally, the random number generator that is used in SAPHIRE was checked for its suitability. The uncertainty analysis testing is covered by software test records 30 through 33.

For the single event uncertainty analysis, all SAPHIRE available distributions were checked. These distributions are: uniform, exponential, lognormal, chi-square, normal, gamma, beta, and histogram. Out of these eight distributions, only one passed all the acceptance criteria tests (the specific acceptance criteria is spelled out in the particular software test record). The predominant reason why seven of eight distributions failed some of the acceptance criteria tests was that SAPHIRE allows the user to input improper distribution parameters. In some cases, SAPHIRE will print an error message indicating that improper distribution parameters were used, but still attempt to perform the uncertainty analysis using the improper parameters. This could lead to incorrect results, and consequently, is deemed unacceptable. Anomaly reports AR-30-0893-001 (exponential), AR-30-0893-005 (chi-square), AR-30-0893-008 (beta), AR-30-0893-010 (normal), AR-30-0893-011 (gamma), AR-30-0893-012 (lognormal), and AR-30-0893-013 (uniform) discusses the lack of parameter validation for the respective distributions.

For those test cases that did not contain improper distribution parameters, another generic problem was found for both the MC and LHS. The problem is that SAPHIRE appears to have an internal arithmetic precision problem when distribution sampled values are small (approximately 1E-14 to 1E-15 or smaller). When a small value is generated by SAPHIRE, the value is frequently rounded down to a value of zero. It is assumed that this problem appears because of the arithmetic operations in the min-cut upper-bound approximation. Anomaly reports AR-30-0893-002 and AR-30-0893-003 give additional detail on this generic problem.

The chi-square distribution has a discrepancy that could result in confusion. When using the chi-square distribution, the SAPHIRE Technical Reference Manual: IRRAS/SARA Version 4.0 (NUREG/CR-5964), states that the chi-square distribution that is sampled (Y) is given by:

$$Y = aX,$$

where X is distributed as $\chi^2(k)$, k is the uncertainty value, and a = (probability value)/(uncertainty value). Thus, if the probability value = 2 and the uncertainty value = 1, then the sample chi-square distribution results should equal $2\chi^2(1)$.

All the tests which had a multiple of the chi-square distribution did not give sampling results as a multiple of the chi-square distribution. Instead, the results only seemed to give the chi-square distribution part [e.g., the results were very close to what would be expected for $\chi^2(1)$, not $2\chi^2(1)$]. Other than this discrepancy of not resulting in a multiple of the chi-square distribution, the chi-square sampling for n degrees of freedom seemed to be close to tabulated values of chi-square percentiles. Consequently, if the user ignores the probability value and only uses the uncertainty value as the degrees of freedom for the chi-square distribution, the results should be as expected. See anomaly report AR-30-0893-004 for details on the test results.

The gamma distribution has two anomaly, one minor and one major. The minor anomaly for the gamma distribution is that when the uncertainty value equals n/2 (where n equals the degrees of freedom) and the probability field equals 2*(uncertainty value), then the gamma distribution will be chi-square distributed with n degrees of freedom. Consequently, it would be expected that the sampling for identical gamma and chi-square distributions should be exactly the same. The Monte Carlo sampling for the gamma distributions which are $\chi^2(1)$ and $\chi^2(2)$ differs a little from the Monte Carlo sampling for the corresponding chi-square distributions. This inconsistency could cause some confusion. The Latin Hypercube sampling for the gamma/chi-square distributions does not display this inconsistency. See anomaly report AR-30-0893-007 for details on the test results.

The major anomaly for the gamma distribution is that with probability value of 1E−4 and uncertainty value of 10,000, the MC sampling seemed to return the same value (7.068E−6) for each sample. Consequently, all the sampling distribution results were in error. Also, no error messages were displayed alerting the user to the problem. Two other statistical packages (@RISK and SAS) were used to see if the choice of parameters for the gamma distribution is a generic problem. Both @RISK and SAS correctly handled the gamma distribution with these parameters. Thus, some extreme parameter values for the gamma distribution could result in erroneous results. The users of SAPHIRE should exercise caution and test specific gamma distributions if the gamma distribution must be used. See anomaly report AR-30-0893-006 for details on the test results.

The exponential distribution has a minor anomaly. When using the exponential distribution, the distribution results should be the same as those from the gamma distribution (when the uncertainty value parameter equals 1 in the gamma distribution). For the LHS analysis, the sampling for EXP(Θ) is not exactly the same as GAMMA(Θ,1) while for the MC sampling the two distributions produce exactly the same sampling. While this is not a critical error (since both sampling methods for the LHS distributions seem to produce accurate results), to be consistent, the EXP(Θ) and GAMMA(Θ,1) sampling for LHS should be the same. (This could also be labeled as another anomaly for the gamma distribution). See anomaly report AR-31-0893-001 for details on the test results.

The beta distribution has a major anomaly. For the beta distribution, the parameters that were specified for several distributions (e.g., (1) prob. value = 0.001, uncertainty value = 0.2; (2) prob. value = 1E−6, uncertainty value = 2; and (3) prob. value = 1E−6, uncertainty value = 800000E) should result in beta distributions which are legitimate. For these beta distributions, IRRAS actions ranged from giving runtime errors (causing the need to recover the database) to running seemingly alright but giving nonsensical results. Thus, some parameter values for the beta distribution could result in erroneous results. The users of SAPHIRE should exercise caution and test specific beta distributions if the beta distribution must be used. See anomaly report AR-30-0893-009 for details on the test results.

No anomalies were found for the multiple event uncertainty analysis (note: only distribution and parameters that checked out alright from the single event uncertainty analysis tests were used for the multiple event uncertainty analysis tests). Also, tests checking the random number generator did not reveal any anomalies.

C-5. CHANGE SET FEATURES

The vital features for the major group of Change Set Features are:

Appendix C

1. Creating change sets

2. Applying changes to models

3. Flags to be applied in change set events.

The change set features were examined using either the DEMO Family or the VV Family. The VV Family has simple event trees and fault trees to test the change set features and most of the results can be determined by inspection.

C-5.1 Creating Change Sets

Using the DEMO Family, the direct assignment of individual probability to an event (software test record 34) was tested by creating a change set with a probability change on basic event C-MOV-A from 5.0E–3 to 0.1. The changes were verified on the change set report, basic event probability report, and the sequence cut set reports. The results were acceptable.

Using the DEMO Family, manual designation of a group of .vents with the same probability (software test record 35) was tested by creating a change set to change the values of basic events C-CV-A and C-CV-B from 1.0E–4 to 0.1 and change the values of basic events C-MOV-1 and E-MOV-1 from 1.0E–3 to 0.1. The results were acceptable.

Using the DEMO Family, establishing search criteria for a class of events (software test record 36) was tested by setting attribute 1 to yes on basic events DG-A and DG-B. Set attribute 3 to yes on basic events C-PUMP-A, - B, and E–PUMP-A, -B. Set attribute 1 and 3 to yes on basic events C-MOV-1 and E–MOV-1. On the class change set menu, put an asterisk in the name block, set calculation type to 1, and set the probability to 0.1. Three attribute combinations were used to check this search criteria. Class attribute 1 is set to yes for the first search criteria test. Set class attribute 3 to yes for the second search criteria test. Set class attributes 1 and 3 to yes for the third search criteria test. The results obtained for STR 36 were correct; however, an anomaly report was generated (see anomaly report AR-36-0793-001) since the menus and the user documentation do not explain how to mark a class of events.

C-5.2 Applying Changes to Models

Entering the change set menus and applying changes to system data (software test records 37 and 38) was tested by the change set features discussed above (software test records 34, 35, and 36); therefore, no separate test procedure was developed.

C-5.3 Flags to be Applied to Change Set Events

The Sensitivity Process Flag ("S" Flag) was checked (software test record 39) but not tested since it is an undocumented feature and has no criteria by which to measure results. Anomaly report AR-39-07-93-002 was generated. It appears that this flag allows the analyst to vary the probability of an event or group of events over a specified range and plot the change to core damage frequency. This is a very useful feature that needs to be documented.

The "X" process flag was tested (software test record 40) in the DEMO Family. The ability of the "X" process flag to substitute a developed event for both a system and a train of the system was tested. The "X" process flag substitutes a developed event probability instead of processing the system logic on the failed event tree path, and for the success event tree path the code will use a "delete" term. The "X" process flag was tested at a system level and at a train level and in both cases the results were acceptable.

The "Y" process flag was tested (software test record 41) in the DEMO Family. The ability of the "Y" process flag to substitute a developed event for both success and failure logic in an event tree was tested. The results were acceptable.

The "T" process flag was tested (software test recore 42) in the VV Family. The ability of the "T" process flag to use system logic on the failure event tree path and the success of system logic on the event tree success path was tested. A change set was created, and sequence logic and cut sets were generated both with and without fault trees. The results were unacceptable. The unacceptable results were generated without fault trees and IRRAS complemented the existing system cut sets incorrectly. IRRAS did not take into account the event tree success logic in generating sequence cut sets. See anomaly report AR-42-0793-001 for more specifics.

The Blank flag was tested (software test record 43) while testing the process flags in the change set features (software test records 34 through 41). The results were acceptable.

C-6. IMPORTANCE MEASURES

Six importance measures are available in SAPHIRE: Fussell-Vessely, Risk Reduction Ratio, Risk Increase Ratio, Birnbaum Importance, Risk Reduction Interval, and Risk Increase Interval. The importance measures were examined utilizing three slightly different approaches. First IRRAS calculations for the CSS system in the DEMO family were independently checked using LOTUS spreadsheet. Second, importance measures for an event in the LOSP event tree in the DEMO family were compared to importance measures calculated by hand. Lastly, IRRAS calculations using a SARA-level database of the LaSalle PRA were compared against importance measures published in the LaSalle PRA documentation. Each of these are briefly described below and are addressed by software test records 44 through 49.

Using the CSS system from the DEMO family in IRRAS, the importance measures are calculated for each of the events appearing in the CSS system cut sets. These IRRAS generated importance measures are then compared to calculations performed using a LOTUS spreadsheet. All six importance measures are included in these calculations. All agreed to three significant digits. This is judged acceptable.

Using the LOSP event tree sequence #3 in the DEMO family, importance measures were calculated by IRRAS. These same importance measures were then manually calculated for event C-MOV-1 by setting the event probability to one and then zero. The resulting sequence frequencies were then used in hand calculations. These hand calculated importance measures were then compared to those reported by IRRAS. All agreed to three significant digits. This is judged acceptable.

Using the SARA-level LaSalle database, importance measures were calculated by IRRAS for the accident sequence identified as "T100." These importance measures were then compared to those documented in the LaSalle PRA (A. C. Payne, Jr. et al., Analysis of the LaSalle Unit 2 Nuclear Power Plant: Risk Methods Integration and Evaluation Program (RMIEP), NUREG/CR-4832, Vol. 3, Part 2, August 1992). Only the Risk Reduction Interval and Risk Increase Interval are published in the LaSalle PRA, therefore, only these two measures could be compared and checked. All agreed to three significant digits. This is judged acceptable.

Appendix C

SOFTWARE TEST RECORD

STR No.   :   STR-01-000-0893
Date   :   8/4/93
Originator   :   RVH

Test Number: Test 1

Test Description: DEMO data base. Cut set probability cutoff—keep only cut sets whose probability is above the cut set probability cutoff. Using DEMO database will set probability cutoff at 0.00, 1E–20, 1E–8, 1E–5, 1E–3, 1E–1, and 1.00 and generate cut sets in the analyze fault tree module for CCS system.

Settings are: Perform cut set prob truncation—YES, (with truncation as stated); Perform event truncation—NO; Perform cut set size truncation—NO; and no starting gate entered.

This will also test STR-03-000-0893, STR-04-000-0893, STR-05-000-0893, and STR-06-000-0893.

Computer system description: IBM Clone, Gateway 2000 486/33DXC, 8Mb Ram, 200IDE hard drive, VGA display, DOS 5.0

Expected values/acceptance criteria: The CCS system was solved using Boolean logic, the cut sets produced were loaded into a LOTUS 1,2,3 spread sheet, values were added for basic events and each cut set value was generated. This is shown in printout. Visual inspection and comparison of the values generated by IRRAS and this printout was considered acceptable.

Input data file names and listings: Since DEMO database was used there are no input files.

Output data file names and listings: After each calculation was performed, a report was produced on the cut set listing in display reports for all results that produced cut sets. For those two cut offs that did not produce results the <CTRL>K command was used to produce a screen copy of the results. All results were combined into one Word Perfect file as shown.

SOFTWARE TEST RECORD

STR No.   :   STR-2-001-0793
Date   :   7/29/93
Originator   :   Steven D. Novack

Test Number: STR-2-001-07-93

Test Description: Determine the maximum number of cutsets saved in the database for a fault tree. To determine this value I created cutsets from one fault tree (IAS) taken from a Peach Bottom system with a cutoff truncation value of 1E–19. Analysis of this fault tree generated more cutsets than the code could save therefore the total number of cutsets were truncated. Next, I saved the cutsets in an ASCII report using the IRRAS report function. Then I loaded the file into MS EXCEL 4.0, smart parsed the results and wrote a macro that stripped the headers off the file and another that counted the total number of events. By adding the total number of events and cutsets I could test the formula given in vital features document.

Computer system description: IBM Clone/Continental/386-33/AT-Bus, DOS 5.0, 12Mb, 2 IDE– 100Mb, SVGA-Windows VGA-DOS

Expected values/acceptance criteria: The expected total number of events and cutsets should be less than 32,700.

Input data file names and listings: Input listings are attached (on disk).

Output data file names and listings: Output files are on disk. The results conflicted with the formula give in the technical reference guide V4.0. I found the number of total cutsets saved in the database to be < 32,000 not < 32,700. See Anomaly report AR-2-07-93-001.

SOFTWARE TEST RECORD

STR No.   :   STR-03-000-0893
Date   :   8/4/93
Originator   :   RVH

Test Number: Test 3

Test Description: DEMO data base. Correct cut set generation—when a fault tree is reduced to cut sets the correct number of cut sets that produce the top events exist. Using DEMO database will set probability cutoff at 0.00, 1E–20, 1E–8, 1E–5, 1E–3, 1E–1, and 1.00 and generate cut sets in the analyze fault tree module for CCS system.

Settings are: Perform cut set prob truncation—YES, (with truncation as stated); Perform event truncation—NO; Perform cut set size truncation—NO; and no starting gate entered.

This will also test STR-01-000-0893, STR-04-000-0893, STR-05-000-0893, and STR-06-000-0893.

Computer system description: IBM Clone, Gateway 2000 486/33DXC, 8Mb RAM, 200IDE hard drive, VGA display, DOS 5.0

Expected values/acceptance criteria: The CCS system was solved using Boolean logic, the cut sets produced were loaded into a LOTUS 1,2,3 spread sheet, values were added for basic events and each cut set value was generated. This is shown in printout. Visual inspection and comparison of the values generated by IRRAS and this printout was considered acceptable.

Input data file names and listings: Since DEMO database was used there are no input files.

Output data file names and listings: After each calculation was performed, a report was produced on the cut set listing in display reports for all results that produced cut sets. For those two cut offs that did not produce results the <CTRL>K command was used to produce a screen copy of the results. All results were combined into one Word Perfect file as shown.

SOFTWARE TEST RECORD

STR No.   :   STR-04-000-0893
Date   :   8/5/93
Originator   :   RVH

Test Number: Test 4

Test Description: DEMO data base. Generate minimal cut set upper bound for alternate cut sets of the selected systems—this is the bounding number for the sum of all the cut sets. Using DEMO database will set probability cutoff at 0.00, 1E–20, 1E–8, 1E–5, 1E–3, 1E–1, and 1.00 and generate cut sets in the analyze fault tree module for CCS system.

Settings are: perform cut set prob truncation—YES, (with truncation as stated); Perform event truncation—NO; Perform cut set size truncation—NO; and no starting gate entered.

This will also test STR-01-000-0893, STR-03-000-0893, STR-05-000-0893, and STR-06-000-0893.

Computer system description: IBM Clone, Gateway 2000 486/33DXC, 8Mb RAM, 200IDE hard drive, VGA display, DOS 5.0

Expected values/acceptance criteria: The CCS system was solved using Boolean logic, the cut sets produced were loaded into a LOTUS 1,2,3 spread sheet, values were added for basic events and each cut set value was generated. This is shown in printout. Visual inspection and comparison of the values generated by IRRAS and this printout was considered acceptable.

Input data file names and listings: Since DEMO database was used there are no input files.

Output data file names and listings: After each calculation was performed, a report was produced on the cut set listing in display reports for all results that produced cut sets. For those two cut offs that did not produce results the <CTRL>K command was used to produce a screen copy of the results. All results were combined into one Word Perfect file as shown.

Appendix C

SOFTWARE TEST RECORD

STR No.   :   STR-05-000-0893
Date   :   8/5/93
Originator   :   RVH

Test Number: Test 5

Test Description: DEMO data base. Generate and quantify the correct minimal cut set upper bound using the min-cut upper bound equation. Using DEMO database will set probability cutoff at 0.00, 1E-20, 1E-8, 1E-5, 1E-3, 1E-1, and 1.00 and generate cut sets in the analyze fault tree module for CCS system.

settings are: Perform cut set prob truncation—YES, (with truncation as stated); Perform event truncation—NO; Perform cut set size truncation—NO; and no starting gate entered.

This will also test STR-01-000-0893, STR-03-000-0893, STR-04-000-0893, and STR-06-000-0893.

Computer system description: IBM Clone, Gateway 2000 486/33DXC, 8Mb RAM, 200IDE hard drive, VGA display, DOS 5.0

Expected values/acceptance criteria: The CCS system was solved using Boolean logic, the cut sets produced were loaded into a LOTUS 1,2,3 spread sheet, values were added for basic events and each cut set value was generated. This is shown in printout. Visual inspection and comparison of the values generated by IRRAS and this printout was considered acceptable.

Input data file names and listings: Since DEMO database was used there are no input files.

Output data file names and listings: After each calculation was performed, a report was produced on the cut set listing in display reports for all results that produced cut sets. For those two cut offs that did not produce results the <CTRL>K command was used to produce a screen copy of the results. All results were combined into one Word Perfect file as shown.

SOFTWARE TEST RECORD

STR No.   :   STR-06-000-0893
Date   :   8/6/93
Originator   :   RVH

Test Number: Test 6

Test Description: DEMO data base. Fault tree created with less than 100 gates. This effort is to test the ability of SAPHIRE to solve a single fault tree. Using DEMO database will set probability cutoff at 0.00, 1E-20, 1E-8, 1E-5, 1E-3, 1E-1, and 1.00 and generate cut sets in the analyze fault tree module for CCS system.

Settings are: Perform cut set prob truncation—YES, (with truncation as stated); Perform event truncation—NO; Perform cut set size truncation—NO; and no starting gate entered.

This will also test STR-01-000-0893, STR-03-000-0893, STR-04-000-0893, and STR-05-000-0893.

Computer system description: IBM Clone, Gateway 2000 486/33DXC, 8Mb RAM, 200IDE hard drive, VGA display, DOS 5.0

Expected values/acceptance criteria: The CCS system was solved using boolean logic, the cut sets produced were loaded i: to a LOTUS 1,2,3 spread sheet, values were added for basic events and each cut set value was generated. This is shown in printout. Visual inspection and comparison of the values generated by IRRAS and this printout was considered acceptable.

Input data file names and listings: Since DEMO database was used there are no input files.

Output data file names and listings: After each calculation was performed, a report was produced on the cut set listing in display reports for all results that produced cut sets. For those two cut offs that did not produce results the <CTRL>K command was used to produce a screen copy of the results. All results were combined into one Word Perfect file as shown.

SOFTWARE TEST RECORD

STR No.	:	STR-7-001-0893
Date	:	8/12/93
Originator	:	Steven D. Novack

Test Number: STR-7-001-08-93

Test Description: Analyze a large fault tree to determine if the code correctly determines the appropriate cutsets. To accomplish this test I combined two fault trees from the Peach Bottom Database (V2 and V3) into a new tree name Test 7. Next I evaluated the new tree and printed the results. Then I down loaded the value block and tree information into a sets input file and uploaded the data into CAFTA. I evaluated the tree in CAFTA, printed and compared the results. I used a cutoff value of 1E-7 and the test tree contained 150 gates.

Computer system description: IBM Clone/Continental/386-33/AT-Bus, DOS 5.0, 12Mb, 2 IDE– 100Mb, SVGA-Windows VGA-DOS

Expected values/acceptance criteria: The cutsets for IRRAS and CAFTA should match.

Input data file names and listings: Input listings are attached (on disk). Included are reports containing the cutsets from CAFTA and IRRAS for comparison purposes.

Output data file names and listings: Output files are on disk. The results from CAFTA and IRRAS matched to two places after the decimal point in scientific notation.

SOFTWARE TEST RECORD

STR No.	:	STR-7A-001-0893
Date	:	8/16/93
Originator	:	Steven D. Novack

Test Number: STR-7A-001-08-93

Test Description: Analyze a large fault tree to determine if the code correctly determines the appropriate cutsets. To accomplish this test I combined five fault trees from the LaSalle Database (ADS,CDS,HPCS,LPCI,RCIC) into a new tree name Test 7-1. Next I evaluated the new tree and printed the results. Then I down loaded the value block and tree information into a sets input file and uploaded the data into CAFTA. I evaluated the tree in CAFTA, printed and compared the results. I used a cutoff value of 1E-7, a truncation of four events per cutset, and the test tree contained 612 gates. In addition to testing the cutsets I attempted to test the maximum number of gates allowed (2500) to analyze a fault tree.

Computer system description: IBM Clone/Continental/386-33/AT-Bus, DOS 5.0, 12Mb, 2 IDE– 100Mb, SVGA-Windows VGA-DOS

Expected values/acceptance criteria: The cutsets for IRRAS and CAFTA should match and the number of maximum gates should be 2500.

Input data file names and listings: Input listings are attached (on disk), also included are a hard copy of the cutset report.

Output data file names and listings: Output files are on disk. The results from CAFTA and IRRAS matched to two places after the decimal point in scientific notation. However the maximum number of gates was limited by DOS memory. I found that CAFTA and IRRAS reported a different number of gates per tree. CAFTA recognized 1856 gates for the Test 7-1 tree while IRRAS recognized only 612. I attempted to test different size trees and found that by changing the amount of RAM memory I could change the size of tree that could be analyzed. However, after maximizing the memory to 627K out of 640K I still was unable to analyze a tree past this point (612 gates) and received a "Too many gates in this tree!" error message in IRRAS and was told that my database needed to be recovered. I found this limitation in conflict with the documented maximum number of gates. An anomaly report will follow.

Appendix C

SOFTWARE TEST RECORD

STR No.   :   STR-09-000-0893
Date   :   8/15/93
Originator   :   RVH

Test Number: Test 9

Test Description: DEMO data base. Analyze small fault trees whose the total number of gates for each fault tree is less than 100. Fault trees should be reduced by hand. This is to test the ability of IRRAS to produce results for more than one fault tree when two or more are marked (using the <F3> key) for generate cut sets.

Using DEMO database will set probability cutoff at 0.00, 1E–8, 1E–5, 1E–3, 1E—1, and 1.00, mark and generate cut sets in the analyze fault tree module both the ECS and CCS systems.

Settings are: Perform cut set prob truncation—YES (with truncation as stated); Perform event truncation—NO; Perform cut set size truncation—NO; and no starting gate entered.

This will also test STR-01-000-0893, STR-03-000-0893, STR-04-000-0893, and STR-05-000-0893.

Computer system description: IBM Clone, Gateway 2000 486/33DXC, 8Mb RAM, 200IDE hard drive, VGA display, DOS 5.0

Expected values/acceptance criteria: The ECS and CCS system were solved using Boolean logic, the cut sets produced were loaded into a LOTUS 1,2,3 spread sheet, values were added for basic events and each cut set value was generated. This is shown in printout. Visual inspection and comparison of the values generated by IRRAS and this printout was considered acceptable.

Input data file names and listings: Since DEMO database was used there are no input files.

Output data file names and listings: After each calculation was performed, a report was produced from the report option for all results that produced cut sets. All results were combined into one Word Perfect file as attached.

SOFTWARE TEST RECORD

STR No.   :   STR-9-001-0793
Date   :   7/29/93
Originator   :   Steven D. Novack

Test Number: STR-9-001-09-93

Test Description: Analyze a large and small fault tree to determine if the code correctly calculates the cutset probability. To accomplish this test I used a small tree (CSS) from the DEMO database and a large fault tree (IAS) from the Peach Bottom database. I extracted the basecase basic event table that includes the event names and probabilities, and the cutset report that includes the cutset members and the calculated probability. Next, I loaded the files into MS EXCEL V4.0 used a previously generated macro to strip the headings from the file and created a second macro to calculate each cutset probability.

Computer system description: IBM Clone/Continental/386-33/AT-Bus, DOS 5.0, 12Mb, 2 IDE– 100Mb, SVGA-Windows VGA-DOS

Expected values/acceptance criteria: The expected cutset probabilities in IRRAS should match the calculated values in EXCEL.

Input data file names and listings: Input listings are attached (on disk). Also included are a hard copy of the Excel macro used to determine the cutset probability.

Output data file names and listings: Output files are on disk. The results matched the IRRAS file for both the large and small fault trees.

SOFTWARE TEST RECORD

STR No.   :   STR-10-001-0793
Date   :   7/26/93
Originator   :   LW

Test Number: Test 10

Test Description: VV data base—1) Generate the possible sequences for an event tree and all logic for each sequence 2) All sequences have logic generated except (a) those with "OK" "SUCCESS", or "IGNORE" as end states and (b) those whose end states begin with the character "@"; 3) Add mutually exclusive top (if non-blank top exist) as a successful top to each sequence logic; 4) Name of list file—output the logic generated in a report form to the screen, printer, or user named file; 5) Numbers—use sequence numbers as the name of each generated sequence; 6) Names—use sequence name contained in the event tree as the name of each generated sequence

Computer system description: IBM Clone, Insight 486, 33 MHz, ISA bus, DOS 5.0, 16 Mb RAM, 340 Mb IDE hard drive, VGA display

Expected values/acceptance criteria: 1) Generated all sequence logic for the VV data base event tree—acceptable; 2) "@" placed on seq 1—sequence not generated; ignore placed on seq 2—sequence not generated; success placed on seq 3—sequence generated—acceptable; 3) Mutually exclusive top substitution added to all sequences of the event tree—acceptable; 4) Logic output to screen, printer and file—acceptable; 5) Sequence numbers used as sequence names—acceptable; 6) Sequence name on event tree as sequence name—acceptable

Input data file names and listings: See attached

Output data file names and listings: See attached

SOFTWARE TEST RECORD

STR No.   :   STR-11-001-0793
Date   :   7/26/93
Originator   :   LW

Test Number: Test 11

Test Description: VV data base—1) Frequency—look up minimum cut set upper bound for marked/highlighted sequences (0.0 Is default)—add frequency to the event tree drawing—changed header for the third column to be frequency; 2) '@' status—change the end state name to be generated (without '@' in the first character) or not generated (with '@' in the first character) for each; 3) Line edit—change any highlighted line; 4) Follow transfer—if transfer event tree exists, edit its logic; 5) Global replace—given the column number and a search string, replace any occurrences of the search string in the given column number with the replace string; 6) Header edit—change the header for the drawing; 7) Change transfer—toggle logic between being a transfer path or a termination path

Computer system description: IBM Clone, Insight 486, 33 MHz, ISA bus, DOS 5.0, 16 Mb RAM, 340 Mb IDE hard drive, VGA display

Expected values/acceptance criteria: 1) Frequency added to event tree sequence—acceptable; 2) @ added to each sequence—sequences not generated—acceptable; 3) Line edit—changed highlighted lines—acceptable; 4) Follow transfer—edit logic—acceptable; 5) Global replace—replace all occurrences of a search string in a column—acceptable; 6) Header edit—event tree header edited to turn on and off header columns—acceptable; 7) Change transfer—transfer toggled on and off, sequence transfers with transfer on and no transfer with transfer off—acceptable

Input data file names and listings: See attached

Output data file names and listings: See attached

NUREG/CR-6145

Appendix C

SOFTWARE TEST RECORD

STR No.   :   STR-12-001-0893
Date   :   8/4/93
Originator   :   LW

Test Number: Test 12

Test Description: VV data base—1A) Multiple branches—if more than one failed branch exists and an exception applies to at least one of them, then all branches must have a replaced top and replacement top. Use event tree ET-1. 1B) in the sequence editor place an "@" on the end state of seq 3 on event tree IE-1. Generate sequence logic.

Rule—a rule of replacement contains a number of conditional tops, tops to be replaced and tops to replace.

2) Add– add an entry at the highlighted cell.
3) Modify– (insert)—add a blank line above the highlighted cell.
4) Delete– delete an entry at the highlighted cell.
5) Restore– last deleted entry at the highlighted cell.
6) Copy– make a copy of an existing rule at user specified location.

Computer system description: IBM Clone, Insight 486, 33 MHz, ISA bus, DOS 5.0, 16 Mb RAM, 340 Mb IDE hard drive, VGA display

Expected values/acceptance criteria: VV data base—link editor—allows user to enter links (exceptions) that are to be applied during sequence logic generation.

1A) Top event substitutions should appear in the sequence logic report. Seq 5 of event tree IE-1 should have a substitution of C1 for C. Unacceptable. See AR-12-0893-001

1B) Top event substitutions should appear in the sequence logic report. Seq 3 should not appear in the logic report (correct) and top event B1 should be substituted for top event B in sequences 4 and 5. Unacceptable. See AR-12-0893-002 Rule—a rule of replacement contains a number of conditional tops, tops to be replaced and tops to replace. 2) Add—add an entry at the highlighted cell. Acceptable. 3) Modify is not on the IRRAS 4.16 Link editor menu. Insert is. Insert—add a blank line above the highlighted cell. Acceptable. 4) Delete—delete an entry at the highlighted cell. Acceptable. 5) Restore— last deleted entry at the highlighted cell. Acceptable. 6) Copy—make a copy of an existing rule at user specified location. Acceptable.

Input data file names and listings: See attached

Output data file names and listings: See attached

SOFTWARE TEST RECORD

STR No.   :   STR-13-001-0893
Date   :   8/9/93
Originator   :   Steven D. Novack

Test Number: STR-13-001-08-93

Test Description: Event Probability cutoff value. Do not discard cut set if cut set failure probability is greater than the cutoff or failure probability for each event in the cutset is greater than or equal to the event probability cutoff. To accomplish this Test I used the Peach Bottom sequence A17 and generated cut sets for 4 different probability cutoff values (1E-2, 1E-9, 1E-11, 5E-12). This test was repeated for cut sets using fault trees and then sequences to generate the cutsets.

Computer system description: IBM Clone/Continental/386-33/AT-Bus, DOS 5.0, 12Mb, 2 IDE- 100Mb, SVGA-Windows VGA-DOS

Expected values/acceptance criteria: The expected values would be consistent with the probability cutoff values in the reports.

Input data file names and listings: Input listings are attached (on disk). Also included are a hard copy of the reports.

Output data file names and listings: Output files are on disk and on hard copy. The results matched the appropriate cutoff probabilities in all cases.

SOFTWARE TEST RECORD

STR No.   :   STR-16-001-0693
Date   :   7/20/93
Originator   :   Steven D. Novack

Test Number: STR-16-001-07-93

Test Description: Determine the maximum number of cut sets saved in the database for one sequence. To determine this value I created cut sets from one sequence taken from a Peach Bottom event tree. The sequence generated more than the saved number of cut sets therefore the total number of cut sets were truncated. Next, I saved the cut sets in an ASCII report using the IRRAS report function, loaded the file into Word Perfect for Windows and split the file in half. Then I loaded the two files into MS EXCEL 4.0, smart parsed the results and wrote a macro that stripped the headers off the file and another that counted the total number of events.

Computer system description: IBM Clone/Continental/386-33/AT-Bus, DOS 5.0, 12Mb, 2 IDE- 100Mb, SVGA-Windows VGA-DOS

Expected values/acceptance criteria: The expected total number of events and cutsets should be less than 32,700.

Input data file names and listings: Input listings are attached (on disk).

Output data file names and listings: Output files are on disk. The results conflicted with the formula give in the technical reference guide V4.0. I found the number of total cut sets saved in the database to be < 32,000 not < 32,700. An anomaly report AR-02-07-93-001.

Appendix C

SOFTWARE TEST RECORD

STR No.   :   STR-17-001-0893
Date   :   8/6/93
Originator   :   LW

Test Number: Test 17

Test Description: VV data base
1A) Event tree ET: create three (3) chg sets: flagset1, flagset2, flagset3 for sequences 2, 3, and 4, respectively. Enter flagset names in "flag set name" block of the event tree modify sequence menu.

   Seq 2—Flagset1: System A, Y flag set, basic event L, house event TRUE—L removed (input to AND gate)

   Seq 3—Flagset2: System B, Y flag set, basic event J, I (ignore J)

   Seq 4—Flagset3: Basic event J, house event FALSE—J removed (input to OR gate)

Generate sequence cut sets on each sequence individually and as a group, no cut offs with fault trees

1B)   Seq 4—Flagset3: basic event J, house event TRUE—J and K removed (input to OR gate)

Generate sequence cut sets on sequence 4, no cut offs with fault trees

1C)   Seq 3—Flagset2: Basic event L, house event FALSE—L and M removed (input to AND gate)

Generate sequence cut sets on sequence 4, no cut offs with fault trees

2) Event tree ET: create three (3) chg sets: flagset1, flagset2, flagset3 for sequences 2, 3, and 4, respectively. Enter flagset names in "flag set name" block of the event tree modify sequence menu.

   Seq 2—Flagset1: Basic event L, house event TRUE—L removed (input to AND gate)

   Seq 3—Flagset2: System B, Y flag set, basic event J, I (ignore J)

   Seq 4—Flagset3: Basic event J, house event FALSE—J removed (input to OR gate)

Generate sequence cut sets on each sequence, no cut offs with fault trees and insert default1 for flag set name in the cut set generation cutoff values menu for sequence 2 cut set generation. Do the same for default2 on sequence 3 and default3 on sequence 4.

3) Using the data in number 2 above, generate sequence cut sets on each sequence, no cut offs with fault trees and insert 'none' for the flag set name in the cut set generation cutoff values menu.

Computer system description: IBM Clone, Insight 486, 33 MHz, ISA bus, DOS 5.0, 16 Mb ram, 340 Mb IDE hard drive, VGA display

Expected values/acceptance criteria:

1A)   Seq 2—Flagset1: System A, Y flag set, basic event L, house event TRUE -L removed (input to AND gate)

   Seq 3—Flagset2: System B, Y flag set, basic event J, I (ignore J)

   Seq 4—Flagset3: Basic event J, house event FALSE—J removed (input to OR gate)

Generate sequence cut sets, no cut offs with fault trees—acceptable

1B)   Seq 4 Flagset3: Basic event J, house event TRUE—J and K removed (input to OR gate)

Generate sequence cut sets on sequence 4, no cut offs with fault trees—acceptable

NUREG/CR-6145   C-20

1C)   Seq 3 Flagset2: Basic event L, house event FALSE—L and M removed (input to AND gate)

Generate sequence cut sets on sequence 4, no cut offs with fault trees—acceptable

2) Event tree ET: create three (3) chg sets: flagset1, flagset2, flagset3 for sequences 2, 3, and 4, respectively. Enter flagset names in "flag set name" block of the event tree modify sequence menu.

Seq 2—Flagset1: Basic event L, house event TRUE—L removed (input to AND gate)—acceptable

Seq 3—Flagset2: System B, Y flag set, basic event J, I (ignore J)—acceptable

Seq 4—Flagset3: Basic event J, house event FALSE—J removed (input to OR gate)—acceptable

3) Using the data in number 2 above, generate sequence cut sets on each sequence, no cut offs with fault trees and insert "NONE" for the flag set name in the cut set generation cutoff values menu. Sequence cut sets will be generated without any change sets. Acceptable.

Input data file names and listings: See attached

Output data file names and listings: See attached

Appendix C

SOFTWARE TEST RECORD

STR No. : STR-18-000-0893
Date : 8/15/93
Originator : RVH

Test Number: Test 18

Test Description: DEMO data base. Update cut sets. Reevaluate existing cut sets using the cut set update feature in IRRAS with both the base and current cases. This will test the following parameters.

A) When the cut set probability cutoff is set, IRRAS discards any cut set whose probability of failure is less than the cutoff value, and that the probability of failure is equal to the product of the probability of each event in the cut set.

B) That irras uses the size truncation option as intended and eliminates any cut set whose size exceeds a specified size.

C) That IRRAS uses the zone truncation option as intended and eliminates any cut set whose number of zone events exceeds the specified zone size.

Using DEMO database will generate both base and current case cut sets for sequence 2, setting all cutoff values to no <N>.

Will the cut set update option for both the current and base case, a) will set cutoffs at combinations of 0.00, 1E–8, 1E–5, 1E–3, and 1E–1; b) will set cutoff for cutset size of 0, 1, 2, and 3.)

The cut set update menu does not provide this option in IRRAS 4.0. It is provided in IRRAS 5.0.

Computer system description: IBM Clone, Gateway 2000 486/33DXC, 8Mb RAM, 200IDE hard drive, VGA display, DOS 5.0

Expected values/acceptance criteria: Sequence 2 was solved using Boolean logic, the cut sets produced were loaded into a LOTUS 1,2,3 spread sheet, values were added for basic events and each cut set value was generated. This is shown in printout. Visual inspection and comparison of the values generated by IRRAS and this printout was considered acceptable.

Input data file names and listings: Since DEMO database was used there are no input files.

Output data file names and listings: After each calculation was performed, a report was produced on the cut set listing in display reports for all results that produced cut sets.

SOFTWARE TEST RECORD

STR No.   :   STR-19-001-0893
Date   :   8/2/93
Originator   :   Steven D. Novack

Test Number: STR-19-001-08-93

Test Description: Analyze a large sequence, not using the fault tree option to determine if the code correctly calculates the cut set probability. To accomplish this Test I used a sequence (A17) from the Peach Bottom database with a probability cutoff value of $1 \times 10E-19$. I used the same basic event table as test #9 that includes the event names and probabilities, and generated a new cut set report that includes the cut set members and the calculated probability. Next, I loaded the files into MS EXCEL V4.0 used a previously generated macro to strip the headings from the file and a second macro to calculate each cut set probability.

Computer system description: IBM Clone/Continental/386-33/AT-Bus, DOS 5.0, 12Mb, 2 IDE—100Mb, SVGA-Windows VGA-DOS

Expected values/acceptance criteria: The expected cutset probabilities in IRRAS should match the calculated values in EXCEL.

Input data file names and listings: Input listings are attached (on disk). Also included are a hard copy of the Excel macro used to determine the cutset probability.

Output data file names and listings: Output files are on disk. The results did not match the IRRAS file for the A17 sequence. The EXCEL values were off by a factor of 1E–4. This is probably due to the initiating event, which is not included in the cutsets and therefore not calculated by the EXCEL macro. Although I will not write an Anomaly for this item it should be noted.

SOFTWARE TEST RECORD

STR No.   :   STR-21-06-93-001
Date   :   6/14/93
Originator   :   Steven D. Novack

Test Number: STR-21-06-93-001

Test Description: Spilt Fraction calculation. Values from two systems were generated using IRRAS. These were then combined using the equation provided in the test protocol to calculate a sequence split fraction by hand. Next the hand calculated values were compared to the IRRAS report values.

Computer system description: IBM Clone/Continental/386-33/AT-Bus, DOS 5.0, 12Mb, 2 IDE—100Mb, SVGA-Windows VGA-DOS

Expected values/acceptance criteria: Demo database used Sequence 2 = 4.77263E–2
Sequence 3 = 1.03371E–3. (see attached)

Output data file names and listings: Input values are derived from the cutset upper bound results of the CCS and ECS trees (see attached). LaSalle database.

Output data file names and listings: The output data is found on the split fraction results screen. The results were consistent with the hand calculated values. However, there may be a problem with not rounding the results of the Split Fraction.

Appendix C

SOFTWARE TEST RECORD

STR No.	:	STR-22-001-06-93
Date	:	6/15/93
Originator	:	Steven D. Novack

Test Number: STR-22-001-06-93

Test Description: Base Case Update. An Alternate case was made that changed 3 basic event probability data. Cut sets for the alternative case and the base case were run. Report outputs of the alternative case, the base case prior to the update and after the update were made for comparison purposes.

Computer system description: IBM Clone/Continental/386-33/AT-Bus, DOS 5.0, 12Mb, 2 IDE—100Mb, SVGA-Windows VGA-DOS

Expected values/acceptance criteria: Demo database used (CCS) C-MOV-1 Base = 1E-3 Alternate = 1E-4 E-MOV-1 Base = 1E-3 Alternate = 1E-4 Tank Base = 1E-7 Alternate 1E-3 (See attached)

Input data file names and listings: Input listings are attached.

Output data file names and listings: The output data is included with the attached listings. This test was never completed since the base case report failed to show the proper base case probabilities. Print screens of the basic event data were included (see attached) to confirm base case event data which did not match the base case probabilities in the cut sets. It seems that the alternate probabilities as opposed to the base case data is appearing in the reports for the base case. This feature needs to be corrected before finalizing this test.

SOFTWARE TEST RECORD

STR No.	:	STR-23-001-06-93
Date	:	6/15/93
Originator	:	Steven D. Novack

Test Number: STR-23-001-06-93

Test Description: Minimal Cut Set Upper Bound. The test procedure included downloading the cut sets and their probabilities (sequence 2 & 3 from the DEMO family) into a text file. The text file was then uploaded into MS Excel 3.0, and smart parsed. Once in a workable form, calculations based on the test procedure for determining the minimal cut set upper bound was applied to the data.

Computer system description: IBM Clone/Continental/386-33/AT-Bus, DOS 5.0, 12Mb, 2 IDE—100Mb, SVGA-Windows VGA-DOS

Expected values/acceptance criteria: DEMO database used (CCS) Sequence 2 MCSUB = 4.63633404E-1
Sequence 3 MCSUB = 3.96970739644809E-3 (see attached)

Input data file names and listings: Input listings are attached. IRRAS DEMO Database.

Output data file names and listings: The output data is included with the attached listings. All values checked out for two significant figures after the decimal point using scientific notation. However, there is still a problem with rounding (or not rounding values). See STR # 21 for more detail. In this computation sequence 2 was off in the third value after the decimal place.

SOFTWARE TEST RECORD

STR No.   :   STR-24-001-06-93
Date   :   6/29/93
Originator   :   Steven D. Novack

Test Number: STR-24-001-06-93

Test Description: Perform Analysis on event tree using fault trees. To accomplish this task I extracted two systems from the LaSalle-4 database in IRRAS, created a small event tree that included these systems and generated cut sets for two sequences. Next I downloaded the fault tree information and uploaded it into CAFTA. Then I created the same event tree and generated cut sets for the same two sequences.

Computer system description: IBM Clone/Continental/386-33/AT-Bus, DOS 5.0, 12Mb, 2 IDE—100Mb, SVGA-Windows VGA-DOS

Expected values/acceptance criteria: Based on the CAFTA results Sequence 2 = 2.54E–01
Sequence 3 = 9.73E–0 (see attached)

Input data file names and listings: Input listings are attached. LaSalle Database.

Output data file names and listings: The output data is included with the attached listings for cutsets greater than 1 x 10-6. All values and the top twenty cutsets checked out.

SOFTWARE TEST RECORD

STR No.   :   STR-25-001-06-93
Date   :   7/1/93
Originator   :   Steven D. Novack

Test Number: STR-25-001-06-93

Test Description: Perform Analysis on event tree using cut sets. To accomplish this task I extracted two systems from the LaSalle-4 database in IRRAS, created a small event tree that included these systems and generated cut sets for one sequence. Next, I downloaded the fault tree information and uploaded it into CAFTA. Then I created the same event tree and generated cut sets for the same sequence.

Computer system description: IBM Clone/Continental/386-33/AT-Bus, DOS 5.0, 12Mb, 2 IDE—100Mb, SVGA-Windows VGA-DOS

Expected values/acceptance criteria: Based on the CAFTA results Sequence 4 = 3.3E–02 (see attached)
Input data file names and listings: Input listings are attached. LaSalle. VBK RCIC.SET

Output data file names and listings: The output data is included with the attached listings for cutsets greater than 1 x 10-5. All values and the top twenty cut sets checked out.

Appendix C

SOFTWARE TEST RECORD

STR No. : STR-27-001-0893
Date : August 1993
Originator : Curtis L. Smith

Test Number: 001

Test Description: Tested the event tree analysis generation of cutsets for five event trees linked together using the top event system fault trees (i.e., using the fault tree logic). The SAPHIRE code passed the acceptance criteria for this test.

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: See attachments for specifics on the event trees and systems that were used to test this feature.

Input data file names and listings: See attachments.

Output data file names and listings: See attachments.

SOFTWARE TEST RECORD

STR No. : STR-29-001-0893
Date : August 1993
Originator : Curtis L. Smith

Test Number: 001

Test Description: Tested the event tree analysis generation of cutsets for five event trees linked together using the system cutsets (i.e., not by using the system fault tree logic). The SAPHIRE code passed the acceptance criteria for this test.

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: See attachments for specifics on the event trees and systems that were used to test this feature.

Input data file names and listings: See attachments.

Output data file names and listings: See attachments.

SOFTWARE TEST RECORD

STR No.	:	STR-30-001-0893
Date	:	August 1993
Originator	:	Curtis L. Smith

Test Number: 001

Test Description: Tested the Monte Carlo sampling for a single event (seed = 4321, 5000 samples). The event tested was for the uniform distribution. Three different uniform distributions were tested: (A) low value = 0, high value = 1; (B) low value = 0, high value = 100,000; and (C) low value = -1, high value = 1. The Monte Carlo results for these three distributions were compared to hand calculated values. The code passed all tests for the uniform distribution Monte Carlo analysis.

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: The expected values for the three distributions are:

Parameter	Distribution A	Distribution B	Distribution C
Mean	0.5	50,000	0
5th percentile	0.05	5,000	-0.1
50th percentile	0.5	50,000	0
95th percentile	0.95	95,000	0.9
Standard Dev.	0.2887	28,868	0.5774
Skewness	0	0	0
Kurtosis	1.8	1.8	1.8

Input data file names and listings: One basic event was input to an OR gate. The basic event had a uniform distribution with the following parameters:

Parameter	Distribution A	Distribution B	Distribution C
Prob. value	0.5	50,000	0
Uncertainty value	1	100,000	1
Calc. type	1	1	1
Uncertainty type	U	U	U
Init. Flag	Y	Y	Y

Output data file names and listings: See attachments.

Appendix C

SOFTWARE TEST RECORD

STR No.   :   STR-31-001-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Test Number: 001

Test Description: Tested the Latin Hypercube (LHS) sampling for a single event (seed = 4321, 5000 samples). The event tested was for the uniform distribution. Three different uniform distributions were tested: (A) low value = 0, high value = 1; (B) low value = 0, high value = 100,000; and (C) low value = -1, high value = 1. The LHS results for these three distributions were compared to hand calculated values. The code passed all tests for the uniform distribution LHS analysis.

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: The expected values for the three distributions are:

Parameter	Distribution A	Distribution B	Distribution C
Mean	0.5	50,000	0
5th percentile	0.05	5,000	-0.1
50th percentile	0.5	50,000	0
95th percentile	0.95	95,000	0.9
Standard Dev.	0.2887	28,868	0.5774
Skewness	0	0	0
Kurtosis	1.8	1.8	1.8

Input data file names and listings: One basic event was input to an OR gate. The basic event had a uniform distribution with the following parameters:

Parameter	Distribution A	Distribution B	Distribution C
Prob. value	0.5	50,000	0
Uncertainty value	1	100,000	1
Calc. type	1	1	1
Uncertainty type	U	U	U
Init. Flag	Y	Y	Y

Output data file names and listings: See attachments.

NUREG/CR-6145

SOFTWARE TEST RECORD

STR No.   :   STR-30-002-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Test Number: 002

Test Description: Tested the Monte Carlo (MC) sampling for a single event (seed = 4321, 5000 samples). The event tested was for the exponential distribution. Eight different exponential distributions were tested: (A) prob. value = -1, (B) prob. value = 0, (C) prob. value = 0.1, (D) prob. value = 1, (E) prob. value = 50, (F) prob. value = 5,000, (G) prob. value = 0.0002, and (H) prob. value = 1E-16. The MC results for these distributions were compared to hand calculated values. The code <u>did not</u> pass all tests for the exponential distribution MC analysis (see reports AR-30-0893-001 and AR-30-0893-002).

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: The expected values for the eight distributions are:

Parameter	A	B	C	D	E	F	G	H
Mean	-1	0	0.1	1	50	5,000	2E-4	1.443E-16
5th percentile	n/a	0	5.13E-3	5.13E-2	2.565	256.5	1E-5	7.4E-18
50th percentile	n/a	0	6.93E-2	6.93E-1	34.65	3,465	1.39E-4	1.0E-16
95th percentile	n/a	0	2.996E-1	2.996	149.8	14,980	5.99E-4	4.3E-16
Standard Dev.	n/a	0	0.1	1	50	5,000	2E-4	1.443E-16
Skewness	n/a	0	2	2	2	2	2	2
Kurtosis	n/a	0	9	9	9	9	9	9

Tests for distribution A, B, and H did not pass the acceptance criteria (see reports AR-30-0893-001 and AR-30-0893-002).

Input data file names and listings: One basic event was input to an OR gate. The basic event had an exponential distribution with the following parameters:

Parameter	A	B	C	D	E	F	G	H
Prob. value	-1	0	0	1	50	5,000	2E-4	1.443E-16
Uncertainty value	0	0	0	0	0	0	0	0
Calc. type	1	1	1	1	1	1	1	1
Uncertainty type	E	E	E	E	E	E	E	E
Init. Flag	Y	Y	Y	Y	Y	Y	Y	Y

Output data file names and listings: See attachments.

Appendix C

SOFTWARE TEST RECORD

STR No.   :   STR-31-002-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Test Number: 002

Test Description: Tested the Latin Hypercube sampling (LHS) for a single event (seed = 4321, 5000 samples). The event tested was for the exponential distribution. Eight different exponential distributions were tested: (A) prob. value = -1, (B) prob. value = 0, (C) prob. value = 0.1, (D) prob. value = 1, (E) prob. value = 50, (F) prob. value = 5,000, (G) prob. value = 0.0002, and (H) prob. value = 1E-16. The LHS results for these distributions were compared to hand calculated values. The code did not pass all tests for the exponential distribution LHS analysis (see reports AR-30-0893-001, AR-30-0893-002, and AR-31-0893-001).

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: The expected values for the eight distributions are:

Parameter	A	B	C	D	E	F	G	H
Mean	-1	0	0.1	1	50	5,000	2E-4	1.443E-16
5th percentile	n/a	0	5.13E-3	5.13E-2	2.565	256.5	1E-5	7.4E-18
50th percentile	n/a	0	6.93E-2	6.93E-1	34.65	3,465	1.39E-4	1.0E-16
95th percentile	n/a	0	2.996E-1	2.996	149.8	14,980	5.99E-4	4.3E-16
Standard Dev.	n/a	0	0.1	1	50	5,000	2E-4	1.443E-16
Skewness	n/a	0	2	2	2	2	2	2
Kurtosis	n/a	0	9	9	9	9	9	9

Tests for distribution A, B, and H did not pass the acceptance criteria (see reports AR-30-0893-001 and AR-30-0893-002). For the remaining distributions, the results were acceptable but differed from what was expected (see AR-31-0893-001).

Input data file names and listings: One basic event was input to an OR gate. The basic event had an exponential distribution with the following parameters:

Parameter	A	B	C	D	E	F	G	H
Prob. value	-1	0	0	1	50	5,000	2E-4	1.443E-16
Uncertainty value	0	0	0	0	0	0	0	0
Calc. type	1	1	1	1	1	1	1	1
Uncertainty type	E	E	E	E	E	E	E	E
Init. Flag	Y	Y	Y	Y	Y	Y	Y	Y

Output data file names and listings: See attachments.

SOFTWARE TEST RECORD

STR No.   :   STR-30-003-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Test Number: 003

Test Description: Tested the Monte Carlo (MC) sampling for a single event (seed = 4321, 5000 samples). The event tested was for the lognormal distribution. Five different lognormal distributions were tested: (A) prob. value = 1E-4, Error Factor (EF) = 10; (B) prob. value = 1E-6, EF = 10; (C) prob. value = 1E-15, EF = 1,000 (D) prob. value = 1E-16, EF = 1,000; and (E) prob. value = 1E-17, EF = 1,000. The MC results for these distributions were compared to hand calculated values. The code <u>did not</u> pass all tests for the lognormal distribution MC analysis (see report AR-30-0893-003).

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: The expected values for the five distributions are:

Parameter	A	B	C	D	E
Mean	1E-4	1E-6	1E-15	1E-16	1E-17
5th percentile	3.75E-6	3.75E-8	1.48E-22	1.48E-23	1.48E-24
50th percentile	3.75E-5	3.75E-7	1.48E-19	1.48E-20	1.48E-21
95th percentile	3.75E-4	3.75E-6	1.48E-16	1.48E-17	1.48E-18
Standard Dev.	2.47E-4	2.47E-6	6.75E-12	6.75E-13	6.75E-14

Tests for distributions C, D, and E did not pa_s the acceptance criteria (see report AR-30-0893-003).

Input data file names and listings: One basic event was input to an OR gate. The basic event had an lognormal distribution with the following parameters:

Parameter	A	B	C	D	E
Prob. value	1E-4	1E-6	1E-15	1E-16	1E-17
Uncertainty value	10	10	1,000	1,000	1,000
Calc. type	1	1	1	1	1
Uncertainty type	L	L	L	L	L
Init. Flag	Y	Y	Y	Y	Y

Output data file names and listings: See attachments.

Appendix C

SOFTWARE TEST RECORD

STR No.   :   STR-31-003-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Test Number: 003

Test Description: Tested the Latin Hypercube sampling (LHS) for a single event (seed = 4321, 5000 samples). The event tested was for the lognormal distribution. Five different lognormal distributions were tested: (A) prob. value = 1E-4, Error Factor (EF) = 10; (B) prob. value = 1E-6, EF = 10; (C) prob. value = 1E-15, EF = 1,000 (D) prob. value = 1E-16, EF = 1,000; and (E) prob. value = 1E-17, EF = 1,000. The LHS results for these distributions were compared to hand calculated values. The code did not pass all tests for the lognormal distribution LHS analysis (see report AR-30-0893-003).

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: The expected values for the five distributions are:

Parameter	A	B	C	D	E
Mean	1E-4	1E-6	1E-15	1E-16	1E-17
5th percentile	3.75E-6	3.75E-8	1.48E-22	1.48E-23	1.48E-24
50th percentile	3.75E-5	3.75E-7	1.48E-19	1.48E-20	1.48E-21
95th percentile	3.75E-4	3.75E-6	1.48E-16	1.48E-17	1.48E-18
Standard Dev.	2.47E-4	2.47E-6	6.75E-12	6.75E-13	6.75E-14

Tests for distributions C, D, and E did not pass the acceptance criteria (see report AR-30-0893-003).

Input data file names and listings: One basic event was input to an OR gate. The basic event had an lognormal distribution with the following parameters:

Parameter	A	B	C	D	E
Prob. value	1E-4	1E-6	1E-15	1E-16	1E-17
Uncertainty value	10	10	1,000	1,000	1,000
Calc. type	1	1	1	1	1
Uncertainty type	L	L	L	L	L
Init. Flag	Y	Y	Y	Y	Y

Output data file names and listings: See attachments.

SOFTWARE TEST RECORD

STR No.   :   STR-30-004-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Test Number: 004

Test Description: Tested the Monte Carlo (MC) sampling for a single event. The event tested was for the lognormal distribution. A single lognormal distribution was tested with prob. value = 1E–3 and Error Factor (EF) = 5. This event was tested for several different sample sizes (500, 1000, 3000, 5000, and 10000 samples) and several different seed values (512, 4321, and 65533). The MC results for these distributions were compared to hand calculated values. The code passed all tests for this lognormal distribution MC analysis.

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: The expected values for the distribution are:

Parameter	Distribution
Mean	1E–3
5th percentile	1.24E–4
50th percentile	6.20E–4
95th percentile	3.10E–3
Standard Dev.	1.27E–3

The skewness and kurtosis for the results were not checked against hand calculated results since these estimators varied depending on the sample size. This change in skewness and kurtosis is probably more dependent on the statistical estimators (which are poor in IRRAS 4.16) rather than the random sampling. A graph is attached showing the variability of the distribution parameters for different sample sizes.

Input data file names and listings: One basic event was input to an OR gate. The basic event had an lognormal distribution with the following parameters:

Parameter	Distribution
Prob. value	1E–3
Uncertainty value	5
Calc. type	1
Uncertainty type	L
Init. Flag	Y

Output data file names and listings: See attachments.

Appendix C

SOFTWARE TEST RECORD

STR No.   :   STR-31-004-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Test Number: 004

Test Description: Tested the Latin Hypercube sampling (LHS) for a single event. The event tested was for the lognormal distribution. A single lognormal distribution was tested with prob. value = 1E–3 and Error Factor (EF) = 5. This event was tested for several different sample sizes (100, 500, 1000, 3000, 5000, and 10000 samples) and several different seed values (512, 4321, and 65533). The LHS results for these distributions were compared to hand calculated values. The code passed all tests for this lognormal distribution LHS analysis.

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: The expected values for the distribution are:

Parameter	Distribution
Mean	1E–3
5th percentile	1.24E–4
50th percentile	6.20E–4
95th percentile	3.10E–3
Standard Dev.	1.27E–3

The skewness and kurtosis for the results were not checked against hand calculated results since these estimators varied depending on the sample size. This change in skewness and kurtosis is probably more dependent on the statistical estimators (which are poor in IRRAS 4.16) rather than the random sampling.

Input data file names and listings: One basic event was input to an OR gate. The basic event had an lognormal distribution with the following parameters:

Parameter	Distribution
Prob. value	1E–3
Uncertainty value	5
Calc. type	1
Uncertainty type	L
Init. Flag	Y

Output data file names and listings: See attachments.

SOFTWARE TEST RECORD

STR No.   :   STR-30-005-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Test Number: 005

Test Description: Tested the Monte Carlo (MC) sampling for a single event (seed = 4321, 5000 samples). The event tested was for the chi-square distribution. Eight chi-square distributions were tested: (A) prob. value = 2, uncertainty value = 1; (B) prob. value = 5, uncertainty value = 1; (C) prob. value = 4, uncertainty value = 2; (D) prob. value = 0.1, uncertainty value = 5; (E) prob. value = 400000, uncertainty value = 5; (F) prob. value = 100, uncertainty value = 100; (G) prob. value = 0, uncertainty value = 0; (H) prob. value = 0, uncertainty value = 1.5; and (I) prob. value = 0, uncertainty value = 1. The MC results for these distributions were compared to hand calculated values. The code did not pass all tests for this chi-square distribution MC analysis.

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: The expected values for the distribution were based upon the following criteria (according to the SAPHIRE Technical Reference Manual IRRAS/SARA 4.0, page 69):

Distribution A should be distributed as $2\chi^2(1)$
Distribution B should be distributed as $5\chi^2(1)$
Distribution C should be distributed as $2\chi^2(2)$
Distribution D should be distributed as $(1/50)\chi^2(5)$
Distribution E should be distributed as $8000\chi^2(5)$
Distribution F should be distributed as $\chi^2(100)$
Distribution G should be undefined
Distribution H should be undefined
Distribution I should be zero

The output from the MC were compared to a standard table of chi-square values. Tests for distributions A, B, C, D, E, and I did not pass the acceptance criteria (see report AR-30-0893-004). Tests for distribution G and H did not pass the acceptance criteria (see report AR-30-0893-005).

Input data file names and listings: One basic event was input to an OR gate. The basic event had an chi-square distribution with the following parameters:

Parameter	A	B	C	D	E	F	G	H	I
Prob. value	2	5	4	0.1	40,000	100	0	0	0
Uncertainty value	1	1	2	5	5	100	0	1.5	1
Calc. type	1	1	1	1	1	1	1	1	1
Uncertainty type	C	C	C	C	C	C	C	C	C
Init. Flag	Y	Y	Y	Y	Y	Y	Y	Y	Y

Output data file names and listings: See attachments.

Appendix C

SOFTWARE TEST RECORD

STR No.   :   STR-31-005-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Test Number: 005

Test Description: Tested the Latin Hypercube sampling (LHS) for a single event (seed = 4321, 5000 samples). The event tested was for the chi-square distribution. Eight chi-square distributions were tested: (A) prob. value = 2, uncertainty value = 1; (B) prob. value = 5, uncertainty value = 1; (C) prob. value = 4, uncertainty value = 2; (D) prob. value = 0.1, uncertainty value = 5; (E) prob. value = 400000, uncertainty value = 5; (F) prob. value = 100, uncertainty value = 100; (G) prob. value = 0, uncertainty value = 0; (H) prob. value = 0, uncertainty value = 1.5; and (I) prob. value = 0, uncertainty value = 1. The LHS results for these distributions were compared to hand calculated values. The code did not pass all tests for this chi-square distribution LHS analysis.

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: The expected values for the distribution were based upon the following criteria (according to the SAPHIRE Technical Reference Manual IRRAS/SARA 4.0, page 69):

Distribution A should be distributed as $2\chi^2(1)$
Distribution B should be distributed as $5\chi^2(1)$
Distribution C should be distributed as $2\chi^2(2)$
Distribution D should be distributed as $(1/50)\chi^2(5)$
Distribution E should be distributed as $8000\chi^2(5)$
Distribution F should be distributed as $\chi^2(100)$
Distribution G should be undefined
Distribution H should be undefined
Distribution I should be zero

The output from the LHS were compared to a standard table of chi-square values. Tests for distributions A, B, C, D, E, and I did not pass the acceptance criteria (see report AR-30-0893-004). Tests for distribution G and H did not pass the acceptance criteria (see report AR-30-0893-005).

Input data file names and listings: One basic event was input to an OR gate. The basic event had an chi-square distribution with the following parameters:

Parameter	A	B	C	D	E	F	G	H	I
Prob. value	2	5	4	0.1	40,000	100	0	0	0
Uncertainty value	1	1	2	5	5	100	0	1.5	1
Calc. type	1	1	1	1	1	1	1	1	1
Uncertainty type	C	C	C	C	C	C	C	C	C
Init. Flag	Y	Y	Y	Y	Y	Y	Y	Y	Y

Output data file names and listings: See attachments.

SOFTWARE TEST RECORD

STR No.   :   STR-30-006-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Test Number: 006

Test Description: Tested the Monte Carlo (MC) sampling for a single event. The event tested was for the chi-square distribution. Three chi-square distributions were tested: (A) prob. value = 1, uncertainty value = 1; (B) prob. value = 2, uncertainty value = 2; (C) prob. value = 5, uncertainty value = 5. The three distributions were tested using different number of sample values (500, 1000, 3000, 5000, and 10000). The MC results for these distributions were compared to hand calculated values. The code did pass all tests for this chi-square distribution MC analysis.

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: The expected values for the distribution were based upon the following criteria (according to the SAPHIRE Technical Reference Manual IRRAS/SARA 4.0, page 69):

Distribution A should be distributed as $\chi^2(1)$
Distribution B should be distributed as $\chi^2(2)$
Distribution C should be distributed as $\chi^2(5)$

The output from the MC sampling were compared to a standard table of chi-square values.

Input data file names and listings: One basic event was input to an OR gate. The basic event had an chi-square distribution with the following parameters:

Parameter	A	B	C
Prob. value	1	2	5
Uncertainty value	1	2	5
Calc. type	1	1	1
Uncertainty type	C	C	C
Init. Flag	Y	Y	Y

Output data file names and listings: See attachments.

NUREG/CR-6145

Appendix C

SOFTWARE TEST RECORD

STR No.   :   STR-31-006-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Test Number: 006

Test Description: Tested the Latin Hypercube sampling (LHS) for a single event. The event tested was for the chi-square distribution. Three chi-square distributions were tested: (A) prob. value = 1, uncertainty value = 1; (B) prob. value = 2, uncertainty value = 2; (C) prob. value = 5, uncertainty value = 5. The three distributions were tested using different number of sample values (100, 500, 1000, 3000, 5000, and 10000). The LHS results for these distributions were compared to hand calculated values. The code did pass all tests for this chi-square distribution LHS analysis.

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: The expected values for the distribution were based upon the following criteria (according to the SAPHIRE Technical Reference Manual IRRAS/SARA 4.0, page 69):

Distribution A should be distributed as $\chi^2(1)$
Distribution B should be distributed as $\chi^2(2)$
Distribution C should be distributed as $\chi^2(5)$

The output from the LHS were compared to a standard table of chi-square values.

Input data file names and listings: One basic event was input to an OR gate. The basic event had an chi-square distribution with the following parameters:

Parameter	A	B	C
Prob. value	1	2	5
Uncertainty value	1	2	5
Calc. type	1	1	1
Uncertainty type	C	C	C
Init. Flag	Y	Y	Y

Output data file names and listings: See attachments.

SOFTWARE TEST RECORD

STR No.   :   STR-30-007-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Test Number: 007

Test Description: Tested the Monte Carlo (MC) sampling for a single event. The event tested was for the normal distribution. A single normal distribution was tested with prob. value = 0 and standard deviation of = 1 (i.e., a standard normal distribution). This event was tested for several different sample sizes (500, 1000, 3000, 5000, and 10000 samples) and several different seed values (512, 4321, and 65533). The MC results for these distributions were compared to hand calculated values. The code passed all tests for this normal distribution MC analysis.

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: The expected values for the distribution are:

Parameter	Distribution
Mean	0
5th percentile	-1.65
50th percentile	0
95th percentile	1.65
Standard Dev.	1
Skewness	0
Kurtosis	3

Input data file names and listings: One basic event was input to an OR gate. The basic event had an normal distribution with the following parameters:

Parameter	Distribution
Prob. value	0
Uncertainty value	1
Calc. type	1
Uncertainty type	N
Init. Flag	Y

Output data file names and listings: See attachments.

Appendix C

SOFTWARE TEST RECORD

STR No.   :   STR-31-007-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Test Number: 007

Test Description: Tested the Latin Hypercube sampling (LHS) for a single event. The event tested was for the normal distribution. A single normal distribution was tested with prob. value = 0 and standard deviation of = 1 (i.e., a standard normal distribution). This event was tested for several different sample sizes (100, 500, 1000, 3000, 5000, and 10000 samples) and several different seed values (512, 4321, and 65533). The LHS results for these distributions were compared to hand calculated values. The code passed all tests for this normal distribution LHS analysis.

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: The expected values for the distribution are:

Parameter	Distribution
Mean	0
5th percentile	-1.65
50th percentile	0
95th percentile	1.65
Standard Dev.	1
Skewness	0
Kurtosis	3

The maximum and minimum values from the LHS sampling were checked against the expected upper or lower values knowing that: (A) if 1000 samples are used, the lower probability interval will be between 0 and 1/1000, (B) the lower interval from the standard normal distribution from 0 to 1/1000 corresponds to a value of -3.08 (taken from table of cumulative standard normal probabilities), and (C) consequently, the minimum LHS sample should be between $-\infty$ and -3.08 for this distribution for 1000 samples. All the minimum and maximum samples checked out to be within their appropriate intervals.

Input data file names and listings: One basic event was input to an OR gate. The basic event had an normal distribution with the following parameters:

Parameter	Value	Parameter	Value
Prob. value	0	Uncertainty value	1
Calc. type	1	Uncertainty type	N
Init. Flag	Y		

Output data file names and listings: See attachments.

SOFTWARE TEST RECORD

STR No.   :   STR-30-008-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Test Number: 008

Test Description: Tested the Monte Carlo (MC) sampling for a single event (seed = 4321, 5000 samples). The event tested was for the gamma distribution. Six gamma distributions were tested: (A) prob. value = 0.5, uncertainty value = 5; (B) prob. value = 100, uncertainty value = 20; (C) prob. value = 1E–4, uncertainty value = 10000; (D) prob. value = 0.01, uncertainty value = 100; (E) prob. value = 1, uncertainty value = 0.5; and (F) prob. value = 2, uncertainty value = 1. The MC results for these distributions were compared to hand calculated values. The code <u>did not</u> pass all tests for the gamma distribution MC analysis.

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: The expected values for the distribution were based upon the following criteria:

Parameter	A	B	C	D	E	F
Mean	0.5	100	1E–4	1E–2	1	2
Standard deviation	0.2236	22.36	1E–6	1E–3	1.4142	2
Skewness	0.894	0.447	0.02	0.2	2.83	2
Kurtosis	4.2	3.3	3.0006	3.06	15	9

Tests for distribution C did not pass the acceptance criteria (see report AR-30-0893-006). Distributions E and F passed the acceptance criteria, but the results differed a little from what was expected (see report AR-30-0893-007).

Input data file names and listings: One basic event was input to an OR gate. The basic event had a gamma distribution with the following parameters:

Parameter	A	B	C	D	E	F
Prob. value	0.5	100	1E–4	1E–2	1	2
Uncertainty value	5	20	1E+4	1E+2	0.5	1
Calc. type	1	1	1	1	1	1
Uncertainty type	G	G	G	G	G	G
Init. Flag	Y	Y	Y	Y	Y	Y

Output data file names and listings: See attachments.

Appendix C

SOFTWARE TEST RECORD

STR No.   :   STR-31-008-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Test Number: 008

Test Description: Tested the Latin Hypercube sampling (LHS) for a single event (seed = 4321, 5000 samples). The event tested was for the gamma distribution. Six gamma distributions were tested: (A) prob. value = 0.5, uncertainty value = 5; (B) prob. value = 100, uncertainty value = 20; (C) prob. value = 1E-4, uncertainty value = 10000; (D) prob. value = 0.01, uncertainty value = 100; (E) prob. value = 1, uncertainty value = 0.5; and (F) prob. value = 2, uncertainty value = 1. The LHS results for these distributions were compared to hand calculated values. The code passed all tests for the gamma distribution LHS analysis.

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: The expected values for the distribution were based upon the following criteria:

Parameter	A	B	C	D	E	F
Mean	0.5	100	1E-4	1E-2	1	2
Standard deviation	0.2236	22.36	1E-6	1E-3	1.4142	2
Skewness	0.894	0.447	0.02	0.2	2.83	2
Kurtosis	4.2	3.3	3.0006	3.06	15	9

Input data file names and listings: One basic event was input to an OR gate. The basic event had a gamma distribution with the following parameters:

Parameter	A	B	C	D	E	F
Prob. value	0.5	100	1E-4	1E-2	1	2
Uncertainty value	5	20	1E+4	1E+2	0.5	1
Calc. type	1	1	1	1	1	1
Uncertainty type	G	G	G	G	G	G
Init. Flag	Y	Y	Y	Y	Y	Y

Output data file names and listings: See attachments.

SOFTWARE TEST RECORD

STR No.   :   STR-30-009-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Test Number: 009

Test Description: Tested the Monte Carlo (MC) sampling for a single event. The event tested was for the gamma distribution. A single gamma distribution was tested with prob. value = 0.1 and uncertainty value of = 1. This event was tested for several different sample sizes (500, 1000, 3000, 5000, and 10000 samples) and several different seed values (512, 4321, and 65533). The MC results for these distributions were compared to hand calculated values. The code passed all tests for this gamma distribution MC analysis.

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: The expected values for the distribution are:

Parameter	Distribution
Mean	0.1
5th percentile	0.00513
50th percentile	0.0693
95th percentile	0.2996
Standard Dev.	0.1
Skewness	2
Kurtosis	9

Input data file names and listings: One basic event was input to an OR gate. The basic event had an gamma distribution with the following parameters:

Parameter	Distribution
Prob. value	0.1
Uncertainty value	1
Calc. type	1
Uncertainty type	G
Init. Flag	Y

Output data file names and listings: See attachments.

Appendix C

SOFTWARE TEST RECORD

STR No.	:	STR-31-009-0893
Date	:	August 1993
Originator	:	Curtis L. Smith

Test Number: 009

Test Description: Tested the Latin Hypercube sampling (LHS) for a single event. The event tested was for the gamma distribution. A single gamma distribution was tested with prob. value = 0.1 and uncertainty value of = 1. This event was tested for several different sample sizes (100, 500, 1000, 3000, 5000, and 10000 samples) and several different seed values (512, 4321, and 65533). The LHS results for these distributions were compared to hand calculated values. The code passed all tests for this gamma distribution LHS analysis.

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: The expected values for the distribution are:

Parameter	Distribution
Mean	0.1
5th percentile	0.00513
50th percentile	0.0693
95th percentile	0.2996
Standard Dev.	0.1
Skewness	2
Kurtosis	9

Input data file names and listings: One basic event was input to an OR gate. The basic event had an gamma distribution with the following parameters:

Parameter	Distribution
Prob. value	0.1
Uncertainty value	1
Calc. type	1
Uncertainty type	G
Init. Flag	Y

Output data file names and listings: See attachments.

SOFTWARE TEST RECORD

STR No.   :   STR-30-010-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Test Number: 010

Test Description: Tested the Monte Carlo (MC) sampling for a single event (seed = 4321, 5000 samples). The event tested was for the beta distribution. Nine beta distributions were tested: (A) prob. value = 0.5, uncertainty value = 0.5; (B) prob. value = 1, uncertainty value = 1; (C) prob. value = -1, uncertainty value = 0.5; (D) prob. value = 0.5, uncertainty value = 0; (E) prob. value = 0.001, uncertainty value = 0.2; (F) prob. value = 1E-6, uncertainty value = 2; (G) prob. value = 1E-6, uncertainty value = 800000; (H) prob. value = 0.5, uncertainty value = 1; and (I) prob. value = 0.5, uncertainty value = 5. The MC results for these distributions were compared to hand calculated values or other MC code (@RISK) calculations. The IRRAS code did not pass all tests for the beta distribution MC analysis.

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: The expected values for the distribution were based upon the following criteria:

Parameter	A	B	C	D	E	F	G	H	I
Mean	0.5	n/a	n/a	n/a	0.001	1E-6	1E-6	0.5	0.5
Std. deviation	0.354	n/a	n/a	n/a	0.0289	5.8E-4	1.1E-6	0.289	0.151

Tests for distributions B through D did not pass the acceptance criteria (see report AR-30-0893-008). Distributions E through G also did not pass the acceptance criteria (see report AR-30-0893-009).

Input data file names and listings: One basic event was input to an OR gate. The basic event had a beta distribution with the following parameters:

Parameter	A	B	C	D	E	F	G	H	I
Prob. value	0.5	1	-1	0.5	0.001	1E-6	1E-6	0.5	0.5
Uncertainty value	0.5	1	0.5	0	0.2	2	800000	1	5
Calc. type	1	1	1	1	1	1	1	1	1
Uncertainty type	B	B	B	B	B	B	B	B	B
Init. Flag	Y	Y	Y	Y	Y	Y	Y	Y	Y

Output data file names and listings: See attachments.

Appendix C

SOFTWARE TEST RECORD

STR No.   :   STR-31-010-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Test Number: 010

Test Description: Tested the Latin Hypercube sampling (LHS) for a single event (seed = 4321, 5000 samples). The event tested was for the beta distribution. Nine beta distributions were tested: (A) prob. value = 0.5, uncertainty value = 0.5; (B) prob. value = 1, uncertainty value = 1; (C) prob. value = -1, uncertainty value = 0.5; (D) prob. value = 0.5, uncertainty value = 0; (E) prob. value = 0.001, uncertainty value = 0.2; (F) prob. value = 1E-6, uncertainty value = 2; (G) prob. value = 1E-6, uncertainty value = 800000; (H) prob. value = 0.5, uncertainty value = 1; and (I) prob. value = 0.5, uncertainty value = 5. The LHS results for these distributions were compared to hand calculated values or other LHS code (@RISK) calculations. The IRRAS code did not pass all tests for the beta distribution LHS analysis.

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: The expected values for the distribution were based upon the following criteria:

Parameter	A	B	C	D	E	F	G	H	I
Mean	0.5	n/a	n/a	n/a	0.001	1E-6	1E-6	0.5	0.5
Std. deviation	0.354	n/a	n/a	n/a	0.0289	5.8E-4	1.1E-6	0.289	0.151

Tests for distributions B through D did not pass the acceptance criteria (see report AR-30-0893-008). Distributions E through G also did not pass the acceptance criteria (see report AR-30-0893-009).

Input data file names and listings: One basic event was input to an OR gate. The basic event had a beta distribution with the following parameters:

Parameter	A	B	C	D	E	F	G	H	I
Prob. value	0.5	1	-1	0.5	0.001	1E-6	1E-6	0.5	0.5
Uncertainty value	0.5	1	0.5	0	0.2	2	800000	1	5
Calc. type	1	1	1	1	1	1	1	1	1
Uncertainty type	B	B	B	B	B	B	B	B	B
Init. Flag	Y	Y	Y	Y	Y	Y	Y	Y	Y

Output data file names and listings: See attachments.

SOFTWARE TEST RECORD

STR No.   :   STR-30-011-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Test Number: 011

Test Description: Tested the Monte Carlo (MC) sampling for a single event (seed = 4321, 5000 samples). The event tested was for the histogram distribution. Four histogram distributions were tested. The distribution parameters are shown below in the "Input data file" section. The MC results for these distributions were compared to hand calculated values or plotted. The IRRAS code passed all tests for the histogram distribution MC analysis.

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: The expected values for the distribution were based upon the following criteria:

Distribution A should be a uniform distribution from 0 to 1.
Distribution B should be a point estimate with a value of 0.5.
Distribution C should have a mean of approximately 0.0248, a standard deviation of approximately 0.0202, and be skewed to the right.
Distribution D should be the same as distribution C.

Input data file names and listings: One basic event was input to an OR gate. The basic event had a histogram distribution with the following parameters:

Parameter	A	B	C	D
Bin/probability (percent/area/height)	1/0.1 (10) 2/0.2 (10) 3/0.3 (10) 4/0.4 (10) 5/0.5 (10) 6/0.6 (10) 7/0.7 (10) 8/0.8 (10) 9/0.9 (10) 10/1 (10)	1/0.5 (100)	0/1E-3 1/1E-2 (0.5) 2/1E-1 (0.5)	0/1E-3 1/1E-2 (55.55) 2/1E-1 (5.555)
Histogram type	Percentage	Percentage	Area	Range
Calc. type	1	1	1	1
Uncertainty type	H	H	H	H
Init. Flag	Y	Y	Y	Y

Output data file names and listings: See attachments.

Appendix C

SOFTWARE TEST RECORD

STR No.   :   STR-31-011-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Test Number: 011

Test Description: Tested the Latin Hypercube sampling (LHS) for a single event (seed = 4321, 5000 samples). The event tested was for the histogram distribution. Four histogram distributions were tested. The distribution parameters are shown below in the "Input data file" section. The LHS results for these distributions were compared to hand calculated values or plotted. The IRRAS code passed all tests for the histogram distribution LHS analysis.

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: The expected values for the distribution were based upon the following criteria:

Distribution A should be a uniform distribution from 0 to 1.
Distribution B should be a point estimate with a value of 0.5.
Distribution C should have a mean of approximately 0.0248, a standard deviation of approximately 0.0202, and be skewed to the right.
Distribution D should be the same as distribution C.

The resulting distribution parameters for A, B, C, and D appear to be acceptable.

Input data file names and listings: One basic event was input to an OR gate. The basic event had a histogram distribution with the following parameters:

Parameter	A	B	C	D
Bin/probability (percent/area/height)	1/0.1 (10) 2/0.2 (10) 3/0.3 (10) 4/0.4 (10) 5/0.5 (10) 6/0.6 (10) 7/0.7 (10) 8/0.8 (10) 9/0.9 (10) 10/1 (10)	1/0.5 (100)	0/1E-3 1/1E-2 (0.5) 2/1E-1 (0.5)	0/1E-3 1/1E-2 (55.55) 2/1E-1 (5.555)
Histogram type	Percentage	Percentage	Area	Range
Calc. type	1	1	1	1
Uncertainty type	H	H	H	H
Init. Flag	Y	Y	Y	Y

Output data file names and listings: See attachments.

SOFTWARE TEST RECORD

STR No.	:	STR-30-012-0893
Date	:	August 1993
Originator	:	Curtis L. Smith

Test Number: 012

Test Description: Tested the Monte Carlo (MC) sampling for a single event. The event tested was for the normal distribution. A single normal distribution was tested with prob. value = 0 and standard deviation of = -1. This event was tested for sample sizes of 5000 and seed of 4321. The analysis should not give results since the standard deviation should be a positive value. The code did not pass all the test for this normal distribution MC analysis.

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: The expected values for the distribution are: Results should not be generated since the standard deviation must be greater than zero. See anomaly report AR-30-0893-010.

Input data file names and listings: One basic event was input to an OR gate. The basic event had an normal distribution with the following parameters:

Parameter	Distribution
Prob. value	0
Uncertainty value	-1
Calc. type	1
Uncertainty type	N
Init. Flag	Y

Output data file names and listings: See attachments.

Appendix C

SOFTWARE TEST RECORD

STR No.   :   STR-31-012-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Test Number: 012

Test Description: Tested the Latin Hypercube sampling (LHS) for a single event. The event tested was for the normal distribution. A single normal distribution was tested with prob. value = 0 and standard deviation of = -1. This event was tested for sample sizes of 5000 and seed of 4321. The analysis should not give results since the standard deviation should be a positive value. The code did not pass all the test for this normal distribution LHS analysis.

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: The expected values for the distribution are: Results should not be generated since the standard deviation must be greater than zero. See anomaly report AR-30-0893-010.

Input data file names and listings: One basic event was input to an OR gate. The basic event had an normal distribution with the following parameters:

Parameter	Distribution
Prob. value	0
Uncertainty value	-1
Calc. type	1
Uncertainty type	N
Init. Flag	Y

Output data file names and listings: See attachments.

SOFTWARE TEST RECORD

STR No.   :   STR-30-013-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Test Number: 013

Test Description: Tested the Monte Carlo (MC) sampling for a single event. The event tested was for the gamma distribution. A single gamma distribution was tested with prob. value = 0 and uncertainty value of = -1. This event was tested for sample sizes of 5000 and seed of 4321. The analysis should not give results since the uncertainty value should be a positive value. The code did not pass all the test for this gamma distribution MC analysis.

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: The expected values for the distribution are: Results should not be generated since the uncertainty value must be greater than zero. See anomaly report AR-30-0893-011.

Input data file names and listings: One basic event was input to an OR gate. The basic event had an gamma distribution with the following parameters:

Parameter	Distribution
Prob. value	0
Uncertainty value	-1
Calc. type	1
Uncertainty type	G
Init. Flag	Y

Output data file names and listings: See attachments.

Appendix C

SOFTWARE TEST RECORD

STR No.   :   STR-31-013-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Test Number: 013

Test Description: Tested the Latin Hypercube sampling (LHS) for a single event. The event tested was for the gamma distribution. A single gamma distribution was tested with prob. value = 0 and uncertainty value of = -1. This event was tested for sample sizes of 5000 and seed of 4321. The analysis should not give results since the uncertainty value should be a positive value. The code did not pass all the test for this gamma distribution LHS analysis.

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: The expected values for the distribution are: Results should not be generated since the uncertainty value must be greater than zero. See anomaly report AR-30-0893-011.

Input data file names and listings: One basic event was input to an OR gate. The basic event had an gamma distribution with the following parameters:

Parameter	Distribution
Prob. value	0
Uncertainty value	-1
Calc. type	1
Uncertainty type	G
Init. Flag	Y

Output data file names and listings: See attachments.

SOFTWARE TEST RECORD

STR No.   :   STR-30-014-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Test Number: 014

Test Description: Tested the Monte Carlo (MC) sampling for a single event. The event tested was for the lognormal distribution. A single lognormal distribution was tested with prob. value = 1E–4 and error factor of = -10. This event was tested for sample sizes of 5000 and seed of 4321. The analysis should not give results since the error factor should be a positive value. The code did not pass all the test for this lognormal distribution MC analysis.

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: The expected values for the distribution are: Results should not be generated since the error factor must be greater than zero. See anomaly report AR-30-0893-012.

Input data file names and listings: One basic event was input to an OR gate. The basic event had an lognormal distribution with the following parameters:

Parameter	Distribution
Prob. value	1E–4
Uncertainty value	-10
Calc. type	1
Uncertainty type	L
Init. Flag	Y

Output data file names and listings: See attachments.

Appendix C

SOFTWARE TEST RECORD

STR No.   :   STR-31-014-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Test Number: 014

Test Description: Tested the Latin Hypercube sampling (LHS) for a single event. The event tested was for the lognormal distribution. A single lognormal distribution was tested with prob. value = 1E-4 and error factor of = -10. This event was tested for sample sizes of 5000 and seed of 4321. The analysis should not give results since the error factor should be a positive value. The code did not pass all the test for this lognormal distribution LHS analysis.

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: The expected values for the distribution are: Results should not be generated since the error factor must be greater than zero. See anomaly report AR-30-0893-012.

Input data file names and listings: One basic event was input to an OR gate. The basic event had an lognormal distribution with the following parameters:

Parameter	Distribution
Prob. value	1E-4
Uncertainty value	-10
Calc. type	1
Uncertainty type	L
Init. Flag	Y

Output data file names and listings: See attachments.

SOFTWARE TEST RECORD

STR No.   :   STR-30-015-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Test Number: 015

Test Description: Tested the Monte Carlo (MC) sampling for a single event. The event tested was for the uniform distribution. A single uniform distribution was tested with prob. value = 2 and uncertainty value of = 1. This event was tested for sample sizes of 5000 and seed of 4321. The analysis should not give results since the uncertainty value should be a larger than the probability value. The code did not pass all the test for this uniform distribution MC analysis.

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: The expected values for the distribution are: Results should not be generated since the uncertainty value must be greater than probability value. See anomaly report AR-30-0893-013.

Input data file names and listings: One basic event was input to an OR gate. The basic event had an uniform distribution with the following parameters:

Parameter	Distribution
Prob. value	2
Uncertainty value	1
Calc. type	1
Uncertainty type	U
Init. Flag	Y

Output data file names and listings: See attachments.

Appendix C

SOFTWARE TEST RECORD

STR No.   :   STR-31-015-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Test Number: 015

Test Description: Tested the Latin Hypercube sampling (LHS) for a single event. The event tested was for the uniform distribution. A single uniform distribution was tested with prob. value = 2 and uncertainty value of = 1. This event was tested for sample sizes of 5000 and seed of 4321. The analysis should not give results since the uncertainty value should be a larger than the probability value. The code did not pass all the test for this uniform distribution LHS analysis.

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: The expected values for the distribution are: Results should not be generated since the uncertainty value must be greater than probability value. See anomaly report AR-30-0893-013.

Input data file names and listings: One basic event was input to an OR gate. The basic event had an uniform distribution with the following parameters:

Parameter	Distribution
Prob. value	2
Uncertainty value	1
Calc. type	1
Uncertainty type	U
Init. Flag	Y

Output data file names and listings: See attachments.

SOFTWARE TEST RECORD

STR No.   :   STR-32-001-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Test Number: 001

Test Description: Tested the Monte Carlo (MC) sampling for a sequence (i.e., a set containing multiple events) (seed = 4321, 5000 samples). The event tested was for a sequence of five exponential distributions ORed together. The exponential distribution parameters were: exponential with probability = 1E-2 (each of the five distributions had the same parameter but were independent, i.e., had different names and no correlation). The MC results for this sequence were compared to other MC calculated values. The IRRAS code passed all tests for this sequence MC analysis.

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: The expected values for the distribution were based upon the following criteria:

If X is distributed as exponential with parameter Θ, and if $Y = \sum_{i=1}^{n} X_i$, then Y will be distributed as gamma(Θ,n). Thus, the sequence was compared to a monte carlo simulation of a gamma(1E-2, 5) distribution.

Input data file names and listings: One basic event was input to an OR gate. The basic event had a histogram distribution with the following parameters:

Output data file names and listings: See attachments.

SOFTWARE TEST RECORD

STR No.   :   STR-33-001-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Test Number: 001

Test Description: Tested the Latin Hypercube sampling (LHS) for a sequence (i.e., a set containing multiple events) (seed = 4321, 5000 samples). The event tested was for a sequence of five exponential distributions ORed together. The exponential distribution parameters were: exponential with probability = 1E-2 (each of the five distributions had the same parameter but were independent, i.e., had different names and no correlation). The LHS results for this sequence were compared to other LHS calculated values. The IRRAS code passed all tests for this sequence LHS analysis.

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: The expected values for the distribution were based upon the following criteria:

If X is distributed as exponential with parameter Θ, and if $Y = \sum_{i=1}^{n} X_i$, then Y will be distributed as gamma(Θ,n). Thus, the sequence was compared to a LHS of a gamma(1E-2, 5) distribution.

Input data file names and listings: One basic event was input to an OR gate. The basic event had a histogram distribution with the following parameters:

Output data file names and listings: See attachments.

Appendix C

SOFTWARE TEST RECORD

STR No.   :   STR-32-002-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Test Number: 002

Test Description: Tested the Monte Carlo (MC) sampling for a sequence (i.e., a set containing multiple events) (seed = 4321, 5000 samples). The event tested was for the failure of the CCS and ECS systems in the SAPHIRE 4.16 DEMO database (i.e., the database that is shipped with the SAPHIRE code). The cutsets and basic event parameters are shown on the attachments. The MC results for these sequences were compared to other calculated values (from the @RISK computer code). The IRRAS code passed all tests for this sequence MC analysis.

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: See the attached @RISK results for the acceptance criteria.

Input data file names and listings: See the attachments for ECS and CCS sequence cutsets and basic events parameters.

Output data file names and listings: See attachments.

SOFTWARE TEST RECORD

STR No.   :   STR-33-002-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Test Number: 002

Test Description: Tested the Latin Hypercube sampling (LHS) for a sequence (i.e., a set containing multiple events) (seed = 4321, 5000 samples). The event tested was for the failure of the CCS and ECS systems in the SAPHIRE 4.16 DEMO database (i.e., the database that is shipped with the SAPHIRE code). The cutsets and basic event parameters are shown on the attachments. The LHS results for these sequences were compared to other calculated values (from the @RISK computer code). The IRRAS code passed all tests for this sequence LHS analysis.

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: See the attached @RISK results for the acceptance criteria.

Input data file names and listings: See the attachments for ECS and CCS sequence cutsets and basic events parameters.

Output data file names and listings: See attachments.

SOFTWARE TEST RECORD

STR No.	:	STR-32-003-0893
Date	:	August 1993
Originator	:	Curtis L. Smith

Test Number: 003

Test Description: Tested the Monte Carlo (MC) sampling for a sequence (i.e., a set containing multiple events) (seed = 4321, 5000 samples). The sequence that was tested was T1-BNU11 sequence (from the T1 event tree) in the Peach Bottom NUREG-1150 database (the PBOT40 database). The sequence contained 650 cutsets. The MC results for this sequence was compared to other calculated values (from the NUREG-1150 report). The IRRAS code passed all tests for this sequence MC analysis.

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: See the attached Peach Bottom NUREG-1150 results for the acceptance criteria.

Input data file names and listings: See the attachments for the sequence MC parameters.

Output data file names and listings: See attachments.

SOFTWARE TEST RECORD

STR No.	:	STR-33-003-0893
Date	:	August 1993
Originator	:	Curtis L. Smith

Test Number: 003

Test Description: Tested the Latin Hypercube sampling for a sequence (i.e., a set containing multiple events) (seed = 4321, 5000 samples). The sequence that was tested was T1-BNU11 sequence (from the T1 event tree) in the Peach Bottom NUREG-1150 database (the PBOT40 database). The sequence contained 650 cutsets. The LHS results for this sequence were compared to other calculated values (from the NUREG-1150 report). The IRRAS code passed all tests for this sequence LHS analysis.

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: See the attached Peach Bottom NUREG-1150 results for the acceptance criteria.

Input data file names and listings: See the attachments for the sequence LHS parameters.

Output data file names and listings: See attachments.

Appendix C

SOFTWARE TEST RECORD

STR No.   :   STR-32-004-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Test Number: 004

Test Description: Tested the Monte Carlo (MC) sampling for a three sequences (seed = 4321, 5000 samples). The sequences that were tested were T1-BNU11, T1-BU11NU21, and T1-P1BNU11 sequences (from the T1 event tree) in the Peach Bottom NUREG-1150 database (the PBOT40 database). The sequences contained a total of 1,330 cutsets. The MC results for these sequences were compared to other calculated values (from the NUREG-1150 report). The IRRAS code passed all tests for this sequence MC analysis.

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: See the attached Peach Bottom NUREG-1150 results for the acceptance criteria.

Input data file names and listings: See the attachments for the sequence MC parameters.

Output data file names and listings: See attachments.

SOFTWARE TEST RECORD

STR No.   :   STR-33-004-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Test Number: 004

Test Description: Tested the Latin Hypercube sampling (LHS) for a three sequences (seed = 4321, 5000 samples). The sequences that were tested were T1-BNU11, T1-BU11NU21, and T1-P1BNU11 sequences (from the T1 event tree) in the Peach Bottom NUREG-1150 database (the PBOT40 database). The sequences contained a total of 1,330 cutsets. The LHS results for these sequences were compared to other calculated values (from the NUREG-1150 report). The IRRAS code passed all tests for this sequence LHS analysis.

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: See the attached Peach Bottom NUREG-1150 results for the acceptance criteria.

Input data file names and listings: See the attachments for the sequence LHS parameters.

Output data file names and listings: See attachments.

SOFTWARE TEST RECORD

STR No.   :   STR-32-005-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Test Number: 005

Test Description: Tested the Monte Carlo (MC) sampling for a three sequences (seed = 4321, 5000 samples). The sequences that were tested were SEQ1, SEQ3, and SEQ5. These sequences were developed using different combinations of the distributions (and applicable parameters) that had previously checked out to be alright. The MC results for these sequences were compared to other calculated values (from the @RISK computer code). The IRRAS code passed all tests for this sequence MC analysis.

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: See the attached @RISK results for the acceptance criteria for each sequence.

Input data file names and listings: The sequences were:

SEQ1= B1*B3 + B2*B3 + B4*B3 + B5*B3 + B6*B3 + B7*B3 + B8*B3 + B9*B3 + B10*B3
SEQ3= B1*B2 + B3*B4 + B5*B6 + B7*B8 + B9*B10
SEQ5= B1*B6 + B2*B6 + B3 + B4*B9 + B5*B2 + B7*B6 + B8*B6 + B2*B10

The events B1 through B10 are:

Event	Distribution	Probability value	Uncertainty value
B1	beta	0.5	5
B2	exponential	0.1	1
B3	exponential	2E-4	1
B4	uniform	0.5	1
B5	uniform	0	1
B6	lognormal	1E-3	5
B7	chi-square	1	1
B8	chi-square	2	2
B9	gamma	0.01	100
B10	gamma	0.5	5

Output data file names and listings: See attachments.

Appendix C

SOFTWARE TEST RECORD

STR No.   :   STR-33-005-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Test Number: 005

Test Description: Tested the Latin Hypercube sampling (LHS) for a three sequences (seed = 4321, 5000 samples). The sequences that were tested were SEQ1, SEQ3, and SEQ5. These sequences were developed using different combinations of the distributions (and applicable parameters) that had previously checked out to be alright. The LHS results for these sequences were compared to other calculated values (from the @RISK computer code). The IRRAS code passed all tests for this sequence LHS analysis.

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: See the attached @RISK results for the acceptance criteria for each sequence.

Input data file names and listings: The sequences were:

SEQ1= B1*B3 + B2*B3 + B4*B3 + B5*B3 + B6*B3 + B7*B3 + B8*B3 + B9*B3 + B10*B3
SEQ3= B1*B2 + B3*B4 + B5*B6 + B7*B8 + B9*B10
SEQ5= B1*B6 + B2*B6 + B3 + B4*B9 + B5*B2 + B7*B6 + B8*B6 + B2*B10

The events B1 through B10 are:

Event	Distribution	Probability value	Uncertainty value
B1	beta	0.5	5
B2	exponential	0.1	1
B3	exponential	2E-4	1
B4	uniform	0.5	1
B5	uniform	0	1
B6	lognormal	1E-3	5
B7	chi-square	1	1
B8	chi-square	2	2
B9	gamma	0.01	100
B10	gamma	0.5	5

Output data file names and listings: See attachments.

SOFTWARE TEST RECORD

STR No.   :   STR-32-006-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Test Number: 006

Test Description: Tested the Monte Carlo (MC) sampling for a family (seed = 4321, 3000 samples). The family that was tested was the Beaver Valley Unit 2 IPE that was loaded into SAPHIRE 4.16 for the PRA Data Loading Project (FIN A6883). The MC results for this family were compared to the IPE reported results. The IRRAS code passed all tests for this family MC analysis.

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: The Beaver Valley Unit 2 IPE identified 1,697 (including 30 flood-related) dominant accident (core damage) sequences. The flood-related sequences are not included in the SAPHIRE database. Consequently, a total of 1,667 sequences were actually loaded into SAPHIRE for this database (and it is expected that the SAPHIRE core damage frequency (CDF) would be slightly lower than the IPE frequency). The CDF related statistics are summarized below.

Beaver Valley Unit 2 CDF Statistics (see IPE Figure 3.4.0-1)

Parameter	IPE Frequency
Mean	1.9E–04
95th Percentile	3.3E–04
50th Percentile	1.6E–04
5th Percentile	9.4E–05

The IRRAS Monte Carlo results seem to agree with the IPE results (when considering that 30 flood-related sequences are not included in the IRRAS results but were quantified in the IPE results).

Input data file names and listings: The BV2-4 database was used.

Output data file names and listings: See attachments.

Appendix C

SOFTWARE TEST RECORD

STR No.   :   STR-33-006-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Test Number: 006

Test Description: Tested the Latin Hypercube sampling (LHS) for a family (seed = 4321, 3000 samples). The family that was tested was the Beaver Valley Unit 2 IPE that was loaded into SAPHIRE 4.16 for the PRA Data Loading Project (FIN A6883). The LHS results for this family were compared to the IPE reported results. The IRRAS code passed all tests for this family LHS analysis.

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: The Beaver Valley Unit 2 IPE identified 1,697 (including 30 flood-related) dominant accident (core damage) sequences. The flood-related sequences are not included in the SAPHIRE database. Consequently, a total of 1,667 sequences were actually loaded into SAPHIRE for this database (and it is expected that the SAPHIRE core damage frequency (CDF) would be slightly lower than the IPE frequency). The CDF related statistics are summarized below.

Beaver Valley Unit 2 CDF Statistics (see IPE Figure 3.4.0-1)

Parameter	IPE Frequency
Mean	1.9E–04
95th Percentile	3.3E–04
50th Percentile	1.6E–04
5th Percentile	9.4E–05

The IRRAS LHS results seem to agree with the IPE results (when considering that 30 flood-related sequences are not included in the IRRAS results but were quantified in the IPE results).

Input data file names and listings: The BV2-4 database was used.

Output data file names and listings: See attachments.

SOFTWARE TEST RECORD

STR No.   :   STR-32-007-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Test Number: 007

Test Description: Tested the Monte Carlo (MC) sampling for endstates (seed = 4321, 3000 samples). The family that was tested was the Beaver Valley Unit 2 IPE that was loaded into SAPHIRE 4.16 for the PRA Data Loading Project (FIN A6883). The MC results for these endstates were compared to the IPE reported results. The IRRAS code passed all tests for this endstate MC analysis.

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: The Beaver Valley Unit 2 IPE defines 17 plant damage states that are associated with the Level 1 event trees. Each plant damage state is defined by a five-character code to define RCS pressure, availability of containment heat removal systems, and status of containment isolation or bypass at the time of reactor vessel failure after core damage, respectively. The IPE endstate results is shown on the attachment.

Most of the endstate values agree with the values reported in the IPE. Thus, it is assumed that IRRAS passes the acceptance criteria for this test.

Input data file names and listings: The BV2-4 database was used.

Output data file names and listings: See attachments.

SOFTWARE TEST RECORD

STR No.   :   STR-33-007-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Test Number: 007

Test Description: Tested the Latin Hypercube sampling (LHS) for endstates (seed = 4321, 3000 samples). The family that was tested was the Beaver Valley Unit 2 IPE that was loaded into SAPHIRE 4.16 for the PRA Data Loading Project (FIN A6883). The LHS results for these endstates were compared to the IPE reported results. The IRRAS code passed all tests for this endstate LHS analysis.

Computer system description: 33 MHz 486, Insight International, AT bus, DOS 5.0, 16 Mbyte RAM, 340 Mbyte IDE hard drive, SVGA monitor.

Expected values/acceptance criteria: The Beaver Valley Unit 2 IPE defines 17 plant damage states that are associated with the Level 1 event trees. Each plant damage state is defined by a five-character code to define RCS pressure, availability of containment heat removal systems, and status of containment isolation or bypass at the time of reactor vessel failure after core damage, respectively. The IPE endstate results is shown on the attachment.

Most of the endstate values agree with the values reported in the IPE. Thus, it is assumed that IRRAS passes the acceptance criteria for this test.

Input data file names and listings: The BV2-4 database was used.

Output data file names and listings: See attachments.

Appendix C

SOFTWARE TEST RECORD

STR No.   :   STR-34-001-0793
Date   :   7/20/93
Originator   :   LW

Test Number: Test 34

Test Description: DEMO Data base: change set: individual probability method—direct assignment of new probability. Create a change set with probability change on basic event C-MOV-A from 5.0E–3 to 0.1. Features 37 and 38 included in this test.

Computer system description: IBM Clone, Insight 486, 33 MHz, ISA bus, DOS 5.0, 16 Mb RAM, 340 Mb IDE hard drive, VGA display

Expected values/acceptance criteria: Expected values: chg prob of C-MOV-A from 5.0E–3 to 0.1 On chg set report, basic event prob report, and seq cut set report. Acceptable

Output data file names and listings: See attached

Input Data file names and listings: See attached

SOFTWARE TEST RECORD

STR No.   :   STR-35-001-0793
Date   :   7/20/93
Originator   :   LW

Test Number: Test 35

Test Description: DEMO data base: change set tagging method—manual designation of a group of events that need the same change definition. Create a chg set to chg the values of basic events C-CV-A, -B from 1.0E–4 to 0.1 and chg the values of basic events C-MOV-1 and E–MOV-1 from 1.0E–3 to 0.1. features 37 and 38 included in this test.

Computer system description: IBM Clone, Insight 486, 33 MHz, ISA bus, DOS 5.0, 16 Mb RAM, 340 Mb IDE hard drive, VGA display

Expected values/acceptance criteria: Expected values: chg prob of basic events C-MOV-1 and E–MOV-1 from 1.0E–3 to 0.1. Chg prob of basic events C-CV-A and C-CV-B from 1.0E–4 to 0.1. Values should be found on chg set report, basic event prob report, and seq cut set report. Acceptable

Input data file names and listings: See attached

Output data file names and listings: See attached

SOFTWARE TEST RECORD

STR No.   :   STR-36-001-0793
Date   :   7/20/93
Originator   :   LW

Test Number: TEST 36

Test Description: DEMO data base: Change set class method—set class attribute 1 to YES on basic events DG-A and DG-B. Set class attribute 3 to YES on basic events C-PUMP-A, -B and E-PUMP-A, -B. Set class attributes 1 and 3 to YES on basic events C-MOV-1 and E-MOV-1. Class chg set: "*" in name field, calc type = 1 and prob = 0.1 for all three tests. Class attribute 1 set to YES for first test. Set class attribute 3 to YES for second test. Set attributes 1 and 3 to YES for third test. Features 37 and 38 included in this test.

Computer system description: IBM Clone, Insight 486, 33 MHz, ISA bus, DOS 5.0, 16 Mb RAM, 340 Mb IDE hard drive, VGA display

Expected values/acceptance criteria: Test 1: DG-A, DG-B, C-MOV-1 and E-MOV-1 values in chg set report, basic event prob report, and sequence cut sets changed to 0.1. Test 2: C-PUMP-A, -B and E-PUMP-A, -B, C-MOV-1 and E-MOV-1 values in chg set report, basic event prob report, and sequence cut sets changed to 0.1. Test 3: C-MOV-1 and E-MOV-1 values in chg set report, basic event prob report and sequence cut sets changed to 0.1. Acceptable. See also AR-36-0793-001

Input data file names and listings: See attached

Output data file names and listings: See attached

SOFTWARE TEST RECORD

STR No.   :   STR-37-001-0793
Date   :   7/23/93
Originator   :   LW

Test Number: Test 37

Test Description: DEMO data base: change set features—enter the change set menu. Create/mark the change set to be used. Generate changes.

Computer system description: IBM Clone, Insight 486, 33 MHz, ISA bus, DOS 5.0, 16 Mb RAM, 340 Mb IDE hard drive, VGA display

Expected values/acceptance criteria: Basic events marked in the change set appear in the change set report, system and/or sequence reports. This feature is tested in STR-34-0793-001, STR-35-0793-001 and STR-36-0793-001. Acceptable

Input data file names and listings: None

Output data file names and listings: None

Appendix C

SOFTWARE TEST RECORD

STR No.   :   STR-38-001-0793
Date   :   7/23/93
Originator   :   LW

Test Number: Test 38

Test Description: DEMO data base: change set features—apply changes to system data. Enter analyze sequence menu. Select the quantification option. Select the system or sequence.

Computer system description: IBM Clone, Insight 486, 33 MHz, ISA bus, DOS 5.0, 16 Mb RAM, 340 Mb IDE hard drive, VGA display

Expected values/acceptance criteria: Values of basic events marked in the change set are changed in system or sequence cut set reports. This feature is tested in STR-34-0793-001, STR-35-0793-001 and STR-36-0793-001. Acceptable

Input data file names and listings: None

Output data file names and listings: None

SOFTWARE TEST RECORD

STR No.   :   STR-39-001-0793
Date   :   7/21/93
Originator   :   LW

Test Number: Test 39

Test Description: DEMO data base: change set features—sensitivity analysis, "S" flag. Create a chg set and set the process flag to "S" on basic events DG-A and DG-B. In the analyze sequences menu select hidden key "P" and highlight seq 2. Enter sensitivity values of 100 points, 0.01 low factor and 100 high factor. Plot figure.

Computer System description: IBM Clone, Insight 486, 33 MHz, ISA bus, DOS 5.0, 16 Mb RAM, 340 Mb IDE hard drive, VGA display

Expected values/acceptance criteria: Plot - See attached

Input data file names and listings: See attached

Output data file names and listings: See attached

SOFTWARE TEST RECORD

STR No.   :   STR-40-001-0793
Date   :   7/21/93
Originator   :   LW

Test Number: Test 40

Test Description: DEMO data base. Change set features: do not expand transfers, "X" flag. Part A: quantify CCS fault tree for system mincut. Insert mincut value (2.12E-2) into basic event base case probability for CCS and CALC Type = 1. Set process flag to "X" for CCS in chg set for sequence quantification. Part B: page CCS fault tree sub tree CCSTRA. Quantify CCS fault tree sub tree CCSTRA for system mincut. Insert mincut value (2.792E-2) into basic event base case probability for CCSTRA and calc type = 1. Set process flag to "X" for CCSTRA in chg set for system and sequence cut set quantification.

Computer system description: IBM Clone, Insight 486, 33 MHz, ISA bus, DOS 5.0, 16 Mb RAM, 340 Mb IDE hard drive, VGA display

Expected values/acceptance criteria: Part A: developed event CCS appears in sequence cut sets with a value of 2.12E-2. Acceptable

Part B: Developed event CCSTRA appears in system and sequence cut sets with a value of 2.792E-2. Acceptable

Input data file names and listings: See attached

Output data file names and listings: See attached

SOFTWARE TEST RECORD

STR No.   :   STR-41-001-0793
Date   :   7/21/93
Originator   :   LW

Test Number: Test 41

Test Description: DEMO data base: change set features—never expand transfers, "Y" flag. Quantify system CCS for system mincut. Insert system mincut value (2.12E-2) into basic event base case probability for CCS and set calc type = 1. Set Y process flag on system CCS in change set for system and sequence cut set quantification.

Computer system description: IBM Clone, Insight 486, 33 MHz, ISA bus, DOS 5.0, 16 Mb RAM, 340 Mb IDE hard drive, VGA display

Expected values/acceptance criteria: In sequence 2 the value of CCS is 9.788E-1. System failure in sequence 3 the value of CCS is 2.12E-2. Acceptable

Input data file names and listings: See attached

Output data file names and listings: See attached

Appendix C

SOFTWARE TEST RECORD

STR No.	:	STR-42-001-0793
Date	:	7/22/93
Originator	:	LW

Test Number: Test 42

Test Description: VV data base: Build event tree et and fault trees A and B (see attached). Set basic event rates as shown on attached basic event probability report. Generate sequence logic. Generate system and sequence cut sets (see attached input files). Sequence 2 of event tree et was used to test the process flags in IRRAS.

Create change set I-flag and set process flag to I on system A. Generate changes. Generate sequence 2 cut sets with no cut offs and with fault trees. Generate sequence 2 cut sets with no cutoffs and without fault trees. See attached output files for both test cases.

Computer system description: IBM Clone, Insight 486, 33 MHz, ISA bus, DOS 5.0, 16 Mb RAM, 340 Mb IDE hard drive, VGA display

Expected values/acceptance criteria: Sequence 2 cut set frequency with fault trees is 1.164E-3 and the cut set is /J,/K,L,M. This is correct.
Sequence 2 cut set frequency without fault trees is 2.362E-3 and the cut sets are /j,l,m freq. = 1.188E-3 and /K,L,M freq. = 1.176E-3. This is incorrect. The correct answer is /J,/K/L/M and mincut 1.164E-3. See AR-42-0793-001

Input data file names and listings: See attached

Output data file names and listings: See attached

SOFTWARE TEST RECORD

STR No.	:	STR-43-001-0793
Date	:	7/22/93
Originator	:	LW

Test Number: Test 43

Test Description: Change set features—blank flag field. This test has been performed in Tests 34 thru 42. See STR-34-0793-001 thru STR-41-0793-001.

Computer system description: IBM Clone, Insight 486, 33 MHz, ISA bus, DOS 5.0, 16 Mb RAM, 340 Mb IDE hard drive, VGA display

Expected values/acceptance criteria: Acceptable

Input data file names and listings:

Output data file names and listings:

SOFTWARE TEST RECORD

STR No.   :   STR-44/45/46/47/48/49-001-0793
Date   :   July 21, 1993
Originator   :   W. J. Galyean

Test Number: STR-44/45/47/47/48/49-001-07-93

Test Description: Using the CSS System from DEMO family in IRRAS, the Importance measures are calculated for each of the events appearing in the CSS System cutsets. These IRRAS generated Importance measures are then compared to calculation performed using a LOTUS 1-2-3 spreadsheet (see attached hard-copy, this spreadsheet is also included on the attached disk). All six Importance measures are included in these calculations.

Computer system description: Rightpoint 80486DX/33/50, DOS 5.0, 16 Mb installed RAM, 340 Mb IDE hard disk drive, TRIDENT 800x600 16c VGA with DataStor Non-Interlaced MNEVGA 28 Monitor, LOTUS 1-2-3, Release 3.1+.

Expected values/acceptance criteria: The IRRAS Importance measures were compared to those generated by the LOTUS spreadsheet. All agreed to three significant digits. This is judged acceptable.

Input data file names and listings: See attached.

Output data file names and listings: See attached.

SOFTWARE TEST RECORD

STR No.   :   STR-44/45/46/47/48/49-002-0793
Date   :   July 21, 1993
Originator   :   W. J. Galyean

Test Number: STR-44/45/46/47/48/49-002-07-93

Test Description: Using the LOSP event tree sequence #3 in the DEMO family, Importance measures were calculated by IRRAS. These same Importance measures were then manually calculated for event C-MOV-1 by setting the event probability to one and then zero. The resulting sequence frequencies were then used in hand calculations using the formulas shown on the attached sheet. These hand calculated Importance measures were then compared to those reported by IRRAS.

Computer system description: Rightpoint 80486DX/33/50, DOS 5.0, 16 Mb installed RAM, 340 Mb IDE hard disk drive, TRIDENT 800x600 16c VGA with DataStor Non-Interlaced MNEVGA 28 Monitor

Expected values/acceptance criteria: The IRRAS Importance measures for event "C-MOV-1" were compared to those calculated by hand. All agreed to three significant digits. This is judged acceptable.

Input data file names and listings: IRRAS DEMO family was used.

Output data file names and listings: See attached IRRAS report listings and hand calculation.

Appendix C

SOFTWARE TEST RECORD

STR No.   :   STR-48/49-003-0793
Date   :   July 21, 1993
Originator   :   W. J. Galyean

Test Number: STR-48/49-003-07-93

Test Description: Using the SARA-level LaSalle database (see attached disks) Importance measures were calculated by IRRAS for accident sequence identified as "T100". These Importance measures were then compared to those documented in LaSalle PRA. Only the Risk Reduction Interval and Risk Increase Intervals are published in the LaSalle PRA, therefore, only these two measures could be compared and checked.

Computer system description: Rightpoint 80486DX/33/50, DOS 5.0, 16 Mb installed RAM, 340 Mb IDE hard disk drive, TRIDENT 800x600 16c VGA with DataStor Non-Interlaced MNEVGA 28 Monitor.

Expected values/acceptance criteria: The IRRAS Importance measures for accident sequence "T100" were compared to those presented in "A. C. Payne, jr. et al., Analysis of the LaSalle Unit 2 Nuclear Power Plant: Risk Methods Integration and Evaluation Program (RMIEP), NUREG/CR-4832, Vol. 3 Part 2, August 1992." All agreed to three significant digits. This is judged acceptable.

Input data file names and listings: See attached disk.

Output data file names and listings: See attached IRRAS report listing.

Appendix D

Vital Features
Verification and Validation Anomaly Reports
Vital Features Verification and Validation
Software Test Records

Appendix D

Vital Features
Verification and Validation Anomaly Reports
Vital Features Verification and Validation
Software Test Records

The anomaly report number is of the form:

AR-VFN-MMYY-XXX

where

AR	-	Acronym for Anomaly Report
VFN	-	Two-digit vital feature number (i.e., 01 to 49)
MM	-	Two-digit code for the month of the test
YY	-	Two-digit code for the year of the test
XXX	-	Three-digit sequential number assigned by the tester.

Appendix D

ANOMALY REPORT

Anomaly Report No.   :   AR-2-7-93--001
Test Record No.   :   STR-2-7-09-001
Date   :   07/21/93
Originator   :   Steven D. Novack

Source/Location: Calculation of the maximum number of saved cutsets for one fault tree and one sequence.

Description: The formula provided by the vital features document is incorrect. It states that the total number of events plus the number of cutsets is less than 32,700. I found the correct value to be 32,000.

The cause is a misinterpretation by the programmer of the space provided for storage of the cutsets.

Impact: The impact is considered to be of a non-critical nature. The appropriate documentation should be changed, otherwise this condition should not directly effect the functionality of the code.

ANOMALY REPORT

Anomaly Report No.   :   AR-7A-08-93-001
Test Record No.   :   STR-7A-08-93-001
Date   :   8/17/93
Originator   :   Steven D. Novack

Source/Location: Analyze a large Fault tree (>1500 gates). Using the Generating cutsets option in Analyze Fault Tree. The error occurred when I was attempting to build a large fault tree by Adding systems together from the LaSalle-4 database.

Description: The maximum number of gates (2500) was never reached and is misleading since the total number of gates is dependant upon the amount of available RAM in the PC. Also I question the IRRAS calculation of the total number of gates (612) since it was less than 3 times the amount CAFTA calculated (1889).

Cause: Wrong limitations documented for the maximum number of gates and possible an incorrect calculation of the total number of gates.

Impact: The impact is considered to be of a non-critical nature. However, leads the user to believe that the problem may be in a different area other than memory. Also it is annoying since IRRAS will trash your database and require recovery actions if it is unable to process the tree. The appropriate documentation should be changed.

ANOMALY REPORT

Anomaly Report No.   :   AR-12-0893-001
Test Record No.   :   STR-12-001-0893
Date   :   8/5/93
Originator   :   LW

Source/Location: VV data base - link editor

Description: Multiple branches

1A) A subsequent substitution of top event C1 for top event C was not made after an initial substitution of top event B1 for top event B called by rule 1

Cause:

Impact: The impact is considered to be of a critical nature.

ANOMALY REPORT

Anomaly Report No.   :   AR-12-0893-002
Test Record No.   :   STR-12-001-0893
Date   :   8/5/93
Originator   :   LW

Source/Location: VV data base—link editor

Description: Multiple branches

1B) The first failure branch of a multiple branch top event was eliminated from sequence logic generation with an "@" symbol in the end state column (seq 3). Sequence logic generation correctly eliminated seq 3 from the logic listing, but top event B1 was not substituted for top event B in sequences 4 and 5.

Cause:

Impact: The impact is considered to be of a critical nature.

Appendix D

ANOMALY REPORT

Anomaly Report No.   :   AR-21-06-93-001
Test Record No.   :   STR-21-06-93-001
Date   :   6/21/93
Originator   :   Steven D. Novack

Source/Location: Calculation of the maximum number of saved cut sets for one fault tree and one sequence.

Description: The formula provided by the vital features document is incorrect. It states that the total number of events plus the number of cut sets is less than 32,700. I found the correct value to be 32,000.

Cause: Misinterpretation of the space provided for storage of cut sets.

Impact: The impact is considered to be of a non-critical nature. The appropriate documentation should be changed, otherwise this should not effect the functionality of the code.

ANOMALY REPORT

Anomaly Report No.   :   AR-22-06-93-001
Test Record No.   :   STR-22-001-06-93
Date   :   6/15/93
Originator   :   Steven D. Novack

Source/Location: Sequence Base Case Update

Description: The Report for the base case cut set results used the alternate change set values instead of the base case probabilities.

Cause: Unknown

Impact: This test has not been completed until feature is corrected. The impact is considered to be of a critical nature.

ANOMALY REPORT

Anomaly Report No.   :   AR-30-0893-001
Test Record No.   :   STR-30-002-0893, STR-31-002-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Source/Location: Performing Monte Carlo (and Latin Hypercube) sampling for the exponential distribution with various parameters.

Description: When using the exponential distribution, parameters less than and equal to zero should not be allowed. When these parameters (i.e., ≤ 0) are used, the IRRAS code gives zeros for the sampled value. Instead, IRRAS should not sample the distribution and should warn the user of the problem.

Cause: Not checking for valid input parameters for the uncertainty distributions.

Impact: The impact is of a critical nature.

ANOMALY REPORT

Anomaly Report No.   :   AR-30-0893-002
Test Record No.   :   STR-30-002-0893, STR-31-002-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Source/Location: Performing Monte Carlo (and Latin Hypercube) sampling for the exponential distribution with small (≈ 1E–16 or smaller) parameters.

Description: When using the exponential distribution, small parameters (≈ 1E–16 or smaller) seem to give erroneous results. For the test case (mean = 1.443E–16), the 5th percentile was shown as zero when it should have been approximately 7.4E–18. Also, the 50th p   and point estimate were shown as 1.11E–16 when they should have been 1.0E–16 and 1.443E–16, respectiv   ms that when the sampled value is very small, IRRAS essentially uses a zero value.

Cause: It could be a limit of the single precision arithmetic that is used (when calculating the mincut upper bound).

Impact: The impact is of a critical nature.

Appendix D

ANOMALY REPORT

Anomaly Report No.	:	AR-30-0893-003
Test Record No.	:	STR-30-003-0893, STR-31-003-0893
Date	:	August 1993
Originator	:	Curtis L. Smith

Source/Location: Performing Monte Carlo (MC) and Latin Hypercube sampling (LHS) for the lognormal distribution.

Description: When using the lognormal distribution, small parameters ($\approx$ 1E–15 or smaller) seem to give erroneous results. It seems that when the sampled value is very small, IRRAS essentially uses a zero value. It is believed that this problem is generic to all sampling types and distributions in IRRAS.

Cause: It could be a limit of the single precision arithmetic that is used (when calculating the mincut upper bound).

Impact: The impact is of a critical nature.

ANOMALY REPORT

Anomaly Report No.	:	AR-30-0893-004
Test Record No.	:	STR-30-005-0893, STR-31-005-0893
Date	:	August 1993
Originator	:	Curtis L. Smith

Source/Location: Performing Monte Carlo (MC) and Latin Hypercube sampling (LHS) for the chi-square distribution.

Description: When using the chi-square distribution, the SAPHIRE Technical Reference Manual: IRRAS/SARA Version 4.0 (NUREG/CR-5964), states that the chi-square distribution that is sampled (Y) is given by: $Y = aX$, where X is distributed as $\chi^2(k)$, k is the uncertainty value, and a = (probability value)/(uncertainty value). Thus, if the probability value = 2 and the uncertainty value = 1, then the sample chi-square distribution results should equal $2\chi^2(1)$.

All the tests which had a multiple of the chi-square distribution (i.e., distributions A through E and I) did not give sampling results as a multiple of the chi-square distribution. Instead, the results only seemed to give the chi-square distribution part [e.g., for distribution A, the results were very close to what would be expected for $\chi^2(1)$, not $2\chi^2(1)$].

Other than this discrepancy of not resulting in a multiple of the chi-square distribution, the chi-square sampling for n degrees of freedom seemed to be close to tabulated values of chi-square percentiles.

Cause: The definition of the use of the chi-square distribution appears to be in error since it seems that the chi-square distribution ignores the probability value and only accepts the uncertainty value as the degrees of freedom for the chi-square distribution.

Impact: The impact is of a critical nature.

ANOMALY REPORT

Anomaly Report No.	:	AR-30-0893-005
Test Record No.	:	STR-30-005-0893, STR-31-005-0893
Date	:	August 1993
Originator	:	Curtis L. Smith

Source/Location: Performing Monte Carlo (MC) and Latin Hypercube sampling (LHS) for the chi-square distribution.

Description: The chi-square distribution is normally defined as being distributed as $\chi^2(n)$, where n is the degrees of freedom. The degrees of freedom is normally defined as an integer larger than zero. Consequently, when non-integer degrees of freedom or values zero or less are used for the chi-square distribution, I would expect IRRAS to stop the sampling and warn the user of a potential problem. For the case when the degrees of freedom was set to zero (distribution G), a warning message did appear on the screen but the sampling continued. No warning messages were printed for the case (distribution H) when the degrees of freedom was not an integer.

Cause: No validation of the chi-square parameter inputs.

Impact: The impact is of a critical nature.

ANOMALY REPORT

Anomaly Report No.	:	AR-30-0893-006
Test Record No.	:	STR-30-008-0893
Date	:	August 1993
Originator	:	Curtis L. Smith

Source/Location: Performing Monte Carlo (MC) sampling for the gamma distribution.

Description: For the gamma distribution with prob. value 1E–4 and uncertainty value 10000, the MC sampling seemed to return the same value (7.068E–6) for each sample. Consequently, all the sampling distribution results were in error. Also, no error messages were displayed alerting the user to the problem. Two other statistical packages (@RISK and SAS) were used to see if the choice of parameters for the gamma distribution is a generic problem. Both @RISK and SAS correctly handled the gamma distribution with these parameters.

Cause: Unknown.

Impact: The impact is of a critical nature.

Appendix D

ANOMALY REPORT

Anomaly Report No.   :   AR-30-0893-007
Test Record No.   :   STR-30-008-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Source/Location: Performing Monte Carlo (MC) sampling for the gamma distribution.

Description: For the gamma distribution, when the uncertainty value equals n/2 (where n equals the degrees of freedom) and the probability field equals 2*(uncertainty value), then the gamma distribution will be chi-square distributed with n degrees of freedom. Consequently, I would expect the sampling for identical gamma and chi-square distributions to be exactly the same. The Monte Carlo sampling for the gamma distributions E and F (which should be $\chi^2(1)$ and $\chi^2(2)$, respectively) differs a little from the Monte Carlo sample for the corresponding chi-square distributions. The Latin Hypercube sampling for the gamma/chi-square distributions does not display this inconsistency.

Cause: Differing sampling method for chi-square and gamma distributions.

Impact: The impact is of a critical nature.

ANOMALY REPORT

Anomaly Report No.   :   AR-30-0893-008
Test Record No.   :   STR-30-010-0893, STR-30-010-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Source/Location: Performing Monte Carlo and Latin Hypercube sampling for the beta distribution.

Description: For the beta distribution, the parameters that were specified for distributions B, C, and D should result in undefined beta distributions. Instead of checking for appropriate input parameters for the beta distribution, IRRAS actions ranged from giving runtime errors (causing the need to recover the database) to running seemingly alright but giving nonsensical results.

Cause: No validation of beta distribution inputs.

Impact: The impact is of a critical nature.

ANOMALY REPORT

Anomaly Report No.	:	AR-30-0893-009
Test Record No.	:	STR-30-010-0893, STR-31-010-0893
Date	:	August 1993
Originator	:	Curtis L. Smith

Source/Location: Performing Monte Carlo and Latin Hypercube sampling for the beta distribution.

Description: For the beta distribution, the parameters that were specified for distributions E, F, and G should result in beta distributions which are legitimate. For these beta distributions, IRRAS actions ranged from giving runtime errors (causing the need to recover the database) to running seemingly alright but giving nonsensical results.

Cause: Unknown.

Impact: The impact is of a critical nature.

ANOMALY REPORT

Anomaly Report No.	:	AR-30-0893-010
Test Record No.	:	STR-30-012-0893, STR-31-012-0893
Date	:	August 1993
Originator	:	Curtis L. Smith

Source/Location: Performing Monte Carlo and Latin Hypercube sampling for the normal distribution.

Description: For the normal distribution, the uncertainty parameter (i.e., the standard deviation) should be a positive number. For this test, the standard deviation was input as a negative number. SAPHIRE still ran the analysis, appearing the take the absolute value of the uncertainty parameter. This could cause confusion and is deemed to be unacceptable.

Cause: No validation of the normal distribution inputs.

Impact: The impact is of a critical nature.

Appendix D

ANOMALY REPORT

`nomaly Report No.	: ´	AR-30-0893-011
Test Record No.	:	STR-30-013-0893, STR-31-013-0893
Date	:	August 1993
Originator	:	Curtis L. Smith

Source/Location: Performing Monte Carlo and Latin Hypercube sampling for the gamma distribution.

Description: For the gamma distribution, the uncertainty parameter should be a positive number. For this test, the uncertainty parameter was input as a negative number. SAPHIRE still ran the analysis, not giving an error message for the MC analysis while giving an error message (but still running the analysis) for the LHS analysis. This could result in incorrect results and is deemed to be unacceptable.

Cause: No validation of the gamma distribution inputs.

Impact: The impact is of a critical nature.

ANOMALY REPORT

Anomaly Report No.	:	AR-30-0893-012
Test Record No.	:	STR-30-014-0893, STR-31-014-0893
Date	:	August 1993
Originator	:	Curtis L. Smith

Source/Location: Performing Monte Carlo and Latin Hypercube sampling for the lognormal distribution.

Description: For the lognormal distribution, the uncertainty parameter (i.e., the error factor) should be a positive number. For this test, the error factor was input as a negative number. SAPHIRE still ran the analysis, appearing the take the point estimate value for each sample. This could cause errors and is deemed to be unacceptable.

Cause: No validation of the lognormal distribution inputs.

Impact: The impact is of a critical nature.

ANOMALY REPORT

Anomaly Report No.   :   AR-30-0893-013
Test Record No.   :   STR-30-015-0893, STR-31-015-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Source/Location: Performing Monte Carlo and Latin Hypercube sampling for the uniform distribution.

Description: For the uniform distribution, the uncertainty parameter should be greater than the mean value (since the uncertainty parameter represents the upper end point of the distribution). For this test, the uncertainty parameter was input as being less than the mean value. SAPHIRE still ran the analysis, appearing to simply swap the upper and lower end points of the distribution. This could cause errors and/or confusion and is deemed to be unacceptable.

Cause: No validation of the uniform distribution inputs.

Impact: The impact is of a critical nature.

ANOMALY REPORT

Anomaly Report No.   :   AR-31-0893-001
Test Record No.   :   STR-31-002-0893
Date   :   August 1993
Originator   :   Curtis L. Smith

Source/Location: Performing Latin Hypercube sampling (LHS) for the exponential distribution.

Description: When using the exponential distribution, the distribution results should be the same as those from the gamma distribution (when the uncertainty value parameter equals 1 in the gamma distribution). For the LHS analysis, the sampling for $EXP(\Theta)$ is not exactly the same as $GAMMA(\Theta,1)$ while for the monte carlo sampling the two distributions produce exactly the same sampling. While this is not a critical error (since both sampling methods for the LHS seem to produce accurate results), to be consistent the $EXP(\Theta)$ and $GAMMA(\Theta,1)$ sampling for LHS should be the same.

Cause: Two different LHS techniques for $EXP(\Theta)$ and $GAMMA(\Theta,1)$.

Impact: The impact is of a non-critical nature.

Appendix D

ANOMALY REPORT

Anomaly Report No.	:	AR-36-0793-001
Test Record No.	:	STR-36-001-0793
Date	:	7/20/93
Originator	:	LW

Source/Location: Generate changes menu - class change

Description: In order to mark a class of events, a "*" must be used, such as *MOV-1 or *MOV*. There is no help screen or discussion in the IRRAS 4.0 reference manual to use this feature or explain the use of these wild cards.

Cause:

Impact: The impact is considered to be of a non-critical nature.

ANOMALY REPORT

Anomaly Report No.	:	AR-39-07-93
Test Record No.	:	STR-39-07-93
Date	:	7/21/93
Originator	:	LW

Source/Location: Analyze sequences menu, analyze sequences

Description: Hidden key "P" allows access to the sensitivity menus but is not shown as a selection at the top of the page. There is no documentation for this feature even though there is a menu selection for it as a process flag in the basic events menu. Therefore, there is no way of knowing if it is performing properly.

Cause:

Impact: The impact is considered to be of a non-critical nature.

ANOMALY REPORT

Anomaly Report No.	:	AR-42-0793-001
Test Record No.	:	STR-42-001-0793
Date	:	7/22/93
Originator	:	LW

Source/Location: Generate event data

Description: Change set features: always expand transfers, "I" flag. A simple test case using the I flag did not output the correct cut sets when the sequence cut sets were generated without fault trees and no cut offs.

Cause:

Impact: The impact is considered to be of a critical nature.

Appendix E

Non-Vital Features
Verification and Validation Test Design Specification

Appendix E

Non-Vital Features
Verification and Validation Test Design Specification

E-1. TEST DESIGN SPECIFICATION IDENTIFIER: IVVDS-1

E-2. INTRODUCTION

The System Analysis Programs for Hands-on Integrated Reliability Evaluation (SAPHIRE) 4.0[1] is being developed at the Idaho National Engineering Laboratory (INEL) for the Nuclear Regulatory Commission (NRC) as a state-of-the-art microcomputer-based probabilistic risk assessment (PRA) model development and analysis tool to address key nuclear plant safety issues. This tool is integrated to give the user the ability to create and analyze fault trees, event trees, and accident sequences on a microcomputer. The programs included in this tool are as follows; Models and Results Database (MAR-D)[1] software, Integrated Reliability and Risk Analysis System (IRRAS)[1] software, Systems Analysis and Risk Assessment (SARA)[2] software, and Fault tree Event Tree, and P&ID (FEP)[1] graphical editor software. Each of these programs perform a specific function in taking a PRA from conceptual state all the way to publication.

The SAPHIRE program is developed for the purpose of performing those functions necessary to create and analyze a complete PRA. IRRAS includes functions that allow the user to create event trees and fault trees, define accident sequences and basic event failure data, solve system and accident sequence fault trees and event trees, quantify cut sets, and perform uncertainty analysis on the results. Also included in IRRAS are features to allow the analyst to generate reports and displays that can be used to document the results of an analysis. The users of IRRAS should be familiar with PRA concepts and the methods used to perform PRA analyses since this program is a detailed technical tool. Although IRRAS makes the PRA process easier through a user friendly design, the sheer complexity of a PRA analysis requires a user to have a more detailed understanding of PRA concepts than is required by other tools in this suite. The IRRAS 4.0 reference manual is available as NUREG/CR-5813, Volume 1.

MAR-D is used primarily for data loading. This program defines a common relational data base structure that is used by the entire suite of programs. Structuring MAR-D in this manner allows all of the software to access and manipulate data created by other software in the system without performing a lengthy conversion. Therefore, data created by IRRAS is immediately available to SARA for sensitivity analysis. MAR-D also provides the facilities for loading and unloading PRA data from the relational data base structure used to store the data to an ASCII format for interchange with other PRA software not included in NRC's suite of programs. This feature provides compatibility with previously developed software systems and allows for maximum data interchange. Elements of this software are included in IRRAS and SARA to allow these programs to load and unload data in the MAR-D format. Normally, the entire MAR-D software is used only by those performing a data loading function and is not required by the end user.

SARA allows the user to review the results of a PRA and perform limited sensitivity analysis on these results. The primary limitation is that the changes in the plant model can only be effected by using the cut set editor. If other simple changes are being simulated, then IRRAS should be used so that new cut sets can be accurately generated. SARA is intended to be used by a less technical oriented user and does not require the level of understanding of PRA concepts as required by IRRAS. This program allows a user to review the information generated and to compare the results to those generated by making limited modifications to data in the PRA. Additionally, this program has the ability to graphically display the information stored in the MAR-D data base. The information includes event trees, fault trees, P&IDs, and uncertainty distributions. Program users can gain a better understanding of the results of a PRA without getting into the details of the construction and analysis work behind the PRA. The SARA reference manual and tutorial are available as NUREG/CR-5303, Volumes 1 and 2, respectively.

Appendix E

The program FEP provides a common access to the suite of graphical tools developed for performing risk assessment. These tools include the graphical event tree, fault tree, and P&ID editors. The event tree and fault tree editors are available through IRRAS; however, the P&ID editor is only accessible through FEP. The event tree editor allows the analyst to construct and modify graphical event trees. The fault tree editor allows the analyst to construct and modify graphical fault trees. The P&ID editor allows the user to construct and modify plant drawings. These drawings can be used to document the modeling used in a PRA. These editors are an integral part of a PRA. The FEP tool allows using a less complex tool when the user only wants to generate graphical displays.

The roll of the INEL in the SAPHIRE program is to develop the software and interface with the community including transfer of technology. IRRAS Version 1.0 was released in February of 1987 to prove the concept of performing probabilistic risk assessments on microcomputers. Version 1.0 contained many of the basic features required for fault tree analysis and the PRA community was pleased with this initial effort. Since the release of Version 1.0, the SAPHIRE system has been made more powerful and user friendly through the incorporation of user comments and code enhancements. The latest version of SAPHIRE has been designated 4.0. Previous versions of SAPHIRE (previously titled IRRAS) include versions 2.0 released officially in June 1990 and 2.5 released August 1991.

SAPHIRE has all the necessary capabilities and functions required to create, modify, reduce, and analyze fault tree models used in the analysis of complex systems and processes. SAPHIRE uses advanced graphic and analytical techniques to maximize the potential found in today's microcomputers. Version 4.0 provides all the capabilities of Version 1.0 plus adds a relational data base facility for managing the data, provides improved functionality, and provides improved algorithm performance.

Since the SAPHIRE computer code is used by a variety of organizations, quality assurance (QA) is necessary to insure that the information produced by the code is correct and that the code can be used in applications that require all code functions to be validated and verified. Extensive Beta testing on the earlier prototype versions of SAPHIRE (titled IRRAS prior to version 4.0) gave credibility to the major functions. Version 4.0 is scheduled to be a production model and therefore, it has to be formally verified and validated to insure quality. Part of the formal verification is the construction of test design specifications.

The purpose of a test design specification is to summarize a computer code's critical and minor features. This test design specification will summarize the minor software features to be tested in test items IRRAS 4.0, SARA 4.0, MAR-D 4.0 and FEP 4.0. Minor or non-critical features are those operational characteristics of software that make solving problems easier but are not required to obtain the final correct results. An example could be the use of a "mouse" to access functions rather than use the keyboard. Where features in IRRAS 4.0 are the same as the features in SARA 4.0, MAR-D 4.0 and FEP 4.0 the same test cases will be run. The four test items are (1) IRRAS 4.0, (2) MAR-D 4.0, (3) SARA 4.0 and (4) FEP 4.0. The minor features for these items are divided into six categories. Category 1[3] contains features associated with fault tree analysis. Category 2[3] two features are connected with event tree analysis. Category 3[3] features deal with data base manipulations. Category 4[3] is associated with Histograms. Category 5[3] concerns the Result Display, while Category 6[3] addresses General Capabilities. The test items and features are as follows:

E-2.1 Test Item IRRAS 4.0

Category 1, fault tree analysis, features are as follows:

- Fault Tree Construction

- Plot Trees

- Load Graphic Trees

- Alpha to Graphics

- Graphical Pager

- Extract Fault Tree.

Category 2, event tree analysis, features are as follows:

- Event Tree Construction

- Plot Trees

- Graphics Load

- Extract Event Trees.

Category 3, data base manipulation, features are as follows:

- Add, modify, delete a family or modify the associated text

- Modify the event tree data records

- Modify the system data records

- Modify the end state data records

- Modify the basic event data records

- Edit the three categories of graphic data records

- Base case update.

Category 4, histograms, features are as follows:

- Range and percentage formats

- Modify a currently existing histogram

- Delete histograms.

Category 5, results display, features are as follows:

- All special function keys: F1, F2, F3, F4, F5

- Reporting functions (on screen and hard copy).

Category 6, general capabilities, features are as follows:

- Field specific as well as on line help (F1)

- Locate Functions (F5)

- Escape options

- Error Messages.

E-2.2 Test Item MAR-D 4.0

Category 3, data base manipulation, features are as follows:

- Add, modify, delete a family or modify the associated text

- Modify the event tree data records

- Modify the system data records

- Modify the end state data records

- Modify the basic event data records

- Edit the three categories of graphic data records

- Base case update.

Category 4, histograms, features are as follows:

- Range and percentage formats

- Modify a currently existing histogram

- Delete histograms.

Category 5, results display, features are as follows:

- All special function keys: F1, F2, F3, F4, F5

- Reporting functions (on screen and hard copy).

Category 6, general capabilities, features are as follows:

- Field specific as well as on line help (F1)

- Locate Functions (F5)

- Escape options

- Error Messages.

E-2.3 Test Item SARA 4.0

Category 3, data base manipulation, features are as follows:

- Add, modify, delete a family or modify the associated text

- Modify the event tree data records

- Modify the system data records

- Modify the end state data records

- Modify the basic event data records

ᵔ Edit the three categories of graphic data records

- Base case update.

Category 4, histograms, features are as follows:

- Range and percentage formats

- Modify a currently existing histogram

- Delete histograms.

Category 5, results display, features are as follows:

- All special function keys: F1, F2, F3, F4, F5

- Reporting functions (on screen and hard copy).

Category 6, general capabilities, features are as follows:

- Field specific as well as on line help (F1)

- Locate Functions (F5)

- Escape options

- Error Messages.

E-2.4 Test Item FEP 4.0

Category 1, fault tree analysis, features are as follows:

- Fault Tree Construction

- Plot Trees.

Category 2, event tree analysis, features are as follows:

- Event Tree Construction

- Plot Trees.

Category 6, general capabilities, features are as follows:

- Field specific as well as on line help (F1)

- Locate Functions (F5)

- Escape options

- Error Messages.

Category 7, FEP P&ID editor

- Build and plot P&IDs.

E-3. TEST ITEMS

The test items for this test design specification are identified in the introduction. The type of testing being performed on these items is minor or non-critical. The non-critical testing of the items will be accomplished by testing the non-critical features. This method departs slightly from the IEEE standard for software documentation.[4] The features to be tested will be listed with each test item rather than separately. The test items to be discussed are IRRAS 4.0, MAR-D 4.0, SARA 4.0 and FEP 4.0.

E-3.1 IRRAS 4.0

IRRAS 4.0 is developed for the purpose of performing those functions necessary to create and analyze a complete PRA. To expediently accomplish these functions, many minor features were added to aid the analyst. The minor features can be categorized as fault tree analysis, event tree analysis, data base manipulations, histograms, results display, and general capabilities.

E-3.1.1 Fault Tree Analysis

The minor features contained within fault tree analysis are fault tree construction, plot trees, load graphic trees, alpha to graphics, graphical pager, and extract fault trees.

Fault Tree Construction is accomplished in a graphical environment with the use of a mouse and graphical symbols. The graphical symbols are listed in a pull down menu and can be selected by clicking the left mouse button on the desired symbol. The analyst has the ability to:

- Create a fault tree by using the fault tree symbols in the "BILD" pull down menu

- Load a fault tree in the "DLS" format

- Edit a fault tree, including moving a portion of the tree diagram to a new location, copying a portion of the tree diagram and moving it to a new location, deleting any portion of the displayed fault tree, duplicating a portion of the fault tree diagram, copying text from one area of the diagram to another, moving text from one area of the diagram to another, editing fault tree text

- Save a fault tree in the "DLS" format

- Add text to a fault tree, including the ability to add text anywhere on the fault tree, adding a name or replacing a name

- View a fault tree, including shifting the figure up, scrolling the figure, zooming in, zooming out, restoring the original figure, showing the name of gates and events, and showing the grid

- Format fault tree diagrams for a Hewlett Packard laser printer, includes sending the diagram to the laser printer

- Display the current diagram file name

- Change attributes of gate and event names, including changing the color and text size of the name assigning a default name for gates and/or events

- Change the attributes of the text, including setting the justification for the diagrams, setting a default text size, and setting a default line type.

Plot Trees gives the analyst the ability to send a specific file to the plotter, generate an HPGL formatted file, and rasterize a selected fault tree diagram for printing. The rasterized file is in a format for a dot matrix printer.

Load Graphic Trees allows the user to load the drawings and logic of a fault tree or group of fault trees into the data base, list all fault trees, and update any cross reference maps.

Alpha to Graphics converts alpha file to a graphics file.

The Graphical Pager breaks up larger drawings into many pages or smaller drawings, locates specific events or labels on the fault tree diagram, and displays the label in a readable format.

Extract Fault Trees allows the user to extract any fault tree files from the data base, list fault tree drawing files that already exist, and delete any of those existing files. These files are extracted in the "DLS" format.

E-3.1.2 Event Tree Analysis

The minor features contained within event tree analysis are event tree construction, plot trees, graphics load, and extract event trees.

Event Tree Construction is accomplished in a graphical environment with the use of a mouse and "Tops" and "Ends" selections from the pull down menu. The "Tops" or Ends" are selected by clicking the left mouse button on the desired selection. The analyst has the ability to:

- Create an event tree by using the event tree symbols in the "Tops" and/or "Ends" pull down menu

- Load an event tree in the "ETG" format

- Edit an event tree, including adding a branch to the event tree, deleting branches from the event tree, copying existing branches of the event tree to a new location, changing a branch to a pass, separating a large drawing into separate pages

- Save an event tree in the "ETG" format

- Select the "Tops" menu, including adding a new event, deleting an event, changing an event name, conserving space on the diagram, changing the displayed width of the event names, changing the text size of the event names, selecting the font type for the event names and headers, editing or adding event name descriptions, specifying the height of the event descriptions, specifying the number of lines in event descriptions, and selecting the font type for the event descriptions

- Select the "Ends" menu, including displaying information about the end state, editing the header, moving a header to another location on the screen, specifying the height or distance between branches, specifying the height of the end state text, and specifying the font type for the end state text

- Add text to an event tree, including selecting the font type for the text, writing text at any desired location on the screen, moving the selected text to a new location on the screen, copying the selected text to a new location, deleting selected text, editing text and setting various text attributes such as color, size font, and justification, setting justification for the diagram, and setting default text size for the diagrams

- View an event tree, including changing the position and size of the displayed and event tree diagram, moving the drawing up, down, right, left, zooming in, zooming out, or restoring the diagram to the original size

- Format event tree diagrams for a Hewlett Packard laser printer, includes sending the diagram to the laser printer

- View and change the name of the family and the event tree file name

- Add, delete, and modify transfer file names

- Use the link editor, including adding new rules or modifying existing rules, inserting blank lines, deleting entire selected rules, and restoring the last deleted rule

- Use any basic event as a top event.

Plot Trees gives the analyst the ability to send a specific file to the plotter, generate an HPGL formatted file, and rasterize a selected fault tree diagram for printing. The rasterized file is in a format for a dot matrix printer.

Graphics Load allows the user to load event tree graphics into the system.

Extract Event Trees allows the user to extract any event tree files from the data base, list existing extracted event trees, and delete any of those existing files. These files are extracted in the "ETG" format.

E-3.1.3 Data Base Manipulations

Data base manipulations are used to keep the data records up-to-date. Manipulation of the data base allows the user to:

- Add, modify, or delete a family or modify associated text

- Modify the event tree data records

- Modify the system data records

- Modify the end state data records

- Modify the Basic event data records

- Modify the attribute data for the basic events

- Modify the graphic name, description, drawing (diagram), and logic for graphics including piping and instrumentation diagrams (P&ID), fault trees, and event trees

- Perform base case updates (overwrite the base case data values stored in the data base with the current data values for each basic event).

E-3.1.4 Histograms

The histogram option allows the analyst to input his own distribution for a variable that can not be expressed with one of the predefined distribution types. The histogram function allows the user to:

- Use either a range or percentage format

- Add a percentage histogram to the data base

- Add a range histogram to the data base

- Modify a currently existing histogram

- Delete histograms.

E-3.1.5 Results Display

The results display present the results for the various SAPHIRE 4.0 modules in a suitable format for reports. The results display allows the user to:

- Get general or field specific help with the <F1> key on all forms where this key is an option

- Mark or remove mark from highlighted name with the <F2> key on any form that displays a list of names

- Mark all entries with the <F3> key on any form that displays a list of names if no entries are marked and clear all marks from entries if any entry is marked

- Mark or remove mark from a range of entries with the <F4> key

- Locate function with <F5> key on any forms that display a list of names and show <F5> as an option

- Return from an operation without performing any functions with the <ESC> key

- Generate reports on screen or to the printer of families and their descriptions; basic event overviews, probabilities, and uncertainties; system and sequence summaries, combinations and logic; cut set data on base case and alternate case; system importance; subsystem data; event tree rules, logic, and initiating events; and system cut sets, the percentage contribution of each cut set to the system, the frequency of the cut set to the system, the frequency of the cut set, and the event names which make up the cut set.

E-3.1.6 General Capabilities

The SAPHIRE 4.0 program has many general capabilities to make the users experience easier. The general capabilities allow the user to:

- Get general or field specific help with the <F1> key on all forms where this key is an option

- Mark or remove mark from highlighted name with the <F2> key on any form that displays a list of names

- Mark all entries with the <F3> key on any form that displays a list of names if no entries are marked and clear all marks from entries if any entry is marked

- Mark or remove mark from a range of entries with the <F4> key

- Locate function with <F5> key on any forms that display a list of names and show <F5> as an option

- Return from an operation without performing any functions with the <ESC> key

- Go directly to a main menu option without returning to the main menu using "Hot keys" with the following functions:

    - <Alt/E> - Exit IRRAS 4.0

    - <Alt/A> - Analyze Sequences

    - <Alt/B> - Build Fault Trees

    - <Alt/C> - Create Event Trees

- <Alt/F> - Fault Tree Analysis

- <Alt/G> - Generate Event Data

- <Alt/M> - Modify Data Base

- <Alt/R> - Report on Data Base

- <Alt/S> - Select Family

- <Alt/U> - Utility Options

- Generate error messages to inform the user of incorrect data, invalid option selection, or failure of an operation

- Generate warning messages to inform the user of potential loss of data if chosen option is completed.

E-3.2 MAR-D 4.0

MAR-D 4.0 is developed primarily for data loading. This program defines a common relational data base structure that is used by IRRAS 4.0, SARA 4.0 and FEP 4.0. The Structure of MAR-D allows all of the software to access and manipulate data created by other software in the system without performing a lengthy conversion. The minor features can be categorized as data base manipulations, histograms, results display, and general capabilities.

E-3.2.1 Data Base Manipulations

Data base manipulations are used to keep the data records up-to-date. Manipulation of the data base allows the user to:

- Add, modify, or delete a family or modify associated text

- Modify the event tree data records

- Modify the system data records

- Modify the end state data records

- Modify the Basic event data records

- Modify the attribute data for the basic events

- Modify the graphic name, description, drawing (diagram), and logic for graphics including piping and instrumentation diagrams (P&ID), fault trees, and event trees

- Perform base case updates (overwrite the base case data values stored in the data base with the current data values for each basic event).

E-3.2.2 Histograms

The histogram option allows the analyst to input his own distribution for a variable that can not be expressed with one of the predefined distribution types. The histogram function allows the user to:

- Use either a range or percentage format

- Add a percentage histogram to the data base

- Add a range histogram to the data base

- Modify a currently existing histogram

- Delete histograms.

E-3.2.3 Results Display

The results display present the results for the various MAR-D 4.0 modules in a suitable format for reports. The results display allows the user to:

- Get general or field specific help with the <F1> key on all forms where this key is an option

- Mark or remove mark from highlighted name with the <F2> key on any form that displays a list of names

- Mark all entries with the <F3> key on any form that displays a list of names if no entries are marked and clear all marks from entries if any entry is marked

- Mark or remove mark from a range of entries with the <F4> key

- Locate function with <F5> key on any forms that display a list of names and show <F5> as an option

- Return from an operation without performing any functions with the <ESC> key

- Generate reports on screen or to the printer of families and their descriptions; basic event overviews, probabilities, and uncertainties; system and sequence summaries, combinations and logic; cut set data on base case and alternate case; system importance; subsystem data; event tree rules, logic, and initiating events; and system cut sets, the percentage contribution of each cut set to the system, the frequency of the cut set to the system, the frequency of the cut set, and the event names which make up the cut set.

E-3.2.4 General Capabilities

The MAR-D 4.0 program has many general capabilities to make the users experience easier. The general capabilities allow the user to:

- Get general or field specific help with the <F1> key on all forms where this key is an option

- Mark or remove mark from highlighted name with the <F2> key on any form that displays a list of names

- Mark all entries with the <F3> key on any form that displays a list of names if no entries are marked and clear all marks from entries if any entry is marked

- Mark or remove mark from a range of entries with the <F4> key

- Locate function with <F5> key on any forms that display a list of names and show <F5> as an option

- Return from an operation without performing any functions with the <ESC> key

- Generate error messages to inform the user of incorrect data, invalid option selection, or failure of an operation

- Generate warning messages to inform the user of potential loss of data if chosen option is completed.

E-3.3 SARA 4.0

With SARA, the user can review the results of a PRA and perform limited sensitivity analysis on these results. The most important limitation is that the changes in the plant model can only be effected by using the cut set editor. Less technically oriented users are the intended target of SARA. This program allows review of generated information and comparison of the results to those generated by making limited modifications to data in the PRA. Additionally, this program has the ability to graphically display the information stored in the MAR-D data base. The minor features can be categorized as data base manipulations, histograms, results display, and general capabilities.

E-3.3.1 Data Base Manipulations

Data base manipulations are used to keep the data records up-to-date. Manipulation of the data base allows the user to:

- Add, modify, or delete a family or modify associated text

- Modify the event tree data records

- Modify the system data records

- Modify the end state data records

- Modify the Basic event data records

- Modify the attribute data for the basic events

- Modify the graphic name, description, drawing (diagram), and logic for graphics including piping and instrumentation diagrams (P&ID), fault trees, and event trees

- Perform base case updates (overwrite the base case data values stored in the data base with the current data values for each basic event).

E-3.3.2 Histograms

The histogram option allows the analyst to input his own distribution for a variable that can not be expressed with one of the predefined distribution types. The histogram function allows the user to:

- Use either a range or percentage format

- Add a percentage histogram to the data base

- Add a range histogram to the data base

- Modify a currently existing histogram

- Delete histograms.

E-3.3.3 Results Display

The results display present the results for the various SARA 4.0 modules in a suitable format for reports. The results display allows the user to:

- Get general or field specific help with the <F1> key on all forms where this key is an option

- Mark or remove mark from highlighted name with the <F2> key on any form that displays a list of names

- Mark all entries with the <F3> key on any form that displays a list of names if no entries are marked and clear all marks from entries if any entry is marked

- Mark or remove mark from a range of entries with the <F4> key

- Locate function with <F5> key on any forms that display a list of names and show <F5> as an option

- Return from an operation without performing any functions with the <ESC> key

- Generate reports on screen or to the printer of families and their descriptions; basic event overviews, probabilities, and uncertainties; system and sequence summaries, combinations and logic; cut set data on base case and alternate case; system importance; subsystem data; event tree rules, logic, and initiating events; and system cut sets, the percentage contribution of each cut set to the system, the frequency of the cut set to the system, the frequency of the cut set, and the event names which make up the cut set

- Generate a graphical analysis which allows viewing plotting of P&ID graphics, viewing and plotting of fault tree diagrams, viewing and plotting of event tree diagrams, making basic events plots, making system plots, making sequence plots, making end state plots, and making family plots.

E-3.3.4 General Capabilities

The SARA 4.0 program has many general capabilities to make the users experience easier. The general capabilities allow the user to:

- Get general or field specific help with the <F1> key on all forms where this key is an option

- Mark or remove mark from highlighted name with the <F2> key on any form that displays a list of names

- Mark all entries with the <F3> key on any form that displays a list of names if no entries are marked and clear all marks from entries if any entry is marked

- Mark or remove mark from a range of entries with the <F4> key

- Locate function with <F5> key on any forms that display a list of names and show <F5> as an option

- Return from an operation without performing any functions with the <ESC> key

- Generate error messages to inform the user of incorrect data, invalid option selection, or failure of an operation

- Generate warning messages to inform the user of potential loss of data if chosen option is completed.

E-3.4 FEP 4.0

FEP provides a common access to the suite of graphical tools developed for performing risk assessment. The tools are the graphical event tree, fault tree, and P&ID editors. The event tree and fault tree editors are available through IRRAS; however, the P&ID editor is only accessible through FEP. The FEP tool allows using a less complex tool when the user only wants to generate graphical displays. The minor features can be categorized as fault tree analysis, event tree analysis, P&ID construction and general capabilities.

Appendix E

E-3.4.1 Fault Tree Analysis

The minor features contained within fault tree analysis are fault tree construction and plot trees.

Fault Tree Construction is accomplished in a graphical environment with the use of a mouse and graphical symbols. The graphical symbols are listed in a pull down menu and can be selected by clicking the left mouse button on the desired symbol. The analyst has the ability to:

- Create a fault tree by using the fault tree symbols in the "BILD" pull down menu

- Load a fault tree in the "PID" format

- Edit a fault tree, including moving a portion of the tree diagram to a new location, copying a portion of the tree diagram and moving it to a new location, deleting any portion of the displayed fault tree, duplicating a portion of the fault tree diagram, copying text from one area of the diagram to another, moving text from one area of the diagram to another, editing fault tree text

- Save a fault tree in the "PID" format

- Add text to a fault tree, including the ability to add text anywhere on the fault tree, adding a name or replacing a name

- View a fault tree, including shifting the figure up, scrolling the figure, zooming in, zooming out, restoring the original figure, showing the name of gates and events, and showing the grid

- Format fault tree diagrams for a Hewlett Packard laser printer, includes sending the diagram to the laser printer

- Display the current diagram file name

- Change attributes of gate and event names, including changing the color and text size of the name assigning a default name for gates and/or events

- Change the attributes of the text, including setting the justification for the diagrams, setting a default text size, and setting a default line type.

FAULT TREES from the "HP PLOTTER MENU" under the Utility Menu gives the analyst the ability to send a specific file generated under FEP to the plotter.

E-3.4.2 Event Tree Analysis

The minor features contained within event tree analysis are event tree construction and plot trees.

Event Tree Construction is accomplished in a graphical environment with the use of a mouse and "Tops" and "Ends" selections from the pull down menu. The "Tops" or Ends" are selected by clicking the left mouse button on the desired selection. The analyst has the ability to:

- Create an event tree by using the event tree symbols in the "Tops" and/or "Ends" pull down menu

- Load an event tree in the "PID" format

- Edit an event tree, including adding a branch to the event tree, deleting branches form the event tree, copying existing branches of the event tree to a new location, changing a branch to a pass, separating a large drawing into separate pages

- Save an event tree in the "PID" format

- Select the "Tops" menu, including adding a new event, deleting an event, changing an event name, conserving space on the diagram, changing the displayed width of the event names, changing the text size of the event names, selecting the font type for the event names and headers, editing or adding event name descriptions, specifying the height of the event descriptions, specifying the number of lines in event descriptions, and selecting the font type for the event descriptions

- Select the "Ends" menu, including displaying information about the end state, editing the header, moving a header to another location on the screen, specifying the height or distance between branches, specifying the height of the end state text, and specifying the font type for the end state text

- Add text to an event tree, including selecting the font type for the text, writing text at any desired location on the screen, moving the selected text to a new location on the screen, copying the selected text to a new location, deleting selected text, editing text and setting various text attributes such as color, size font, and justification, setting justification for the diagram, and setting default text size for the diagrams

- View an event tree, including changing the position and size of the displayed and event tree diagram, moving the drawing up, down, right, left, zooming in, zooming out, or restoring the diagram to the original size

- Format event tree diagrams for a Hewlett Packard laser printer, includes sending the diagram to the laser printer

- View and change the name of the family and the event tree file name

- Add, delete, and modify transfer file names

- Use the link editor, including adding new rules or modifying existing rules, inserting blank lines, deleting entire selected rules, and restoring the last deleted rule

- Use any basic event as a top event.

Event Trees under the "HP Plotter Menu" from the "Utility" option gives the analyst the ability to send a specific file generated under FEP to the plotter.

E-3.4.3 FEP P&ID Editor

The minor features contained within P&ID editor are P&ID construction and plot P&IDs.

P&ID Construction is accomplished in a graphical environment with the use of a mouse and "Edit" selection from the pull down menu. The "Edit" is selected by clicking the left mouse button on the desired selection. The analyst has the ability to;

- Create a P&ID by using the "Edit" selection from the pull down menu

- Load an P&ID in the "PID" format

- Edit a P&ID, including adding a line to the P&ID, deleting lines and symbols form the P&ID, copying lines and symbols of the P&ID to a new location, and creating P&ID Symbols

- Save a P&ID in the "PID" format

- Add text to an P&ID, including selecting the font type for the text, writing text at any desired location on the screen, moving the selected text to a new location on the screen, copying the selected text to a new location, deleting selected text, editing text and setting various text attributes such as color, size font, and justification, setting justification for the diagram, and setting default text size for the diagrams

- View a P&ID, including changing the position and size of the displayed P&ID diagram, moving the drawing up, down, right, left, zooming in, zooming out, or restoring the diagram to the original size

- Format P&ID for a Hewlett Packard laser printer, includes sending the diagram to the laser printer.

P&ID under the "HP Plotter Menu" from the "Utility" option gives the analyst the ability to send a specific file generated under FEP to the plotter.

E-3.4.4 General Capabilities

The FEP 4.0 program has many general capabilities to make the users experience easier. The general capabilities allow the user to:

- Get general or field specific help with the <F1> key on all forms where this key is an option

- Mark or remove mark from highlighted name with the <F2> key on any form that displays a list of names

- Mark all entries with the <F3> key on any form that displays a list of names if no entries are marked and clear all marks from entries if any entry is marked

- Mark or remove mark from a range of entries with the <F4> key

- Locate function with <F5> key on any forms that display a list of names and show <F5> as an option

- Return from an operation without performing any functions with the <ESC> key

- Generate error messages to inform the user of incorrect data, invalid option selection, or failure of an operation

- Generate warning messages to inform the user of potential loss of data if chosen option is completed.

E-4. FEATURES NOT TO BE TESTED

The features not to be tested are in test items MAR-D and FEP. In MAR-D, the "DATA Summary" menu features will not be tested. These are composed of the "Qualify" option, the "Plant Details" option, the "PRA Summary" option and the "Core Damage Summary" option.

The features that will not be tested in test item FEP are found under the "Utility" option. The features are the "P&ID Conversion 1-4" option and the "Fault Tree Conversion 2-4" option.

E-5. APPROACH

The approach to the testing of the non-vital features for the SAPHIRE 4.0 suite of codes will be directed by the grouped features. Feature grouping will be by the action words that occur in the feature descriptions. The feature descriptions include those that have the following in their description:

- "Build" or "Construct"

- "Create"

- "Plot"

- "Load"

- "Edit"

- "Extract"

- "Save"

- "Select"

- "View"

- "Add"

- "Format"

- "Change"

- "Modify"

- "Delete"

- "Alpha to Graphics"

- "Graphical Pager"

- "Use link editor"

- "Base case updates"

- "Use range or percentage format"

- "<F#>"

- "<ESC>"

- "Generate Reports"

- "Hot Keys"

- "Generate Error Messages"

- "Generate Warning Messages."

A specific approach to testing will be taken for each feature description.

For the features that have "build" or "construct" in their description, a large, medium and small tree or figure will be created or borrowed from a completed and reviewed commercial or Department of Energy (DOE) PRA. The tree or figure will be input using the build or construct commands. The completed tree will be printed and compared to the original tree or hand drawing. Actual events in the trees will be checked under the data base section. For the P&ID function, the check of the completed drawing will be all that is done.

Appendix E

The features described with the term "plot" will be tested by plotting out large, medium and small trees or diagrams constructed in the build/construct section. By using data from a previous section the tests can be economized.

"Load" features will be tested using the large, medium and small trees or diagrams from the construct/build options as well as data bases from commercial or DOE PRAs. Data files will be constructed in the load format and loaded into the program.

To test the "Edit" features, the tester will make changes to the trees, diagrams and data bases and verify that the changes have been made.

The "Save" features will be tested by saving files of various sizes and recovering them with a load feature and verifying that the saved information is identical to the information input.

The "View" features will be tested by executing the various views for data files of various sizes.

"Add" features will be tested by adding the items from the options to various data files.

The "Format" features will be tested by exercising the different format options for various data files.

Testing the "Change" features will involve changing various data files and verifying that the changes have been made.

The "Delete" features will be tested by deleting items from data files and verifying that the deletion has been accomplished.

The "Alpha to Graphics" features will be tested by inputting gate descriptions, performing the "Alpha to Graphics" function, and ascertaining whether or not the gate descriptions have been transferred to the graphical representation of the data files (fault tree, event tree, and P&ID graphical files).

The "Graphical Pager" features will be tested by executing this function on large, medium, and small fault trees and verifying that the segmentation of the fault tree is correct.

The "Extract" features will be tested by extracting the large, medium and small trees from the data base and verifying that they are the same as the input.

The "Use Link Editor" will be tested by developing link files using the Link Editor.

To test the "Base Case Update" features, base case updates will be made to various data files and the updates verified.

"Use Range Or Percentage Format" features will be tested by developing and inputting user distributions in the "Range" and "Percentage" format and verifying that these distributions are correct.

The "<F#>" features will be tested by exercising the <F1>, <F2>, <F3>, <F4>, and the <F5> keys and ensuring that they function as described in the software specifications.

The "<ESC>" feature will be tested by exercising the key and ensuring that it functions as described in the software specifications.

The "Generate Reports" features will be tested by generating various reports on various data bases and checking the report content against the software specifications.

The "Hot Keys" features will be tested by executing the various hot keys and ensuring that they respond as described in the software specifications.

Testing the "Generate Error Messages" features involves inputting incorrect information and verifying that the correct error message defined in the software specifications is displayed.

Testing the "Generate Warning Messages" features involves inputting incorrect information and verifying that the correct warning message defined in the software specifications is displayed.

E-6. TEST ITEM PASS/FAIL CRITERIA

Pass/fail criteria for the test items are listed in the following table. The table is organized by columns. The first column has the feature group, the second column has the test feature, the third column has the test item and the fourth column has the success criteria.

Feature group	Test feature	Test item	Success criteria
Build/Construct	Fault Tree Construction	IRRAS 4.0	Able to input a fault tree **and** the tree that is input matches the source
Build/Construct	Create Event Trees	IRRAS 4.0	Able to input an event tree **and** the tree that is input matches the source
Plot	Plot (Fault) Trees	IRRAS 4.0	Fault trees are identical in content and structure to the input trees.
Plot	Plot (Event) Trees	IRRAS 4.0	Event Trees are identical in content and structure to the input trees.
Load	Load Graphic (Fault) trees	IRRAS 4.0	When a fault tree is loaded, the drawing and logic of the tree is the same as the source.
Load	Graphics Load (Event Trees)	IRRAS 4.0	When an event tree is loaded, the drawing and logic of the tree is the same as the source.
Edit	Edit Fault Trees	IRRAS 4.0	Successfully make changes using fault tree edit functions on selected trees: Attributes Move Copy Duplicate Delete Copy Text Move Text Edit Text Global Text
Edit	Edit Event Trees	IRRAS 4.0	Successfully make changes using event tree edit functions in selected trees and verify changes: Add Delete Copy Pass Page
Extract	Extract Fault Trees	IRRAS 4.0	Fault trees extracted are identical to the tree that was input.

NUREG/CR-6145

Feature group	Test feature	Test item	Success criteria
Save	Save Fault Trees	IRRAS 4.0	Saved information is identical to that input
Save	Save Event Trees	IRRAS 4.0	Saved information is identical to that input
Select	Select Family	IRRAS 4.0	Change family and verify the selected family
View	View Fault Trees	IRRAS 4.0	Successfully perform view functions on selected trees: Page ↑,↓,←,→ Scroll Zoom In Zoom Out Zoom Restore Show/No Name Show/No Text Grid
View	View Event Trees	IRRAS 4.0	Successfully perform view functions on selected trees: Page ↑,↓,←,→ Scroll Zoom In Zoom Out Zoom Restore Show/No Text Grid
0Add	Add Family Name	IRRAS 4.0	New family name is added successfully.
Format	Format Fault Trees	IRRAS 4.0	Format a diagram for Epson and laser printers and print to verify success
Format	Format Event Trees	IRRAS 4.0	Format a diagram for Epson and laser printers and print to verify success
Change	Generate Changes	IRRAS 4.0	Create new current event data for a selected uncertainty analysis
Change	Report Changes	IRRAS 4.0	Create reports to reflect the event modifications that currently exist within the data base for the three report types: 1. Unaffected Events 2. Affected Events 3. All Events
Modify	Modify Family	IRRAS 4.0	Family name and associated text is changed successfully.
Modify	Modify Event Trees	IRRAS 4.0	Event tree data records are successfully changed
Modify	Modify Systems	IRRAS 4.0	System data records are successfully changed
Modify	Modify End States	IRRAS 4.0	The end state data records are successfully changed

Feature group	Test feature	Test item	Success criteria
Modify	Modify Basic Events	IRRAS 4.0	Basic event attribute and uncertainty data is successfully changed
Modify	Modify Attributes: System Location Failure Mode Class Attributes Basic Event Type Trains	IRRAS 4.0	Data base attribute data is successfully changed
Modify	Modify Gates	IRRAS 4.0	Gate records are successfully changed.
Modify	Modify Graphics: Fault Trees Event Trees P&IDs	IRRAS 4.0	Fault tree, event tree, and P&ID graphics data records are successfully changed.
Modify	Modify Histograms	IRRAS 4.0	Histogram data is successfully changed.
Delete	Delete Family Name	IRRAS 4.0	Family name is successfully deleted after removing all files.
Alpha to Graphics	Alpha to Graphics for fault trees	IRRAS 4.0	The gate descriptions and event descriptions appear in the graphics fault tree.
Graphical Pager	Graphical Pager for Fault Trees	IRRAS 4.0	Faults trees produced by the Pager are equivalent to the source tree.
Use Link Editor	Link Editor	IRRAS 4.0	Successfully perform link editor functions on selected event trees: Add new rules; Modify existing rules; Insert blank lines; Delete selected rules; Restore last deleted rule.
Base Case Updates	Overwrite Base Case Data	IRRAS 4.0	Successfully overwrite base case data values stored in the data base with the current data values for each basic event.
Use Range or Percentage format	Range/Percentage Format	IRRAS 4.0	Perform edit histogram functions on selected histograms in the percentage and range formats: Add; Modify; Delete.
<F1>	Help	IRRAS 4.0	Display "help" message for all forms where <F1> is an option.
<F2>	Mark/Unmark	IRRAS 4.0	Mark and remove mark from highlighted names on all forms where <F2> is an option.

Feature group	Test feature	Test item	Success criteria
<F3>	Mark All	IRRAS 4.0	Mark all entries and remove mark from all entries in all forms where <F3> is an option.
<F4>	Mark Range	IRRAS 4.0	Mark and remove mark from a range of entries on all forms where <F4> is an option.
<F5>	Locate	IRRAS 4.0	Locate a selected sequence on all forms where <F5> is an option.
<ESC>	Escape	IRRAS 4.0	Return from selected operations using the <ESC> key.
Generate Reports	Report on Data Base	IRRAS 4.0	Generate reports on the screen and to the printer for; Families and their descriptions; Basic event overviews, probabilities, and uncertainties; System and sequence summaries, logic, cut sets
Hot Keys	<Alt/E>	IRRAS 4.0	Perform the main menu function, "Exit IRRAS 4.0" without returning to the main menu.
Hot Keys	<Alt/A>	IRRAS 4.0	Perform the main menu function, "Analyze Sequences" without returning to the main menu.
Hot Keys	<Alt/B>	IRRAS 4.0	Perform the main menu function, "Build Fault Trees" without returning to the main menu.
Hot Keys	<Alt/C>	IRRAS 4.0	Perform the main menu function, "Create Event Trees" without returning to the main menu.
Hot Keys	<Alt/F>	IRRAS 4.0	Perform the main menu function, "Fault Tree Analysis" without returning to the main menu.
Hot Keys	<Alt/G>	IRRAS 4.0	Perform the main menu function, "Generate Event Data" without returning to the main menu.
Hot Keys	<Alt/M>	IRRAS 4.0	Perform the main menu function, "Modify Data Base" without returning to the main menu.
Hot Keys	<Alt/R>	IRRAS 4.0	Perform the main menu function, "Report on Data Base" without returning to the main menu.
Hot Keys	<Alt/S>	IRRAS 4.0	Perform the main menu function, "Select Family" without returning to the main menu.

Feature group	Test feature	Test item	Success criteria
Hot Keys	<Alt/U>	IRRAS 4.0	Perform the main menu function, "Utility Options" without returning to the main menu.
Generate Error Messages	"Plotter Not Attached"	IRRAS 4.0	Error message is displayed.
Generate Error Messages	"Invalid Non-Alpha Key"	IRRAS 4.0	Error message is displayed.
Generate Error Messages	"Incorrect Data"	IRRAS 4.0	Error message is displayed.
Generate Error Messages	"Selected Option Not Available"	IRRAS 4.0	Error message is displayed.
Generate Warning Messages	"Loss of Data"	IRRAS 4.0	Warning message is displayed.
Load	Load MAR-D: Family Attributes Basic Event System Event Tree Endstate Sequence Gate Change Sets	MAR-D 4.0	Successfully load a data file of the appropriate data type to the data base.
Load	Load SETS: Basic Event System Sequence	MAR-D 4.0	Successfully load a data file to the currently selected family.
Edit	Edit Families	MAR-D 4.0	Successfully make changes to families using the family edit function and verify changes: Add Modify Delete Text
Edit	Edit Event Trees	MAR-D 4.0	Successfully make changes using event tree edit functions in selected trees and verify changes: Add Modify Delete Text Sequences Base Case Update Clear Alternate Case

Appendix E

Feature group	Test feature	Test item	Success criteria
Edit	Edit Systems	MAR-D 4.0	Successfully make changes to data base using edit functions and verify changes: Add Modify Delete Text Sequences Base Case Update Clear Alternate Case
Edit	Edit End States	MAR-D 4.0	Successfully make changes to data base using edit functions and verify changes: Add Modify Delete Text
Edit	Edit Basic Events	MAR-D 4.0	Successfully make changes to data base using edit functions and verify changes: Add Modify Delete Remove Unused Events
Edit	Edit Attributes: System Location Failure Mode Class Attributes Basic Event Type Trains	MAR-D 4.0	Successfully make changes to data base using edit functions and verify changes: Add Modify Delete
Edit	Edit Gates	MAR-D 4.0	Successfully make changes to data base using edit functions and verify changes: Add Modify Delete
Edit	Edit Graphics: Fault Tree Event Tree P&ID	MAR-D 4.0	Successfully make changes to data base using edit functions and verify changes: Add Modify Delete
Edit	Edit Histograms	MAR-D 4.0	Successfully make changes to data base using edit functions and verify changes: Add Modify Delete

Feature group	Test feature	Test item	Success criteria
Extract	Extract MAR-D Data: Family Attributes Basic Event System Event Tree Endstate Sequence Gate Change Sets	MAR-D 4.0	Successfully extract data from data base using the "Extract Data" function and verify identical to the source.
Extract	Extract SETS Data: Basic Event System Sequence	MAR-D 4.0	Successfully extract data from data base using the "Extract Data" function and verify identical to the source.
Extract	Extract FRANTIC Data: Basic Event System Sequence	MAR-D 4.0	Successfully extract data from data base using the "Extract Data" function and verify identical to the source.
Select	Select Family	MAR-D 4.0	Change family and verify the selected family
Add	Add: Family Event Trees Systems End States Basic Events Attributes Gates Graphics Histograms	MAR-D 4.0	New data is added successfully.
Modify	Modify Family	MAR-D 4.0	Family name and associated text is changed successfully.
Modify	Modify Event Trees	MAR-D 4.0	Event tree data records are successfully changed
Modify	Modify Systems	MAR-D 4.0	System data records are successfully changed
Modify	Modify End States	MAR-D 4.0	The end state data records are successfully changed
Modify	Modify Basic Events	MAR-D 4.0	Basic event attribute and uncertainty data is successfully changed

Appendix E

Feature group	Test feature	Test item	Success criteria
Modify	Modify Attributes: System Location Failure Mode Class Attributes Basic Event Type Trains	MAR-D 4.0	Data base attribute data is successfully changed
Modify	Modify Gates	MAR-D 4.0	Gate records are successfully changed.
Modify	Modify Graphics: Fault Trees Event Trees P&IDs	MAR-D 4.0	Fault tree, event tree, and P&ID graphics data records are successfully changed.
Modify	Modify Histograms	MAR-D 4.0	Histogram data is successfully changed.
Delete	Delete: Family Event Trees Systems End States Basic Events Attributes Gates Graphics Histograms	MAR-D 4.0	Data is successfully deleted.
Base Case Updates	Overwrite Base Case Data	MAR-D 4.0	Successfully overwrite base case data values stored in the data base with the current data values for each basic event.
Use Range or Percentage format	Range/Percentage Format	MAR-D 4.0	Perform edit histogram functions on selected histograms in the percentage and range formats: Add; Modify; Delete.
<F1>	Help	MAR-D 4.0	Display "help" message for all forms where <F1> is an option.
<F2>	Mark/Unmark	MAR-D 4.0	Mark and remove mark from highlighted names on all forms where <F2> is an option.
<F3>	Mark All	MAR-D 4.0	Mark all entries and remove mark from all entries in all forms where <F3> is an option.
<F4>	Mark Range	MAR-D 4.0	Mark and remove mark from a range of entries on all forms where <F4> is an option.
<F5>	Locate	MAR-D 4.0	Locate a selected sequence on all forms where <F5> is an option.

Feature group	Test feature	Test item	Success criteria
<ESC>	Escape	MAR-D 4.0	Return from selected operations using the <ESC> key.
Reports	Summary Reports	MAR-D 4.0	Generate a summary report on the screen to the printer, and send to a file.
Reports	Detail Reports	MAR-D 4.0	Generate a detail report on the screen to the printer, and send to a file.
Generate Error Messages	"Plotter Not Attached"	MAR-D 4.0	Error message is displayed.
Generate Error Messages	"Invalid Non-Alpha Key"	MAR-D 4.0	Error message is displayed.
Generate Error Messages	"Incorrect Data"	MAR-D 4.0	Error message is displayed.
Generate Error Messages	"Selected Option Not Available"	MAR-D 4.0	Error message is displayed.
Generate Warning Messages	"Loss of Data"	MAR-D 4.0	Warning message is displayed.
Plot	Plots: Basic Event System Sequence End State Family	SARA 4.0	Plots match data input.
Edit	Edit Family	SARA 4.0	Successfully make changes using family edit functions and verify changes: Add Modify Delete Text
Edit	Edit Event Trees	SARA 4.0	Successfully make changes using event tree edit functions in selected trees and verify changes: Add Modify Delete Text Sequences Base Case Update Clear Alternate Case
Edit	Edit Systems	SARA 4.0	Successfully make changes using edit functions and verify changes: Add Modify Delete Text

Appendix E

Feature group	Test feature	Test item	Success criteria
Edit	Edit End States	SARA 4.0	Successfully make changes using edit functions and verify changes: Add Modify Delete Text Base Case Update Clear Alternate Case
Edit	Edit Basic Events	SARA 4.0	Successfully make changes using edit functions and verify changes: Add Modify Delete Remove Unused Events
Edit	Edit Attributes: System Location Failure Mode Class Attributes Basic Event Type Trains	SARA 4.0	Successfully make changes using edit functions and verify changes: Add Modify Delete
Edit	Edit Gates	SARA 4.0	Successfully make changes using edit functions and verify changes: Add Modify Delete
Edit	Edit Graphics: Fault Trees Event Trees P&ID Tree	SARA 4.0	Successfully make changes using edit functions and verify changes: Add Modify Delete
Edit	Edit Histograms	SARA 4.0	Successfully make changes using edit functions and verify changes: Add Modify Delete
Select	Select Family	SARA 4.0	Change family and verify the selected family
Display	Display: P&ID Interface Fault Tree Event Tree	SARA 4.0	Successfully display selected graphical diagrams.
Add	Add Family Name	SARA 4.0	New family name is added successfully.
Add	Add Change Sets	SARA 4.0	Change sets are added successfully.
Add	Add Cutsets	SARA 4.0	Cutsets are successfully added.

Feature group	Test feature	Test item	Success criteria
Change	Generate Changes	SARA 4.0	Create new current event data for a selected uncertainty analysis
Change	Report Changes	SARA 4.0	Create reports to reflect the event modifications that currently exist within the data base for the three report types: 1. Unaffected Events 2. Affected Events 3. All Events
Modify	Modify Family	SARA 4.0	Family name and associated text is changed successfully.
Modify	Modify Event Trees	SARA 4.0	Event tree data records are successfully changed
Modify	Modify Systems	SARA 4.0	System data records are successfully changed
Modify	Modify End States	SARA 4.0	The end state data records are successfully changed
Modify	Modify Basic Events	SARA 4.0	Basic event attribute and uncertainty data is successfully changed
Modify	Modify Attributes: System Location Failure Mode Class Attributes Basic Event Type Trains	SARA 4.0	Data base attribute data is successfully changed
Modify	Modify Gates	SARA 4.0	Gate records are successfully changed.
Modify	Modify Graphics: Fault Trees Event Trees P&IDs	SARA 4.0	Fault tree, event tree, and P&ID graphics data records are successfully changed.
Modify	Modify Histograms	SARA 4.0	Histogram data is successfully changed.
Modify	Modify Event Data	SARA 4.0	Successfully modify the identification fields of the change set.
Modify	Modify Cutsets	SARA 4.0	Changed cutsets match the source.
Delete	Delete Family Name	SARA 4.0	Family name is successfully deleted after removing all files.
Delete	Delete cutsets	SARA 4.0	Cut sets are successfully deleted.
Delete	Delete Change Sets	SARA 4.0	Change sets are successfully deleted.
Base Case Updates	Overwrite Base Case Data	SARA 4.0	Successfully overwrite base case data values stored in the data base with the current data values for each basic event.

Feature group	Test feature	Test item	Success criteria
Use Range or Percentage format	Range/Percentage Format	SARA 4.0	Perform edit histogram functions on selected histograms in the percentage and range formats: Add; Modify; Delete.
\<F1\>	Help	SARA 4.0	Display "help" message for all forms where \<F1\> is an option.
\<F2\>	Mark/Unmark	SARA 4.0	Mark and remove mark from highlighted names on all forms where \<F2\> is an option.
\<F3\>	Mark All	SARA 4.0	Mark all entries and remove mark from all entries in all forms where \<F3\> is an option.
\<F4\>	Mark Range	SARA 4.0	Mark and remove mark from a range of entries on all forms where \<F4\> is an option.
\<F5\>	Locate	SARA 4.0	Locate a selected sequence on all forms where \<F5\> is an option.
\<ESC\>	Escape	SARA 4.0	Return from selected operations using the \<ESC\> key.
Generate Reports	Reports	SARA 4.0	Generate reports on the screen and to the printer for; Families and their descriptions; Event Trees; Basic event overviews, probabilities, and uncertainties; System and sequence summaries.
Generate Error Messages	"Plotter Not Attached"	SARA 4.0	Error message is displayed.
Generate Error Messages	"Invalid Non-Alpha Key"	SARA 4.0	Error message is displayed.
Generate Error Messages	"Incorrect Data"	SARA 4.0	Error message is displayed.
Generate Error Messages	"Selected Option Not Available"	SARA 4.0	Error message is displayed.
Generate Warning Messages	"Loss of Data"	SARA 4.0	Warning message is displayed.
Build/Construct	Fault Tree Construction	FEP 4.0	Able to input a fault tree and the tree that is input matches the source
Build/Construct	Create Event Trees	FEP 4.0	Able to input an event tree and the tree that is input matches the source
Plot	Plot (Fault) Trees	FEP 4.0	Fault trees are identical in content and structure to the input trees.

Feature group	Test feature	Test item	Success criteria
Plot	Plot (Event) Trees	FEP 4.0	Event Trees are identical in content and structure to the input trees.
Plot	Plot P&ID Drawings	FEP 4.0	P&ID Drawings are identical in content and structure to the input trees.
Load	Load Fault Trees	FEP 4.0	When a fault tree is loaded, the drawing and logic of the tree is the same as the source.
Load	Load Event Trees	FEP 4.0	When an event tree is loaded, the drawing and logic of the tree is the same as the source.
Load	Load P&IDs	FEP 4.0	When a P&ID diagram is loaded, the diagram and logic if the diagram is the same as the source.
Edit	Edit Fault Trees	FEP 4.0	Successfully make changes using fault tree edit functions on selected trees: Attributes Move Copy Duplicate Delete Copy Text Move Text Edit Text Global Text
Edit	Edit Event Trees	FEP 4.0	Successfully make changes using event tree edit functions in selected trees and verify changes: Add Delete Copy Pass Page
Edit	Edit P&IDs	FEP 4.0	Successfully make changes using P&ID edit functions in selected diagrams and verify changes: Delete Copy Move Scale Rotate ↔ Flip ↑↓ Flip Adjust Symbols Line Attributes

Feature group	Test feature	Test item	Success criteria
Extract	Extract Fault Trees	FEP 4.0	Fault trees are extracted and are identical to the source trees.
			Extracted fault trees are successfully deleted.
Extract	Extract Event Trees	FEP 4.0	Event trees are extracted and are identical to the source trees.
			Extracted event trees are successfully deleted.
Extract	Extract P&ID Diagrams	FEP 4.0	P&ID diagrams are extracted and are identical to the source diagrams.
			Extracted P&ID diagrams are successfully deleted;
Save	Save Fault Trees	FEP 4.0	Saved information is identical to that input
Save	Save Event Trees	FEP 4.0	Saved information is identical to that input
Save	Save P&IDs	Fep 4.0	Saved information is identical to that input
Select	Select Family	FEP 4.0	Change family and verify the selected family
View	View Fault Trees	FEP 4.0	Successfully perform view functions on selected trees: Page $\uparrow,\downarrow,\leftarrow,\rightarrow$ Scroll Zoom In Zoom Out Zoom Restore Show/No Name Show/No Text Grid
View	View Event Trees	FEP 4.0	Successfully perform view functions on selected trees: Page $\uparrow,\downarrow,\leftarrow,\rightarrow$ Scroll Zoom In Zoom Out Zoom Restore Show/No Text Grid
View	View P&IDs	FEP 4.0	Successfully perform view functions on selected diagrams: Page $\uparrow,\downarrow,\leftarrow,\rightarrow$ Scroll Zoom In Zoom Out Zoom Restore Zoom Previous Page

Feature group	Test feature	Test item	Success criteria
Add	Add Event Tree Events	FEP 4.0	New events are added successfully added.
Format	Format Fault Trees	FEP 4.0	Format a diagram for Epson and laser printers and print to verify success
Format	Format Event Trees	FEP 4.0	Format a diagram for Epson and laser printers and print to verify success
Modify	Modify Event Trees	FEP 4.0	Event tree graphics are successfully changed
Delete	Delete Fault Trees	FEP 4.0	All or part of selected fault trees are successfully deleted.
Delete	Delete Event Trees	FEP 4.0	All or part of selected fault trees are successfully deleted.
Delete	Delete P&IDs	FEP 4.0	All or part of selected diagrams are successfully deleted.
Alpha to Graphics	Alpha to Graphics for fault trees	FEP 4.0	The gate descriptions and event descriptions appear in the graphics fault tree.
<F1>	Help	FEP 4.0	Display "help" message for all forms where <F1> is an option.
<F2>	Mark/Unmark	FEP 4.0	Mark and remove mark from highlighted names on all forms where <F2> is an option.
<F3>	Mark All	FEP 4.0	Mark all entries and remove mark from all entries in all forms where <F3> is an option.
<F4>	Mark Range	FEP 4.0	Mark and remove mark from a range of entries on all forms where <F4> is an option.
<F5>	Locate	FEP 4.0	Locate a selected sequence on all forms where <F5> is an option.
<ESC>	Escape	FEP 4.0	Return from selected operations using the <ESC> key.
Generate Error Messages	"Plotter Not Attached"	FEP 4.0	Error message is displayed.
Generate Error Messages	"Invalid Non-Alpha Key"	FEP 4.0	Error message is displayed.
Generate Error Messages	"Incorrect Data"	FEP 4.0	Error message is displayed.
Generate Error Messages	"Selected Option Not Available"	FEP 4.0	Error message is displayed.
Generate Warning Messages	"Loss of Data"	FEP 4.0	Warning message is displayed.

Appendix E

1. K. D. Russell et al., *Integrated Reliability and Risk Analysis System (IRRAS) Version 4.0, Volume 1: Reference Manual (Draft)*, NUREG/CR-5813, EGG-2664, December 1991.

2. K. D. Russell et al., *Systems Analysis and Risk Assessment System (SARA) Version 4.0, Volume 1: Reference Manual (Draft)*, NUREG/CR-5303, EGG-2668, December 1991.

3. S. T. Wood and K. D. Russell, *Integrated Reliability and Risk Analysis System (IRRAS), Version 4.0, Functional Requirements Document (Draft)*, Under DOE Contract No. DE-AC07-76ID01570, EG&G Idaho, Inc., Idaho Falls, Idaho, December 1991.

4. IEEE Std 829-1983, "IEEE Standard For Software Documentation," Institute of Electrical and Electronic Engineers.

Appendix F

Non-Vital Features
Verification and Validation Test Procedure

CONTENTS

FIGURES

TABLES

Appendix F

Non-Vital Features
Verification and Validation Test Procedure

F-1. INTRODUCTION

The Idaho National Engineering Laboratory (INEL) is developing a computer code for the U.S. Nuclear Regulatory Commission (NRC). The System Analysis Programs for Hands-on Integrated Reliability Evaluation (SAPHIRE), Version 4.0 (Russell et al. 1992) is a probabilistic risk assessment (PRA) tool used to address key nuclear plant safety issues. The role of the INEL in the SAPHIRE program is to develop the software and to communicate with the PRA community, including technology transfer.

The INEL has been developing the SAPHIRE system [formerly titled the Integrated Reliability and Risk Analysis System (IRRAS)] since 1987, and has continued to make the system more powerful and user friendly by incorporating user comments and making other code enhancements. IRRAS Version 1.0 was release in February 1987 to prove the concept of performing PRAs on microcomputers. Version 1.0 contained many of the basic features required for fault tree analysis. The other versions of SAPHIRE are Version 2.0 released officially in June 1990 and Version 2.5 released in August 1991. The latest version of SAPHIRE is designated 4.0.

SAPHIRE has all the necessary capabilities and functions required to create, modify, reduce, and analyze fault tree models used in the analysis of complex systems and processes. The program uses advanced graphic and analytical techniques to maximize the potential found in today's microcomputers. Version 4.0 provides all the capabilities of Version 1.0 plus a relational database for managing the data, improved functionality, and improved algorithm performance.

Several programs are integrated in SAPHIRE to allow the user to create and analyze fault trees, event trees, and accident sequences on a microcomputer. The programs included are

- IRRAS (Russell et al. 1992)

- Systems Analysis and Risk Assessment (SARA) (Russell et al. 1991)

- Fault Tree, Event Tree, and Piping and Instrumentation Diagram (FEP) graphical editor (Russell et al. 1992)

- Models and Results Database (MAR-D) (Russell et al. 1992).

Each of these programs performs a specific function in the development of a PRA.

SAPHIRE is scheduled to be a production model for a variety of organizations; therefore, a formal verification and validation (V&V) must be documented to ensure quality assurance. Quality assurance is necessary to ensure that the information produced by the code is correct. Part of the formal V&V effort is to construct test procedure specifications. Test procedure specifications are the procedures that are performed to confirm that the features of the code are functioning correctly. This document contains the procedures to test the nonvital features of SAPHIRE 4.0. Table F-1 identifies the various feature groups,

Appendix F

the specific features to be tested, the program that contains the feature, and the success criteria for the items that are tested in this test procedure specification.

In this document, the steps to test the primary programs (IRRAS, FEP, MAR-D, and SARA) and special functions (i.e., hot keys) of SAPHIRE are grouped into sections called Procedures. Procedure 1 contains the steps to test the nonvital fault tree features of the IRRAS program. The steps to test the event tree nonvital features of IRRAS are in Procedure 2. The steps to test the nonvital features associated with FEP are provided in Procedure 3. Procedure 4 contains the steps to test the nonvital features of MAR-D. Within each Procedure, the steps that are used to test the features of a specific program or special function are divided into sections. The steps in each section should be followed and if any step is not completed successfully, an anomaly report should be generated and submitted to the SAPHIRE 4.0 V&V committee.

Table F-1. SAPHIRE 4.0 V&V test procedure specifications.

Feature group	Test feature	Test item(s)	Success criteria	Procedure section
Build graphic trees	Fault tree construction	IRRAS 4.0 FEP 4.0	Able to input a fault tree and the tree that is input matches the source	IVVPRO-1 2
Create event trees	Create event trees	IRRAS 4.0 FEP 4.0	Able to input an event tree and the tree that is input matches the source	IVVPRO-2 1
Plot trees	Plot (fault) trees	IRRAS 4.0	Fault trees are identical in content and structure to the input trees	IVVPRO-1 2
Plot trees	Plot (event) trees	IRRAS 4.0	Event trees are identical in content and structure to the input trees	IVVPRO-2 2
Load graphic trees	Load graphic (fault) trees	IRRAS 4.0	When a fault tree is loaded, the drawing and logic of the tree is the same as the source	IVVPRO-1 6
Graphics load	Graphics load (event trees)	IRRAS 4.0	When an event tree is loaded, the drawing and logic of the tree is the same as the source	IVVPRO-2 6
Buirr graphic trees	Edit fault trees	IRRAS 4.0 FEP 4.0	Successfully make changes using fault tree edit functions on selected trees: Attributes Move Copy Duplicate Delete Copy text Move text Edit text Global text	IVVPRO-1 4
Create event Trees	Edit event trees	IRRAS 4.0 FEP 4.0	Successfully make changes using event tree edit functions and verify changes in selected trees: Add Delete Copy Pass Page	IVVPRO-2 1
Extract fault trees	Extract fault trees	IRRAS 4.0	Fault trees extracted are identical to the tree that was input.	IVVPRO-1 7

Table F-1. SAPHIRE 4.0 V&V test procedure specifications.

Feature group	Test feature	Test item(s)	Success criteria	Procedure section
Build graphic trees	Save fault trees	IRRAS 4.0 FEP 4.0	Saved information is identical to that input	IVVPRO-1 2
Create event trees	Save event trees	IRRAS 4.0 FEP 4.0	Saved information is identical to that input	IVVPRO-2 1
Select	Select family	IRRAS 4.0 SARA 4.0 MAR-D 4.0 FEP 4.0	Change family and verify the selected family	IVVPRO-1 1
Build graphic trees	View fault trees	IRRAS 4.0 SARA 4.0 FEP 4.0	Successfully perform view functions on selected trees: Page $\uparrow,\downarrow,\leftarrow,\rightarrow$ Scroll Zoom in Zoom out Zoom restore Show/no Name Show/no Text Grid	IVVPRO-1 4
Create event trees	View event trees	IRRAS 4.0 SARA 4.0 FEP 4.0	Successfully perform view functions on selected trees: Page $\uparrow,\downarrow,\leftarrow,\rightarrow$ Scroll Zoom in Zoom out Zoom restore Show/no text Grid	IVVPRO-1 5
Modify data base	Add family name	IRRAS 4.0 SARA 4.0 MAR-D 4.0	New family name is added successfully	IVVPRO-1 1
Plot trees	Format fault trees	IRRAS 4.0	Format a diagram for Epson and laser printers and print to verify success	IVVPRO-1 2
Plot trees	Format event trees	IRRAS 4.0	Format a diagram for Epson and laser printers and print to verify success	IVVPRO-2 2
Modify data base	Modify family	IRRAS 4.0 SARA 4.0 MAR-D 4.0	Family name and associated text is changed successfully	IVVPRO-1 1
Modify data base	Modify event trees	IRRAS 4.0 SARA 4.0 MAR-D 4.0	Event tree data records are successfully changed	IVVPRO-4 3
Modify data base	Modify systems	IRRAS 4.0 SARA 4.0 MAR-D 4.0	System data records are successfully changed	IVVPRO-4 3
Modify data base	Modify end states	IRRAS 4.0 SARA 4.0 MAR-D 4.0	The endstate data records are successfully changed	IVVPRO-4 4

NUREG/CR-6145

Appendix F

Table F-1. SAPHIRE 4.0 V&V test procedure specifications.

Feature group	Test feature	Test item(s)	Success criteria	Procedure section
Modify data base	Modify basic events	IRRAS 4.0 SARA 4.0 MAR-D 4.0	Basic event attribute and uncertainty data are successfully changed	IVVPRO-4 4
Modify data base	Modify Attributes: System Location Failure mode Class attributes Basic event type Trains	IRRAS 4.0 SARA 4.0 MAR-D 4.0	Database attribute data are successfully changed	IVVPRO-4 4
Modify data base	Modify gates	IRRAS 4.0 SARA 4.0 MAR-D 4.0	Gate records are successfully changed	IVVPRO-4 5
Modify data base	Modify graphics: Fault trees Event trees P&IDs	IRRAS 4.0 SARA 4.0 MAR-D 4.0	Fault tree, event tree, and P&ID graphics data records are successfully changed	IVVPRO-4 5
Modify data base	Delete family name	IRRAS 4.0 SARA 4.0 MAR-D 4.0	Family name is successfully deleted after removing all files	IVVPRO-1 1
Alpha to graphics	Alpha to graphics for fault trees	IRRAS 4.0 FEP 4.0	The gate descriptions and event descriptions appear in the graphics fault tree	IVVPRO-1 6
Graphical pager	Page fault trees	IRRAS 4.0	Faults trees produced by the Pager are equivalent to the source tree	IVVPRO-1 7
Link event trees	Link editor	IRRAS 4.0 SARA	Successfully perform link editor functions on selected event trees: Add new rules Modify existing rules Insert blank lines Delete selected rules Restore last deleted rule	IVVPRO-2 6
Function keys	<F1> help	IRRAS 4.0 SARA 4.0 MAR-D 4.0 FEP 4.0	Display "help" message for all forms where <F1> is an option	TBD
Function keys	<F2> mark/unmark	IRRAS 4.0 SARA 4.0 MAR-D 4.0 FEP 4.0	Mark and remove a mark from highlighted names on all forms where <F2> is an option	TBD
Function keys	<F3> mark all	IRRAS 4.0 SARA 4.0 MAR-D 4.0 FEP 4.0	Mark all entries and remove a mark from all entries in all forms where <F3> is an option	TBD
Function keys	<F4> mark range	IRRAS 4.0 SARA 4.0 MAR-D 4.0 FEP 4.0	Mark and remove a mark from a range of entries on all forms where <F4> is an option	TBD

Table F-1. SAPHIRE 4.0 V&V test procedure specifications.

Feature group	Test feature	Test item(s)	Success criteria	Procedure section
Function keys	<F5> locate	IRRAS 4.0 SARA 4.0 MAR-D 4.0 FEP 4.0	Locate a selected sequence on all forms where <F5> is an option	TBD
Function keys	<ESC> escape	IRRAS 4.0 SARA 4.0 MAR-D 4.0 FEP 4.0	Return from selected operations using the <ESC> key	TBD
Reports	Report on data base	IRRAS 4.0 SARA 4.0 MAR-D 4.0	Generate reports on the screen and to the printer for Families and their descriptions Basic event overviews, probabilities, and uncertainties System and sequence summaries, logic, cutsets	IVVPRO-4 6
Hot keys	<Alt/E>	IRRAS 4.0	Perform the main menu function, "Exit IRRAS 4.0" without returning to the main menu	TBD
Hot keys	<Alt/A>	IRRAS 4.0	Perform the main menu function "Analyze Sequences" without returning to the main menu	TBD
Hot keys	<Alt/B>	IRRAS 4.0	Perform the main menu function "Build Fault Trees" without returning to the main menu	TBD
Hot keys	<Alt/C>	IRRAS 4.0	Perform the main menu function "Create Event Trees" without returning to the main menu	TBD
Hot keys	<Alt/F>	IRRAS 4.0	Perform the main menu function "Fault Tree Analysis" without returning to the main menu	TBD
Hot keys	<Alt/G>	IRRAS 4.0	Perform the main menu function "Generate Event Data" without returning to the main menu	TBD
Hot keys	<Alt/M>	IRRAS 4.0	Perform the main menu function "Modify Data-base" without returning to the main menu	TBD
Hot keys	<Alt/R>	IRRAS 4.0	Perform the main menu function "Report on Data-base" without returning to the main menu	TBD
Hot keys	<Alt/S>	IRRAS 4.0	Perform the main menu function "Select Family" without returning to the main menu	TBD
Hot keys	<Alt/U>	IRRAS 4.0	Perform the main menu function "Utility Options" without returning to the main menu	TBD

Appendix F

Table F-1. SAPHIRE 4.0 V&V test procedure specifications.

Feature group	Test feature	Test item(s)	Success criteria	Procedure section
Load	Load MAR-D: Family Attributes Basic event System Event tree Endstate Sequence Gate Change sets	IRRAS 4.0 SARA 4.0 MAR-D 4.0	Successfully load a data file of the appropriate data type to the data base	IVVPRO-4 1
Extract	Extract MAR-D Data: Family Attributes Basic event System Event tree Endstate Sequence Gate Change sets	IRRAS 4.0 MAR-D 4.0	Successfully extract data from data base using the "Extract Data" function and verify that data are identical to the source	IVVPRO-4 2
Graphics analysis	Plots: Basic event System Sequence End state Family	SARA 4.0	Plots match data input	TBD
Graphics analysis	Display: P&ID interface Fault tree Event tree	SARA 4.0	Successfully display selected graphical diagrams	TBD
Utility	Plot P&ID drawings	FEP 4.0	P&ID drawings are identical in content and structure to the input trees	IVVPRO-3 2
P&ID Editor	Load P&IDs	FEP 4.0	When a P&ID diagram is loaded, the diagram and logic of the diagram is the same as the source	IVVPRO-3 4
P&ID editor	Edit P&IDs	FEP 4.0	Successfully make changes using P&ID edit functions in selected diagrams and verify changes: Delete Copy Move Scale Rotate ↔ Flip ↑↓ Flip Adjust Symbols Line Attributes	IVVPRO-3 1

Table F-1. SAPHIRE 4.0 V&V test procedure specifications.

Feature group	Test feature	Test item(s)	Success criteria	Procedure section
Utility	Extract P&ID diagrams	FEP 4.0	P&ID diagrams are extracted and are identical to the source diagrams	IVVPRO-3 8
			Extracted P&ID diagrams are successfully deleted	
P&ID editor	Save P&IDs	FEP 4.0	Saved information is identical to that input	IVVPRO-3 4
P&ID editor	View P&IDs	FEP 4.0	Successfully perform view functions on selected diagrams: Page ↑,↓,←,→ Scroll Zoom in Zoom out Zoom restore Zoom previous page	IVVPRO-3 5
P&ID editor	Delete P&IDs	FEP 4.0	All or part of selected diagrams are successfully deleted	IVVPRO-3 1

F-2. HARDWARE REQUIREMENTS

SAPHIRE 4.0 system requires the following hardware configurations:

- IBM-PC/XT/AT, PS2, or 100% compatible with 640K of RAM

- DOS 3.3 or later

- 20 Mb hard disk (minimum)

- Math coprocessor (optional)

- 16 color enhanced monitor (EGA or better)

- Graphics input devices — keyboard or mouse.

If an enhanced graphics adapter is used, it must have the memory expansion option to extend the standard 4 colors to 16. This option is an upgrade to the IBM EGA board, but is usually standard on boards manufactured by other vendors. SAPHIRE 4.0 *does not* support the 4-color mode on the EGA adapter.

The recommended configuration contains a VGA color monitor and adapter, a mouse as the graphics input device, and a math coprocessor. The keyboard can serve as the graphics input device but is not as user-friendly as the mouse. SAPHIRE 4.0 will run faster if a math coprocessor is present.

The SAPHIRE 4.0 system requires the above specified amount of random access memory (RAM). The memory must be available for the program and not used by memory resident programs, such as *Side-Kick*. The MS/DOS command CHKDSK can be used to display the amount of memory available on your machine. Also, certain memory resident programs may interfere with the execution of SAPHIRE 4.0.

The DOS CONFIG.SYS file must also contain certain minimum configuration parameters for SAPHIRE 4.0 to execute properly. The following parameters must be included in the CONFIG.SYS file, if not already present.

- FILES=30

- BUFFERS=15

- DEVICE=ANSI.SYS

For the SAPHIRE 4.0 test procedures, the following hardware configurations are recommended:

- IBM-AT 386/33MHZ or 100% compatible (or better) with 4M of RAM

- DOS 5.0 or later

- 200 Mb hard disk (or larger)

- 387 Math coprocessor (or better)

- Color Video Graphics Array (VGA) Monitor (or better)

- Graphics input devices — keyboard or mouse

F-3. SOFTWARE REQUIREMENTS

The following software testing uses the July 21, 1992 release of the SAPHIRE 4.0 code suite.

F-4. PROCEDURE 1—IRRAS NONVITAL FAULT TREE FEATURES

F-4.1 Introduction

IRRAS includes functions that allow the user to create fault trees and event trees, define accident sequences and basic event failure data, solve system and accident sequence fault trees and event trees, quantify cut sets, and perform uncertainty analysis on obtained results. IRRAS also allows the analyst to generate reports and displays to document the results of an analysis.

This procedure tests the nonvital features associated with the fault tree analysis functions included in IRRAS. These nonvital fault tree analysis features include fault tree construction, plot trees, load graphic trees, alpha to graphics, graphical pager, and extract fault trees. Table F-2 lists the features, success criteria and the options that are tested in each of the sections. The steps in each section of this procedure should be followed. If any step is not completed successfully, an anomaly report should be generated and submitted to the SAPHIRE 4.0 V&V committee.

Table F-2. IRRAS fault tree test procedure specifications.

Sections	Feature group	Features	Success criteria	Options tested
1	Modify Database	Modify family	To add, modify and delete the family name	Add, Modify Text, & Delete
	Select Family	Select Family	To change the family and verify the selected family	Select Family

Table F-2. IRRAS fault tree test procedure specifications.

Sections	Feature group	Features	Success criteria	Options tested
2	Build Graphic Trees	Bild (Build)	To construct a fault tree.	Line, And, Or, Bevent, Table, N/N OR, Uevent, Transin, House, Inhibit, Not And, Not OR BBevent, Rtrans, LTrans, UdTrans Horbox, & Verbox
		File	To load and perform file manipulations	Laser, New, Save, List, Load
		Edit	To modify fault tree diagrams	Move, Copy, Duplicate, Delete,
		Text	To create titles, labels, descriptions, and names for the fault tree	Show Name, Rep Name, Tble Entry, Edit Table Write Text, Edit Text
	Plot Trees	HPGL File Raster File	Fault trees should be identical in content and structure	HPGL File Raster File
3	Build Graphic Trees	File	To load and perform file manipulations	List
		Text	To create titles, labels, descriptions, and names for the fault tree	Font, Write Text, Show Name, Find Name, Clear Find, Edit Table, Sho Cevent, Rep Cevent,
		Edit	To modify fault tree diagrams	Copy Text, Move Text, Edit Text, Global Text,
4	Build Graphic Trees	Edit	To modify fault tree diagrams	Attributes, Fill Color, Line Color, Line Type, Text Color, Text Size, Text Just, Text Font, Name Font, Name Font, Name Size, Shape Type,
		Views	To change the position and size of the displayed diagram	Zin, Zres, Page ↑, Page ↓, Page ←, Page →, Scroll, Zout Show Name, No Name, Show Text, No Text

Table F-2. IRRAS fault tree test procedure specifications.

Sections	Feature group	Features	Success criteria	Options tested
		Show	To clear the screen and redisplay the defined diagram	Show
5	Build Graphic Trees	File	To load and save drawings and perform file manipulations	Tran →, ←Tran FName, Merge Logic
6	Load Graphic Trees	Load	To load Graphic Trees	Load
	Alpha to Graphics	Convert	To convert the alphanumeric logic for a system or subsystem to a graphical format	Convert
7	Extract Fault Trees	Display	To display all the extracted trees	Display
		Clear	Clear all extracted trees from a file	Clear
		Extract	Remove specific trees from the data base	Extract
	Graphical Pager	Page	To break up larger drawing into several pages	Show, Find, No Tab, Table, Boxed,No Box, Page

F-4.2 Section 1

1. Access the PRADATA.B1 directory. Type **IRRAS** and press <enter> to access the *IRRAS* menu. Place the cursor on the "MODIFY Database" option and press <enter> to access the *Modify Database* menu.

2. From the *Modify Database* menu, place the cursor on the "Family" option and press <enter> to access the *Edit Family* screen. Type A in order to add a new family and press <enter> to access the *Add Family* screen.

3. Type **TEST** in the space for Name. Now move the cursor in the Description space. Type "This is a test" and press <enter> twice to return to the *Edit Family* screen. Verify from the *Edit Family* screen that a new family name and description have been added to the database.

4. Repeat step 3 but type FLTREE in the space for name and type **IRRAS fault tree test procedure specification** in the space for Description.

5. Exit the *Edit Family* screen and return to the *IRRAS* menu. Place the cursor on the "SELECT Family" option and press <enter> to access the *Select Family* screen.

6. Place the cursor on the family name called FLTREE and press <enter>. Note: *By invoking this option you have returned to the IRRAS menu.* Verify that the family name was selected by observing the name displayed on the top of the *IRRAS* menu.

7. Press <enter> to access the *Select Family* screen again. Place the cursor on the family name called TEST and press <enter>. Verify that the family name was successfully selected by observing the name displayed on the top of the *IRRAS* menu.

8. From the *IRRAS* menu, place the cursor on the "MODIFY Database" option and press <enter> to access the *Modify Database* menu. Place the cursor on the "Family" option and press <enter> to access the *Edit Family* screen.

9. Type **M** and place the cursor on the family name called Test. Press <enter> to access the *Modify Family* screen.

10. Place the cursor in the space for Description and type **new information**. Press <enter> to return to the *Edit Family* screen. Verify that the descriptive information was modified.

11. From the *Edit Family* screen type **T** for text, and place the cursor on the family name called Test, and press <enter> to access the *Edit Text* screen. Note: *The screen should be blank.*

12. Press **<Alt-A>** and type **text of your choice**. Verify that the typed text is displayed on the screen.

13. Press **<Alt-B>**. Verify that a line was added before the current line. Type **text of your choice**.

14. Press the **<Alt-D>** keys. Verify that the line of text was deleted. Then press **<Alt-R>**. Verify that the line of text is restored on the screen.

15. Place the cursor at the beginning of the text that was typed in Step 12. Press **<Ctrl-End>**. Verify that the text is deleted from the screen.

16. Press the **<ESC>** key to return to the *Edit Family* screen. The message "**Text record not modified**" will be displayed. Press <enter> again to return to the *Edit Text* screen. Verify that the screen is blank.

17. Press **<Alt-A>** and type **text of your choice**. Verify that the typed text is displayed on the screen.

18. Delete a character by pressing the **<Del>** key. Verify that the character was deleted. Insert a space by pressing the **<Ins>** key. Verify that a space was created.

19. Press **<Ctrl-Z>** to return to the *Edit Family* screen. The message "**Text record modified**" will be displayed on the bottom of the screen. Press <enter> again to access the *Edit Text* screen. Verify that the revisions made were saved. Exit and return to the *Edit Family* screen. Then exit the *Edit Family* screen and return to the *IRRAS* menu.

20. Place the cursor on the "SELECT Family" option and press <enter> to access the *Select Family* screen. Place the cursor on the family name called FLTREE and press <enter> to return to the *IRRAS* menu. The family name will be displayed in the top left corner of the *IRRAS* menu.

21. From the *IRRAS* menu place the cursor on the "MODIFY Database" option and press <enter> to access the *Modify Database* menu. Place the cursor on the "Family" option and press <enter> to access the *Edit Family* screen.

22. Type **D** and place the cursor on the family name called Test. Press <enter> to access the *Delete Family* screen. Press <enter> again. A warning message will be displayed on the screen. Type **Y** and press <enter>. The message "**Deletion completed**" will be displayed at the bottom of the screen. Verify from the *Edit Family* screen that the family name called Test was deleted. Then exit from the *Edit Family* screen and return to the *IRRAS* menu

23. IF: Steps 1 through 22 were performed successfully,

THEN: Go to Step 24.

ELSE: Record which step(s) was unsuccessful. Generate an anomaly report and submit it to the SAPHIRE 4.0 V&V committee.

24. End of Section 1.

F-4.3 Section 2

1. From the *IRRAS* menu, place the cursor on the "Build Fault Trees" option and press <enter> to access the *Fault Tree Graphics System* menu. Place the cursor on the "Build Graphic Trees" option and press <enter> to access the *Primary Fault Tree Graphics* menu.

2. Place the cursor on the "BILD" option and press the **left mouse button** to access the *BUILD* menu.

3. Place the cursor on the "Line" option and press the **left mouse button**. The message "**Enter points for line**" will be displayed on the bottom left corner on the screen. Place the cursor at a desired location and press the **left mouse button**. Drag the cursor in the direction where you want the line. Press the **left mouse button** again to enter the second point of the line. Press the **right mouse button** to terminate the line. Verify that a line appears on the screen at the desired location. Press the **right mouse button** again to terminate the selected option

4. Place the cursor on the "And" option on the *BUILD* menu and press the **left mouse button**. Drag the symbol to the desired location and press the **left mouse button** again to position the symbol on the screen. Press the **right mouse button** to terminate the selected option.

5. Repeat Step 4 for the remaining options on the *BUILD* menu. Verify that each symbol placed on the screen is the same symbol that was selected from the *BUILD* menu.

6. Return to the *Primary Fault Tree Graphics* menu by placing the cursor on any option and press the **right mouse button**. Place the cursor on the "FILE" option and press the **left mouse button** to access the *FILE* menu.

7. Place the cursor on the "Laser" option and press the **left mouse button** to obtain a printout of the symbols. Verify that the symbols on the printout match the symbols displayed on the screen.

8. From the *Primary Fault Tree Graphics* menu, place the cursor on the "EDIT" option and press the **left mouse button** to access the *EDIT* menu.

9. Place the cursor on the "Move" option and press the **left mouse button**. The message "**Box region to be moved- press CANCEL to quit**" will be displayed in the bottom left corner of the screen.

10. Place the cursor near a symbol to be moved and press the **left mouse button**. Drag the cursor until the symbol is located inside the box and press the **left mouse button** again. The message "**Pick reference point - press CANCEL to reselect.**" will be displayed on the bottom left corner of the screen.

11. Place the cursor on the symbol and press the **left mouse button**. The message "**Position box at new location - press CANCEL to reselect.**" will be displayed on the bottom left corner of the screen. Place the box and the cursor at a new desired location and press the **left mouse button** again. Verify that the symbol has moved to the new location. Press the **right mouse button** twice to terminate the selected option.

12. From the *EDIT* menu, place the cursor on the "Copy" option and press the **left mouse button**. The message "**Box region to be copied - press CANCEL to quit.**" will be displayed in the bottom left corner on the screen.

13. Place the cursor near the symbol to be copied and press the **left mouse button**. Drag the cursor until the symbol is located inside the box and press the **left mouse button** again. The message **"Pick reference point - press CANCEL to reselect."** will be displayed in the bottom left corner of the screen.

14. Place the cursor on the symbol and press the **left mouse button**. The message **"Position box at new location - press CANCEL to reselect."** will be displayed in the bottom left corner of the screen. Place the box and the cursor at a new location and press the **left mouse button** again. Verify that the symbol was copied to a new location. Press the **right mouse button** twice to terminate the selected option.

    Note: *The "Copy" option will copy the gate(s) and/or event(s) in the boxed region and new gate and/or event number(s) will be assigned. The "Show Name" option will help verify the "Copy" option.*

15. From the *Primary Fault Tree Graphics* menu, place the cursor on the "TEXT" option and press the **left mouse button** to access the *TEXT* menu.

16. Place the cursor on the "Show Name" option on the *TEXT* menu and press the **left mouse button**. The message **"Pick the shape"** will be displayed on the bottom left corner of the screen.

17. Place the cursor on the copied symbol and press the **left mouse button**. Observe the event/gate name in the bottom left corner on the screen. Place the cursor on the copied symbol and press the **left mouse button** again. Verify that a new event/gate name was assigned.

18. Return to the *EDIT* menu, and place the cursor on the "Duplicate" option, and press the **left mouse button**. The message **"Box region to be duplicated - press CANCEL to quit."** will be displayed on the bottom left corner of the screen.

19. Place the cursor near the symbol to be duplicated and press the **left mouse button**. Drag the cursor until the symbol is located inside the box and press the **left mouse button** again. The message **"Pick reference point - press CANCEL to reselect."** will be displayed in the bottom left corner of the screen.

20. Place the cursor on the symbol and press the **left mouse button**. The message **"Position box at new location - press CANCEL to reselect."** will be displayed in the bottom left corner of the screen. Place the box and the cursor at a new location and press the **left mouse button** again. Verify that the symbol was duplicated at the new location. Press the **right mouse button** twice to terminate the selected option.

    Note: *The "Duplicate" option will duplicate the gate(s) and/or event(s) in the boxed region. The gate and/or event number(s) in the new location will be the same. The "Show Name" option will help verify the "Duplicate" option.*

21. From the *Primary Fault Tree Graphics* menu, place the cursor on the "TEXT" option and press the **left mouse button** to access the *TEXT* menu.

22. Place the cursor on the "Show Name" option on the *TEXT* menu and press the **left mouse button**. The message **"Pick the shape"** will be displayed in the bottom left corner of the screen.

23. Place the cursor on the duplicated symbol and press the **left mouse button**. Observe the event/gate name in the bottom left corner of the screen. Place the cursor on the duplicated symbol and press the **left mouse button** again. Verify that the same event/gate name was assigned. Press the **right mouse button** to terminate the selected option. Press the **right mouse button** again to terminate the *TEXT* menu.

24. From the *EDIT* menu, place the cursor on the "Delete" option and press the **left mouse button**. The message **"Pick region to be deleted"** will be displayed in the bottom left corner of the screen.

25. Place the cursor near the symbol to be deleted and press the **left mouse button**. Drag the cursor until the box surrounds the symbol to be deleted and press the **left mouse button**. The message **"Delete region? Left = delete, Right = cancel"** will be displayed in the bottom left corner of the screen. Press the **left mouse button**. Verify that the selected region on the screen was deleted. Press the **right mouse button** to terminate the selected option. Press the **right mouse button** again to cancel the *EDIT* menu.

26. Return to the *Primary Fault Tree Graphics* menu. Place the cursor on the "FILE" option and press the **left mouse button** to access the *FILE* menu. Place the cursor on the "New" option and press the **left mouse button**. The message **"Current drawing not saved. Enter C to continue>"** will be displayed in the bottom left corner of the screen. Type C and press <enter>. Verify that a new screen appears for a new editing session.

27. With a new editing screen, place the cursor on the "BILD" option and press the **left mouse button** to access the *BUILD* menu on the screen.

28. Using the options found in the *BUILD* menu, create the fault tree that is shown in Figure F-1. When the fault tree is completed, place the cursor on the *BUILD* menu and press the **right mouse button** to cancel the *BUILD* menu.

29. From the *Primary Fault Tree Graphics* menu place the cursor on the "TEXT" option and press the **left mouse button** to access the *TEXT* menu.

Note: *The gate and event names can be added to the fault tree by using the "Rep Name" option.*

30. Place the cursor on the "Rep Name" option and press the **left mouse button**. The message **"Pick shape to be named"** will be displayed in the bottom left corner of the screen.

31. Place the cursor on a symbol and press the **left mouse button**. The message **"Enter new name or CR for=> <"** will be displayed in the bottom left corner of the screen. Type the appropriate name for the selected symbol and press <enter>. To verify the name of the selected symbol, press the **left mouse button**. Press the **right mouse button** to cancel the option.

32. Continue selecting and naming the remaining symbols as shown in Figure F-1. When all the shapes have been named, press the **right mouse button** to terminate the "Rep Name" option.

Note: *Table entries can be added by using the "Table Entry" option from the TEXT menu, this will be used in creating Figure F-2.*

33. Place the cursor on the "Tble Entry" option and press the **left mouse button**. The message **"Pick Table"** will be displayed on the bottom left corner of the screen. Place the cursor on a table and press the **left mouse button**. The message **"Enter name <terminate with CR>"** will be displayed on the bottom left corner of the screen.

34. Type the **first table entry** and press the **left mouse button**. Continue entering the remaining table entries. When finished, press the **left mouse button** again. Then press the **right mouse button** to cancel the selected option.

Note: *The Edit Table option will be used to verify the Tble Entry option.*
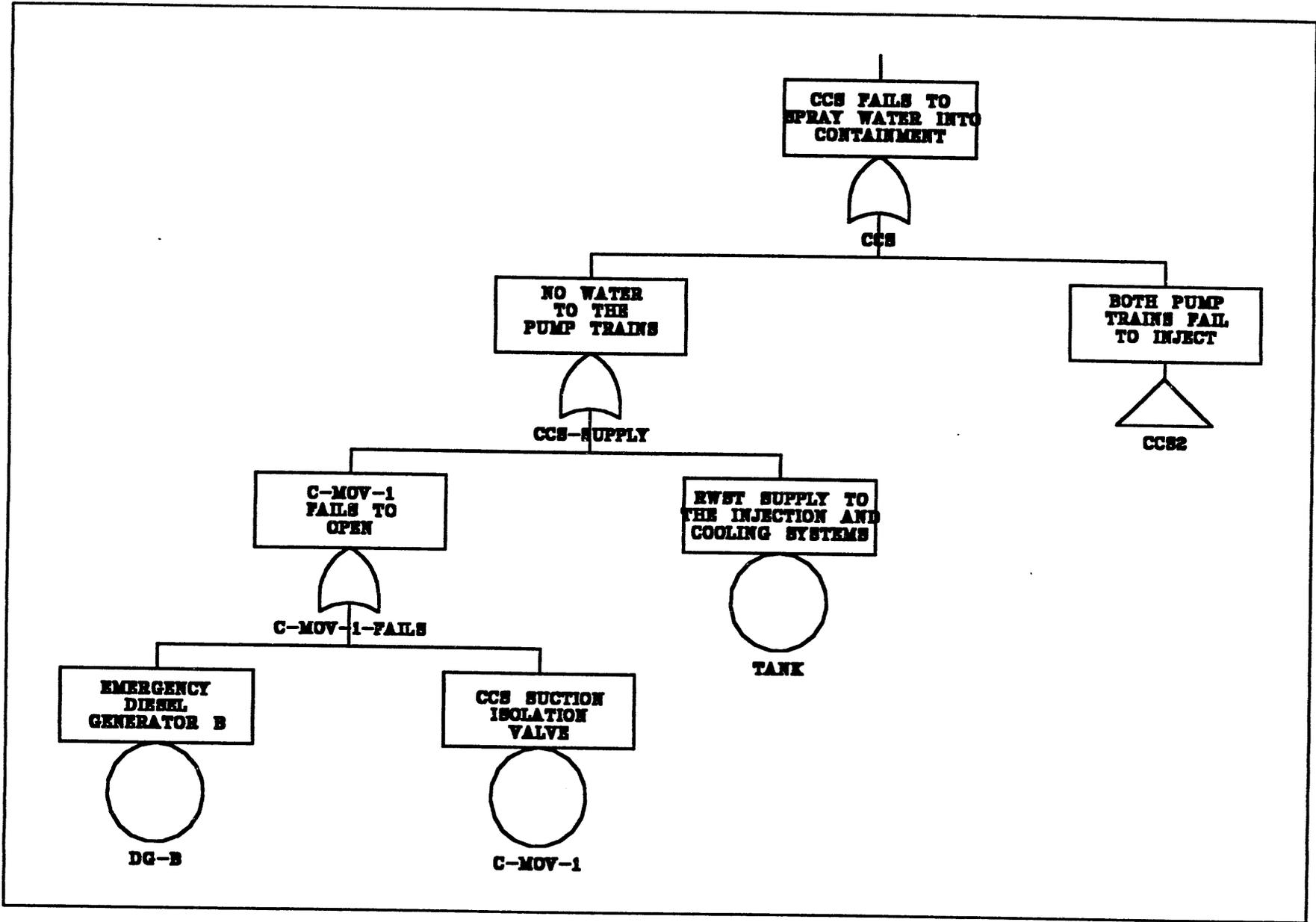
Figure F-1. First containment cooling system fault tree.

35. Place the cursor on the "Edit Table" option and press the **left mouse button**. The message **"Pick gate, event, or tab"** will be displayed on the bottom left corner on the screen. Place the cursor on the Table entry and press the **left mouse button**. Verify that a window is displayed on the screen with the table entries. Press the **right mouse button** twice to cancel the selected option.

    Note: Written text can be added to each event or gate by using the "Write Text" option.

36. From the *TEXT* menu, place the cursor on the "Write Text" option and press the **left mouse button**. The message **"Pick location for text."** will be displayed on the bottom left corner of the screen.

37. Place the cursor on the desired symbol and press the **left mouse button** to access a window. Type the **appropriate text** in accordance with Figure F-1 and then press the **<ESC>** key.

38. Repeat Step 37. for the remaining gates and events. When finished, press the **right mouse button** to terminate the "Write Text" option.

    Note: *The "Edit Text" option will be used to verify the "Write Text" option.*

39. Place the cursor on the "Edit Text" option and press the **left mouse button**. The message **"Box text to be edited"** will be displayed in the bottom left corner of the screen. Place the cursor near a symbol and press the **left mouse button**. Drag the cursor until the box surrounds the desired symbol. Press the **left mouse button** to access the text window. Verify and compare the written text to Figure F-1. Then press the **<ESC>** key to terminate the window. Press the **right mouse button** to terminate the "Edit Text" option. Press the **right mouse button** again to terminate the *TEXT* menu.

40. Return to the *Primary Fault Tree Graphics* menu. Place the cursor on the "File" option and press the **left mouse button** to access the *FILE* menu.

41. Place the cursor on the "Save" option and press the **left mouse button**. The message **"Enter file name>"** will be displayed on the bottom left corner of the screen. Type the file name **CCS1** and press **<enter>** to save the file.

42. Place the cursor on the "List" option and press the **left mouse button**. The message **"Pick the file to load"** will be displayed in the bottom left corner of the screen. Verify that CCSI fault tree has been saved and is in the list of files on the screen. Press the **right mouse button** to terminate the "List" option.

43. From the *FILE* menu, place the cursor on the "NEW" option and press the **left mouse button**. Verify that a new screen appears.

44. With a new editing screen, create the fault tree and text shown in Figure F-2 using the applicable steps in Steps 27 through 40 of this section.

45. From the *File* menu, place the cursor on the "Save" option and press the **left mouse button**. The message **"Enter file name>"** will be displayed on the bottom left corner of the screen. Type the file name **CCS2** and press **<enter>** to save the file.

46. Place the cursor on the "Exit" option and press the **left mouse button** to access the *Fault Tree Graphics System* menu. Select the "Plot Trees" option and press **<enter>** to access the *Plot Graphics* menu.

47. Type **H**, press **<F3>** to mark the newly created fault trees, and press **<enter>** to generate an HPGL file. Then type **R**, press **<F3>**, and press **<enter>** to generate an RAS file.
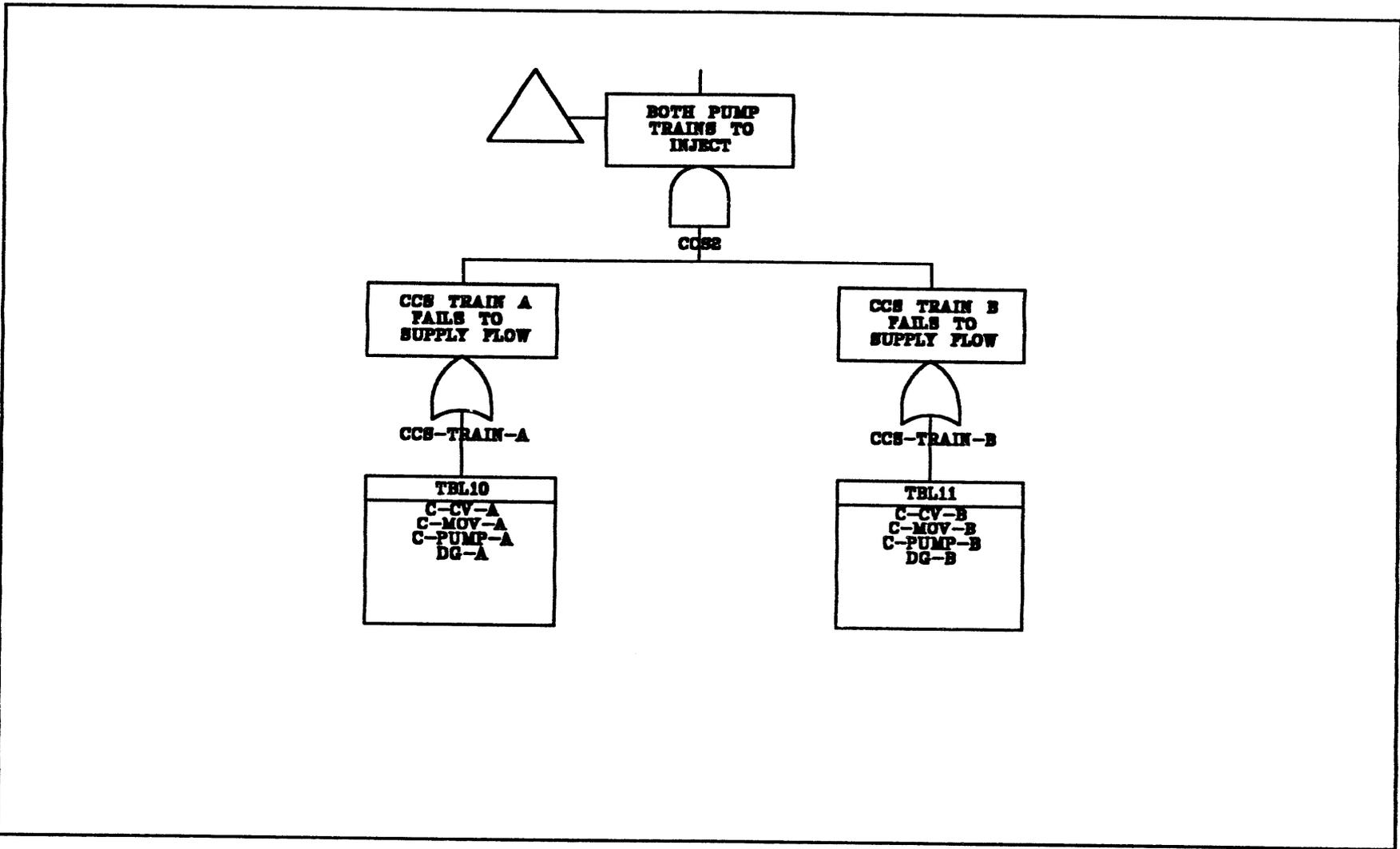
Figure F-2. Second containment cooling system fault tree.

48. After the HPGL and RAS files have been created, exit IRRAS and go to the PRADATA.B1\XXXXX subdirectory. Note: *The subdirectory XXXXX is the family name FLTREE.* Type ..\PLOTHPG *.HPG and press <enter> to obtain a printout of the HPGL files. Note: *PLOTHPG will only work on an HP LaserJet III or IIIP.* When the printout is completed, type ..\PLOTRAS *.RAS and press <enter> to obtain a printout of the RAS files. Note: *PLOTRAS will work on any laser printer.*

49. Verify that the printouts of the fault trees have the correct gates, events, names, and text by comparing with Figures F-1 and F-2.

50. Return to the PRADATA.B1 directory. Type **IRRAS** and press <enter> to access the *IRRAS* menu. Place the cursor on the "Build Fault Trees" option and press <enter> to access the *Fault Tree Graphics System* menu. Then place the cursor on the "Build Graphic Trees" option and press <enter> to access the *Primary Fault Tree Graphics* menu.

51. Place the cursor on the "FILE" option and press the **left mouse button** to access the *File* menu. Place the cursor on the "Load" option and press the **left mouse button**. The message "Enter file name>" will be displayed in the bottom left corner of the screen. Type the file name CCS1 and press <enter>. Verify that the diagram appears on the screen.

52. Place the cursor on the "TEXT" option and press the **left mouse button** to access the *TEXT* menu on the screen.

53. Place the cursor on the "Edit Text" option and press the **left mouse button**. The message "**Box text to be edited.**" will be displayed in the bottom left corner of the screen.

54. Place the cursor near a symbol and press the **left mouse button**. Drag the cursor until the box surrounds the desired symbol. Press the **left mouse button** to access the text window. Modify the written text. Then press the <ESC> key to terminate the window. Press the **right mouse button** to terminate the "Edit Text" option. Press the **right mouse button** again to terminate the *TEXT* menu.

Note: *The "Laser" option from the FILE menu will be used to verify the "Write Text" option.*

55. From the *Primary Fault Tree Graphics* menu, place the cursor on the "FILE" option and press the **left mouse button** to access the *FILE* menu. Place the cursor on the "Laser" option and press the **left mouse button** to obtain a printout. From the printout verify the changes made.

56. Exit and return to the *IRRAS* menu.

57. **IF:**   Steps 1 through 55 were performed successfully,

   THEN: Go to Step 58.

   ELSE: Record which step(s) was unsuccessful. Generate an anomaly report and submit it to the SAPHIRE 4.0 V&V committee.

58. End of Section 2.

F-4.4 Section 3

1. From the *IRRAS* menu, place the cursor on the "Build Fault Trees" option and press <enter> to access the *Fault Tree Graphics System* menu. Place the cursor on the "Build Graphic Trees" option and press <enter> to access the *Primary Fault Tree Graphics* menu.

2. Place the cursor on the "File" option. Press the **left mouse button** to access the *FILE* menu. Then place the cursor on the "List" option and press the **left mouse button**. A list of file names will be

displayed on the screen. Place the cursor on file CCS1 and press the **left mouse button**. Verify that the fault tree is on the screen.

3. Place the cursor on the "TEXT" option and press the **left mouse button** to access the *TEXT* menu. From the *Primary Fault Tree Graphics* menu, place the cursor on the "0.50" option and press the **left mouse button**. The message "**Enter text size>**" will be displayed in the bottom left corner of the screen. Change the text size to 2 and press **<enter>**.

4. Return to the *TEXT* menu. Place the cursor on the "FONT" option and press the **left mouse button** to access the *FONT* menu. Place the cursor on the desired font and press the **left mouse button**. Return to the *TEXT* menu. Place the cursor on the "Write Text" option and press the **left mouse button**. The message "**Pick location for text**" will be displayed in the bottom left corner of the screen.

5. Place the cursor at the location where text has not been written, press the **left mouse button** to access the text window. Type some text and press the **<ESC>** key. Verify that the written text changed to the desired font. Press the **right mouse button** to terminate option.

6. Repeat Steps 4 and 5 for the remaining fonts. Press the **right mouse button** to terminate the *TEXT* menu.

7. From the *Primary Fault Tree Graphics* menu, place the cursor on the "EDIT" option and press the **left mouse button** to access the *EDIT* menu.

8. Place the cursor on the "Delete" option and press the **left mouse button**. The message "**Pick region to be deleted**" will be displayed in the bottom left corner of the screen. Place the cursor near the text that was written in Step 5. Drag the cursor until the box surrounds the text and press the **left mouse button**. The message "**Delete region? Left = delete, Right = cancel**" will be displayed in the bottom left corner of the screen. Press the **left mouse button**. Verify that the text was deleted. Press the **right mouse button** to terminate the "Delete" option. Then press the **right mouse button** to cancel the *EDIT* menu.

9. From the *Primary Fault Tree Graphics* menu, place the cursor on the "TEXT" option and press the **left mouse button** to access the *TEXT* menu.

10. Place the cursor on the "Show Name" option and press the **left mouse button**. The message "**Pick the shape**" will be displayed on the bottom left corner of the screen. Place the cursor on the desired symbol and press the **left mouse button** again. Verify and record the symbol name displayed on the bottom left corner of the screen. Continue to verify the remaining symbol names. Press the **right mouse button** to terminate the "Show Name" option.

11. Place the cursor on the "Find Name" option and press the **left mouse button**. The message "**Enter name>**" will be displayed in the bottom left corner of the screen. Type the **symbol name** recorded in Step 10 and press the **left mouse button**. Verify that a broken dotted line is displayed on the screen and surrounds the specified symbol.

12. Place the cursor on the "Clear Find" option and press the **left mouse button**. Verify that the broken dotted line highlighting the specified symbol disappears.

13. Place the cursor on the "Edit Table" option and press the **left mouse button**. The message "**Pick gate, event, or tab**" will be displayed on the bottom left corner of the screen. Place the cursor on the table symbol and press the **left mouse button**. The message "**Pick the name to modify,- use '@' in 1st col to delete**" will be displayed on the bottom left corner of the screen. Verify the text (basic event name) in the table.

14. Place the cursor on the written text (basic event name) in the window and press the **left mouse button**. Verify the basic event name displayed on the bottom left corner of the screen. Type a **new basic event name** and press the **left mouse button**. Verify that the basic event name was changed on the table. Press the **right mouse button** twice to cancel the "Edit Table" option.

15. Return to the *Primary Fault Tree Graphics* menu. Place the cursor on the "BILD" option to access the *BUILD* menu. Place the cursor on the Inhibit gate symbol and press the **left mouse button**. Place the Inhibit gate symbol on the fault tree and press the **left mouse button**. Press the **right mouse button** to cancel the option. Press the **right mouse button** again to terminate the *BUILD* menu.

16. Return to the *TEXT* menu. Place the cursor on the "Sho Cevent" option and press the **left mouse button**. The message **"Pick the INHIBIT gate"** will be displayed in the bottom left corner of the screen. Place the cursor on the INHIBIT gate. Verify the gate name displayed on the bottom left corner on the screen. Record the gate name and press the **right mouse button** to terminate the "Sho Cevent" option.

17. Place the cursor on the "Rep Cevent" option and press the **left mouse button**. The message **"Pick the INHIBIT gate"** will be displayed in the bottom left corner of the screen. Place the cursor on the INHIBIT gate and press the **left mouse button**. The message **"Enter CEVENT name>"** will be displayed in the bottom left corner of the screen. Type a **new Cevent name** and press the **left mouse button**. Press the **right mouse button** to terminate the "Rep Cevent" option.

18. Place the cursor on the "Sho Cevent" option and press the **left mouse button**. The message **"Pick the INHIBIT gate"** will be displayed in the bottom left corner of the screen. Place the cursor in the INHIBIT gate and press the **left mouse button**. Verify the gate name displayed in the bottom left corner of the screen has changed. Press the **right mouse button** to terminate the "Sho Cevent" option. Press the **right mouse button** again to cancel the *TEXT* menu.

19. Return to the *Primary Fault Tree Graphics* menu. Place the cursor on the "EDIT" option and press the **left mouse button** to access the *EDIT* menu.

20. Place the cursor on the "Copy Text" option and press the **left mouse button**. The message **"Pick text to be copied - press CANCEL to quit."** will be displayed in the bottom left corner of the screen. Place the cursor near the text to be copied and press the **left mouse button** again. Drag the cursor until the box surrounds the text and press the **left mouse button**. The message **"Pick reference point - press CANCEL to reselect."** will be displayed in the bottom left corner of the screen.

21. Select a reference point by placing the cursor on the text and press the **left mouse button**. Place the box at a new location where the text can be copied and press the **left mouse button**. Verify that the text was copied at the new location on the screen. Press the **right mouse button** twice to terminate the "Copy Text" option.

22. Place the cursor on the "Move Text" option and press the **left mouse button**. The message **"Box region to be moved - press CANCEL to quit."** will be displayed in the bottom left corner of the screen. Place the cursor near the text to be moved and press the **left mouse button**. Drag the cursor until the box surrounds the text and press the **left mouse button** again. The message **"Pick reference point - press CANCEL to reselect."** will be displayed in the bottom left corner of the screen.

23. Select a reference point by placing the cursor on the text to be moved and pressing the **left mouse button**. Place the box at a new location where the text can be moved and press the **left mouse button**. Verify that the text was moved to the new location on the screen. Press the **right mouse button** twice to terminate the "Move Text" option.

24. Place the cursor on the "Global Txt" option and press the **left mouse button**. The message **"Enter search string"** will be displayed in the bottom left corner of the screen. Type a **phase of written text**

used in the fault tree in the search string entry and press the **left mouse button**. The message "**Enter replacement string**" will be displayed in the bottom left corner of the screen. Type the **replacement string text** and press the **left mouse button**.

Note: *The Edit txt option will be used to verify the "Global Txt" option.*

25. Place the cursor on the "Edit txt" option and press the **left mouse button**. The message "**Box text to be edited**" will be displayed at the bottom of the screen. Drag the cursor until the box surrounds the text. Press the **left mouse button** to access the text window. Verify that all occurrences with that particular phrase were replaced. Press the <ESC> key and then the **right mouse button** to cancel the option.

26. <u>IF</u>: Steps 1 through 24 were performed successfully

   <u>THEN</u>: Go to Step 26.

   <u>ELSE</u>: Record which step(s) was unsuccessful. Generate an anomaly report and submit it to the SAPHIRE 4.0 V&V committee.

26. End of Section 3.

F-4.5 Section 4

1. Return to the *EDIT* menu. Place the cursor on the "ATTRIBUTES" option and press the **left mouse button** to access the *EDIT ATTR* menu.

2. Place the cursor on the "Fill Col" option and press the **left mouse button**. The message "**Pick a new color from the color bar**" will be displayed in the bottom left corner of the screen. Select a new color by placing the cursor on a color from the color bar located at the bottom of the *Primary Fault Tree Graphics* menu. Press the **left mouse button**. The message "**Pick shapes to be modified**" will be displayed in the bottom left corner of the screen.

3. Place the cursor near a desired symbol and press the **left mouse button**. Drag the cursor until a box surrounds the shape to be modified and press the **left mouse button** again. Verify that the symbol changed to the selected color. Press the **right mouse button** to terminate the "Fill Col" option.

4. Repeat Steps 2 and 3 for the remaining colors.

5. Place the cursor on the "Line Col" option and press the **left mouse button**. The message "**Pick a new color from the color bar**" will be displayed in the bottom left corner of the screen. Place the cursor on the color bar located on the *Primary Fault Tree Graphics* menu and select a new color by pressing the **left mouse button**. The message "**Pick line(s) to be modified**" will be displayed in the bottom left corner of the screen.

6. Place the cursor near a line and press the **left mouse button**. Drag the cursor until a box surrounds the line to be modified and press the **left mouse button** again. Verify that the line changed to the selected color. Press the **right mouse button** to terminate the "Line Col" option.

7. Repeat Steps 5 and 6 for the remaining colors.

8. Place the cursor on the "Line Type" option and press the **left mouse button**. A small window will display the line options. Also, the message "**Pick line type**" will be displayed in the bottom left corner of the screen. Place the cursor on the dashed line option and press the **left mouse button**. The message "**Pick line(s) to be modified**" will be displayed in the bottom left corner of the screen.

9.   Place the cursor near the line to be modified and press the **left mouse button**. Drag the cursor until a box surrounds the line and press the **left mouse button** again. Verify that the line on the screen changed to a dashed line. Press the **right mouse button** to terminate the "Line Type" option.

10. Repeat Steps 8 and 9 for the dotted and solid line options.

11. Place the cursor on the "Text Col" option and press the **left mouse button**. The message **"Pick a new color from the color bar"** will be displayed in the bottom left corner of the screen. Place the cursor on a new color from the color bar and press the **left mouse button**. The message **"Pick text to be modified"** will be displayed in the bottom left corner of the screen.

12. Place the cursor near some text on the screen and press the **left mouse button**. Drag the cursor until a box surrounds the text to be modified and press the **left mouse button** again. Verify that the text changed to the selected color. Press the **right mouse button** to terminate "Line Col" option.

13. Repeat Steps 11 and 12 for the remaining colors.

14. Place the cursor on the "Text Size" option and press the **left mouse button**. The message **"Enter text size option >"** will be displayed in the bottom left corner of the screen. Type a new text size of **0.70** and press the **left mouse button**. The message **"Pick text to be modified"** will be displayed in the bottom left corner of the screen.

15. Place the cursor near the text to be modified and press the **left mouse button**. Drag the cursor until a box surrounds the text and press the **left mouse button** again. Verify that the text changed to the selected size. Press the **right mouse button** to terminate the "Text Size" option.

16. Repeat Steps 14 and 15 but change the text size to **0.80, 1.0,** and **1.5**.

17. Place the cursor on the "cntr" (center) option located on the *Primary Fault Tree Graphics* menu and press the **left mouse button**. This will toggle the justification from "cntr" to "rght" (right) justification.

18. Return to the *Edit Attr* menu. Place the cursor on the "Text Just" option and press the **left mouse button**. The message **"Pick text to be modified"** will be displayed in the bottom left corner of the screen. Place the cursor near the text to be modified and press the **left mouse button**. Drag the cursor until the box surrounds the desired text and press the **left mouse button** again. Verify that the text changed to the selected justification. Press the **right mouse button** to terminate the "Text Just" option.

19. Repeat Steps 17 and 18 but change the text justification from "rght" to "left" and then back to "cntr."

20. Return to the *Primary Fault Tree Graphics* menu. Place the cursor on the "VIEW" option and press the **left mouse button** to access the *VIEWS* menu. Place the cursor on the "Zin" (zoom in) option and press the **left mouse button**. The message **"Pick two corners of box"** will be displayed in the bottom left corner of the screen.

21. Place the cursor near one corner of an area with text to zoom in on and press the **left mouse button**. Drag the cursor until a box surrounds the desired area and then press the **left mouse button** again. Verify that the selected area on the screen was enlarged (zoomed in).

22. Return to the *Primary Fault Tree Graphics* menu. Place the cursor on the "EDIT" option and press the **left mouse button** to access the *EDIT* menu. Place the cursor on the "ATTRIBUTES" option and press the **left mouse button** to access *EDIT ATTR* menu.

23. Place the cursor on the "Text Font" option and press the **left mouse button** to access the *FONT* menu. Place the cursor on the desired font and press the **left mouse button**. The message **"Pick text to be modified"** will be displayed in the bottom left corner of the screen.

24. Place the cursor near the text that is to be modified and press the **left mouse button**. Drag the cursor until a box surrounds the text and press the **left mouse button** again. Verify that the text changed to the selected font. Press the **right mouse button** to terminate the "Text Font" option.

25. Repeat Steps 23 and 24 for several fonts but return to the original font.

26. Place the cursor on the "Name Font" option and press the **left mouse button** to access the *FONT* menu. Place the cursor on the desired font and press the **left mouse button**. The message **"Pick shapes to be modified"** will be displayed in the bottom left corner of the screen.

27. Place the cursor near the symbol that is to be modified and press the **left mouse button**. Drag the cursor until the box surrounds the shape to be modified and then press the **left mouse button** again. Verify that the symbol name changed to the selected font. Press the **right mouse button** to terminate the "Name Font" option.

28. Repeat Steps 26 and 27 for several Fonts but return to the original font.

29. Return to the *VIEWS* menu and place cursor on the "Zres" (zoom restore) option and press the **left mouse button**. Verify that the region and text returned to the original size. Press the **right mouse button** to terminate the *VIEWS* menu.

30. Return to the *Primary Fault Tree Graphics* menu. Place the cursor on the "EDIT" option and press the **left mouse button** to access the *EDIT* menu. Place the cursor on the "ATTRIBUTES" option and press the **left mouse button** to access *EDIT ATTR* menu.

31. Place the cursor on the "Name Col" option and press the **left mouse button**. The message **"Pick a new color from the color bar"** will be displayed in the bottom left corner of the screen. Place the cursor on the color bar located at the *Primary Fault Tree Graphics* menu. Press the **left mouse button** to select a new color. The message **"Pick shapes to be modified will be displayed"** in the bottom left corner of the screen.

32. Place the cursor near the symbol that is to be modified and press the **left mouse button**. Drag the cursor until the box surrounds the shape to be modified and press the **left mouse button** again. Verify that the symbol name changed to the selected color. Press the **right mouse button** to terminate the "Name Col" option.

33. Repeat Steps 31 and 32 for the remaining colors.

34. Place the cursor on the "Name Size" option and press the **left mouse button**. The message **"Enter text size"** will be displayed in the bottom left corner of the screen. Type **0.70** for the text size and press the **left mouse button**. The message **"Pick shapes to be modified"** will be displayed in the bottom left corner of the screen.

35. Place the cursor near the symbol that is to be modified and press the **left mouse button**. Drag the cursor until the box surrounds the desired symbol and press the **left mouse button** again. Verify that the symbol name changed to the selected size. Press the **right mouse button** to terminate the "Name Size" option.

36. Repeat Steps 34 and 35 for a text size of 0.60 and 1.0.

37. Place the cursor on the "Shape Type" option and press the **left mouse button** to access the *Symbol* menu. The message **"Select the new shape type, or <cancel> to quit"** will be displayed in the bottom left corner of the screen.

38. Place the cursor on the "And" option and press the **left mouse button**. The message "Select shapes to be changed to new type will be displayed in the bottom left corner of the screen.

39. Place the cursor near the symbol that is to be modified and press the **left mouse button**. Drag the cursor until the box surrounds the symbol to be modified and press the **left mouse button** again. Verify that the symbol changed to the selected shape on the screen. Press the **right mouse button** to terminate the "Symbol" option.

40. Return to the *Symbol* menu and repeat Steps 38 and 39 for the remaining symbols. Return to the original symbol when completed. Press the **right mouse button** to terminate the "Symbol" option.

41. Return to the *Primary Fault Tree Graphics* menu. Place the cursor on the "SHOW" option and press the **left mouse button**. Verify that all of the menus are cleared from the screen.

42. Return to the *Primary Fault Tree Graphics* menu. Place the cursor on the "VIEWS" option to access the *VIEWS* menu. Place the cursor on the "Page ↑" option and press the **left mouse button** to move the diagram up a page. The message **"View changed"** will be displayed in the bottom left corner of the screen. Verify that the screen is blank.

43. Place the cursor on the "Page ↓" option and press the **left mouse button** to move the diagram down a page. Verify that the fault tree returns to the screen.

44. Place the cursor on the "Page ←" option and press the **left mouse button** to move the diagram to the left one page. Verify that the screen is blank.

45. Place the cursor on the "Page →" option and press the **left mouse button** to move the diagram to the right one page. Verify that the fault tree returns to the screen.

46. Place the cursor on the "Scroll" option and press the **left mouse button**. Verify that a white outline box appears and a cursor is in the center of the box. Place the cursor on the desired location and press the **left mouse button**. Verify that the figure on the screen moves relative to the cursor position.

47. Place the cursor on the "Zout" option and press the **left mouse button**. The message **"View changed."** will be displayed in the bottom left corner of the screen. Verify that the figure is reduced approximately 50%. Place the cursor on the "Zres" option and press the **left mouse button**. Verify that the display returns to original display size.

Note: *The "SHOW" and "Grid" options will be used to test the "Sh Name", "No Name" and "SH Text" options.*

48. Place the cursor on the "Sh Name" option and press the **left mouse button**. Verify that the "Sh Name" option changes to the "No Name" option.

49. Place the cursor on the "Sh Text" option and press the **left mouse button**. Verify that the option changes to the "No Text" option.

50. Return to the *Primary Fault Tree Graphics* menu. Place the cursor on the "SHOW" option and press the **left mouse button**. Verify that all the text and symbol names are cleared from the screen.

51. Return to the *Primary Fault Tree Graphics* menu. Place the cursor on the "VIEW" option and press the **left mouse button** to access the *VIEWS* menu. Place the cursor on the "No Name" option and press the **left mouse button**. Verify that it changes to the "Sh Name" option.

52. Place the cursor on the "No Text" option and press the **left mouse button**. Verify that it changes to "Sh Text" option.

53. Place the cursor on the "Grid" option and press the **left mouse button**. Verify that a grid appears on the screen. Verify that all text and symbol names appear on the screen.

54. Place the cursor on the "Grid" option again and press the **left mouse button** to obtain the original screen background. Verify that the original screen background with symbols and text appears. Place the cursor on the *VIEWS* menu and press the **right mouse button** to terminate the *VIEWS* menu.

55. **IF:**   Steps 1 and 54 were performed successfully,

   THEN: GO TO Step 56.

   ELSE: Record which step(s) was unsuccessful. Generate an anomaly report and submit it to the SAPHIRE 4.0 V&V committee.

56. End of Section 4.

F-4.6 Section 5

1. Return to the *Primary Fault Tree Graphics* menu. Place the cursor on the "FILE" option and press the **left mouse button** to access the *File* menu.

2. Place the cursor on the "NEW" option and press the **left mouse button**. The message **"Current drawing not saved. Enter C to continue>"** will be displayed in the bottom left corner of the screen. Type C to obtain a new screen. Verify that the drawing and *File* menu disappears.

3. Return to the *Primary Fault Tree Graphics* menu. Place the cursor on the "FILE" option and press the **left mouse button** to access the *File* menu.

4. Place the cursor on the "Load" option and press the left mouse button. The message **"Enter file name>"** will be displayed in the bottom left corner of the screen. Type the file name **CCS1** and press **<enter>**. Verify that the fault tree is displayed on the screen.

5. Return to the *Primary Fault Tree Graphics* menu. Place the cursor on the "FILE" option and press the **left mouse button** to access the *File* menu.

6. Place the cursor on the "Tran →" option and press the **left mouse button**. The message **"Pick TRANSIN for transfer"** will be displayed in the bottom left corner of the screen. Place the cursor on the transin symbol and press the **left mouse button**. Verify on the screen that the transfers to the new fault tree were successful.

7. Return to the *Primary Fault Tree Graphics* menu. Place the cursor on the "FILE" option and press the **left mouse button** to access the *File* menu.

8. Place the cursor on the "←Tran" option and press the **left mouse button**. A list of files will be displayed on the screen. Also the message **"Pick the file to transfer to"** will be displayed in the bottom left corner of the screen. Place the cursor on the file named CCS1 and press the **left mouse button**. Ve.ify that the file transferred to the CCS1 fault tree.

9. Return to the *Primary Fault Tree Graphics* menu. Place the cursor on the "FILE" option and press the **left mouse button** to access the *FILE* menu.

10. Place the cursor on the "FName" option and press the **left mouse button**. Verify that the file name is displayed in the bottom left corner of the screen.

11. Place the cursor on the "Merge" option and press the **left mouse button**. A list of files will be displayed on the screen. Also, the message **"Pick the file to load"** will be displayed in the bottom left corner of the screen. Place the cursor on the CCS1 file and press the **left mouse button**. The message **"Pick the location for top center of merged file."** will be displayed in the bottom left corner of the screen. Find a location for the top center of the merged file and press the **left mouse button** again.

    Note: *The "Zout" option from the VIEW menu will be used to test the "Merge" option.*

12. Then return to the *Primary Fault Tree Graphics* menu. Place the cursor on the "VIEW" option and press the **left mouse button** to access the *VIEW* menu.

13. Place the cursor on the "Zout" option and press the **left mouse button**. Verify that the selected fault tree file merged to the current fault tree file. Place the cursor on the "Zres" option and press the **left mouse button** to return the fault tree to its original size. Press the **right mouse button** to terminate the *VIEW* menu.

14. Place the cursor on the "List" option from the file menu and press the **left mouse button**. A list of files will be displayed on the screen. Place the cursor on the file named CCS1 and press the **left mouse button**. The message **"Current drawing not saved. Type G to get anyway>"** will be displayed in the bottom left corner of the screen. Type **G** and press **<enter>**. The fault tree will be displayed on the screen.

    Note: *The Logic Editor option from the Fault Tree Analysis menu will be used to verify the "Logic" option from the FILE menu.*

15. Return to the *Primary Fault Tree Graphics* menu. Place the cursor on the "FILE" option. Press the **left mouse button** to access the *FILE* menu.

16. Then place the cursor on the "Logic" option and press the **left mouse button**. The message **"Enter file name or CR for file CCS1 >"** will be displayed in the bottom left corner of the screen. Type the file name CCS1 and press the **left mouse button**. The message **"Logic saved"** will be displayed in the bottom left corner of the screen.

17. Place the cursor on the "EXIT" option and return to the *IRRAS* menu. Place the cursor on the "MODIFY Database" option and press **<enter>** to access the *Modify Database* menu. Then place the cursor on the "Systems" option and press **<enter>** to access the *Edit System* menu.

18. Type **A** and press **<enter>** to access the *Add System* screen. Type the fault tree file **CCS1** in the space for Name. Move the cursor to the space for Descriptions. Then type **containment cooling system** and press **<enter>** to return to the "Edit Systems screen. Verify that the Name and Descriptions were added.

19. Then exit the *Edit Systems* screen and return to the *Modify Database* menu. Exit the *Modify Database* menu and return to the *IRRAS* main menu.

20. Place the cursor on the "FAULT Tree Analysis" option and press **<enter>** to access the *Fault Tree Analysis* menu.

21. Place the cursor on the "Logic Editor" option and press **<enter>** to access the System Logic Editor screen. Type **L** and place the cursor on file name CCS1 and press **<enter>** to access the *Logic Editor*

screen. Compare with Figure F-1 and verify the gate names, gate types, and inputs to the gates. Return to the *IRRAS* main menu.

22. IF: Steps 1 through 21 were performed successfully

     THEN: Go to Step 23.

     ELSE: Record which step(s) was unsuccessful. Generate an anomaly report and submit it to the SAPHIRE 4.0 V&V committee.

23. End of Section 5.

F-4.7 Section 6

1. Exit from the IRRAS program and change the current directory to the PRADATA.B1\XXXXX subdirectory. Note: *The PRADATA.B1\XXXXX subdirectory is the family name.* Copy the file name ECS.DLS from drive A: to the C:\PRADATA.B1\XXXXX subdirectory. Then change the current directory to the PRADATA.B1 directory and type **IRRAS**.

2. Place the cursor on the "BUILD Fault Trees" option and press <enter> to access the *Fault Tree Graphics System* menu. Place the cursor on the "Logic Graphic Trees" option and press <enter> to access the *Load Fault Tree Graphics* screen.

3. Type **L** for load and place the cursor on the file name ECS, and press <enter>. The message **"Fault tree successfully loaded"** will be displayed at the bottom of the screen.

4. Exit the *Load Fault Tree Graphics* screen and return to the *Faul. Tree Graphics System* menu. Place the cursor on the "Build Graphics Trees" option and press <enter>. From the *Primary Fault Tree Graphics* menu, place the cursor on the "FILE" option and press the **left mouse button** to access the *FILE* menu.

5. Place the cursor on the "LIST" option and press the **left mouse button**. Verify that the file name ECS has been added to the list of file names. Place the cursor on the file and press the **left mouse button**. Verify that the fault tree appears on the screen.

6. Place the cursor on the "EXIT" option and press the **left mouse button** to return to the *Fault Tree Graphics System* menu. Exit and return to the *IRRAS* menu. Place the cursor on the "MODIFY Database" option and press <enter> to access the *Modify Database* menu. Then place the cursor on the "Systems" option and press <enter> to access the *Edit System* screen.

7. Type **A** for add and press <enter> to access the *Add System* screen. Type the file name ECS in the space for Name and press <enter>. The message **"Record added"** will be displayed on the bottom of the screen. Verify that the file name ECS was added to the *Edit Systems* screen. Exit the *Edit Systems* screen and return to the *Modify Database* menu. Then exit the *Modify Database* menu and return to the *IRRAS* menu.

8. Place the cursor on the "FAULT Tree Analysis" option and press <enter> to access the *Fault Tree Analysis* menu. Place the cursor on the "Logic Editor" option and press <enter>. Type **L** and place the cursor on the fault tree name ECS, and press <enter> to access the *Logic Editor* screen. Verify the logic by verifying the gate names, gate types, and gate inputs. Return to the *IRRAS* menu when finished.

9. Place the cursor on the "BUILD Fault Trees" option and press <enter> to access the *Fault Tree Graphic System* menu. Then place the cursor on the "Build Graphic Trees" option and press <enter>.

10. Place the cursor on the "FILE" option located at the *Primary Fault Tree Graphics* menu and press the **left mouse button** to access the *FILE* menu. Place the cursor on the "LIST" option and press the **left mouse button** to access the list of file names. Place the cursor on file name ECS and press the **left mouse button** to access the fault tree.

11. Place the cursor on the "VIEW" option and press the **left mouse button** to access the *VIEWS* menu. Then place the cursor on the "Zin" option and press the **left mouse button**. The message **"Pick two corners of box"** will be displayed in the bottom left corner of the screen. Place the cursor near a basic event symbol and press the **left mouse button**. Drag the cursor until the box surrounds the basic event and press the **left mouse button** again. Record the written text and basic event name.

12. Place the cursor on the "EXIT" option and return to the *IRRAS* menu. Place the cursor on the "MODIFY Data Base" option and press <enter> to access the "Modify Data Base" option. Place the cursor on the "Basic Events" option and press <enter> to access the *Edit Events* screen.

13. Type **M** and place the cursor on the basic event name and text that was recorded in Step 11. Press <enter> to access the *Modify Event* screen. Revise the basic event names and descriptions and press <enter> to return to the *Edit Events* screen. Verify and record these changes. Then exit from the *Edit Events* screen and return to the *IRRAS* menu.

14. Place the cursor on the "Build Fault Trees" option and press <enter> to access the *Fault Tree Graphics System* menu. Then place the cursor on the "Alpha to Graphics Trees" option and press <enter> to access the *Alpha to Graphics* screen.

15. Place the cursor on the name ECS and press <enter>. A window will be displayed with a set of questions. Assign "N" for (Use Tables) and "Y" for (Use Boxed Events) and press <enter>. Exit out of the *Alpha to Graphics* screen and return to the *Fault Tree Graphics System* menu.

16. Place the cursor on the "Build Graphics Trees" option and press <enter>. From *Primary Fault Tree Graphics*, place the cursor on the "FILE" option and press the **left mouse button** to access the *FILE* menu.

17. Place the cursor on the "LIST" option and press the **left mouse button** to access the list of file names. Place the cursor on the file name ECS and press <enter> to access the fault tree.

18. From the *Primary Fault Tree Graphics* menu place the cursor on the "VIEW" option and press the **left mouse button** to access the *VIEWS* menu. Place the cursor on the "Zin" option and press the **left mouse button**.

19. Place the cursor near the basic event that was recorded in Step 11 and press the **left mouse button**. Drag the cursor until the box surrounds the basic event and press the **left mouse button** again. Verify that the revisions to the text and basic event name were made. Place the cursor on the "EXIT" option and return to the *Fault Tree Graphics System* menu.

20. IF: Steps 1 through 19 were performed successfully

    THEN: Go to Step 21.

    ELSE: Record which step(s) was unsuccessful. Generate an anomaly report and submit it to the SAPHIRE 4.0 V&V committee.

21. End of Section 6.

F-4.8 Section 7

1. Return to the *Fault Tree Graphics System* menu. Place the cursor on the "Build Graphic Trees" option and press <enter> to access the *Primary Fault Tree Graphics* menu. Place the cursor on the "FILE" option and press the **left mouse button** to access the *FILE* menu. Place the cursor on the "LIST" option and press the **left mouse button**. Record the list of file names that appear on the screen. Then press the **right mouse button** to cancel the option.

2. Place the cursor on the "EXIT" option and press the **left mouse button** to return to the *Fault Tree Graphics System* menu. Place the cursor on the "eXtract Fault Trees" option and press <enter> to access the *Extract Graphic Fault Trees* screen.

3. Type **D** and press <enter>. Verify that a different *Extract Graphic Fault Tree* screen appears that contains only the file names. Press <enter> to return to the previous *Extract Graphic Fault Trees* screen.

4. Type **C** to clear extracted trees and press <enter>. The message **"WARNING All existing "*.DLS" files will be deleted!! Do you wish to proceed? Y/N"** will be displayed on the screen. Type **Y** and press <enter>. Observe the message at the bottom of the screen.

5. Exit from the *Extract Graphics Fault Trees* screen and return to the *Fault Tree Graphics System* menu. Place the cursor on the "Build Graphic Trees" option and press <enter> to access the *Primary Fault Tree Graphics* menu. Place the cursor on the "FILE" option and press the **left mouse button** to access the *FILE* menu.

6. Place the cursor on the "LIST" option and press the **left mouse button**. Verify that no list of file names appears on the screen.

7. Place the cursor on the "EXIT" option and press the **left mouse button** to return to the *Fault Tree Graphics System* menu. Place the cursor on the "eXtract Fault Trees" option and press <enter> to access the *Extract Graphic Fault Tree* screen.

8. Type **X** and press <enter> to extract trees. The message **"Process all entries?"** will be displayed at the bottom of the screen. Type **Y**.

9. Exit from the *Extract Graphics Fault Tree* screen and return to the *Fault Tree Graphics System* menu. Place the cursor on the "Build Graphic Trees" option and press <enter> to access the *Primary Fault Tree Graphics* menu. Place the cursor on the "FILE" option and press the **left mouse button** to access the *FILE* menu.

10. Place the cursor on the "LIST" option and press the **left mouse button**. Verify that a list of files appears on the screen. Place the cursor on one of the files and press the **left mouse button**. Verify that the selected fault tree appears on the screen.

11. Place the cursor on the "EXIT" option and return to the *Fault Tree Graphics System* menu. Place the cursor on the "Graphical Pager" option and press <enter> to access the *Fault Tree Pager* screen.

12. Place the cursor on the fault tree named ECS. Record the file name and press <enter>. A fault tree will appear with a menu.

13. Place the cursor on the "Show" option and press the **left mouse button**. The message **"Pick a shape"** will be displayed in the bottom left corner of the screen. Place the cursor on the top gate shown on the screen and press the **left mouse button**. Verify that the symbol name appears in the bottom left corner on the screen. Record the symbol name. Press the **right mouse button** to terminate the "Show" option.

14. Place the cursor on the "Find" option and press the **left mouse button**. The message **"Enter name or <CR> for ECS>"** will be displayed in the bottom left corner of the screen. Enter the symbol name recorded in Step 13. Verify that a broken dotted line surrounds the specified symbol.

15. Place the cursor on the "No Tab" option and press the **left mouse button**. Verify that the "No Tab" option changes to the "Table" option. Place the cursor on the "Boxed" option and press the **left mouse button**. Verify that the "Boxed" option changes to the "No Box" option.

16. Place the cursor on the "Page" option and press the **left mouse button**. The message **"Select a Gate for the desired page break"** will be displayed in the bottom left corner of the screen.

17. Place the cursor on the gate for the desired page break and press the **left mouse button**. The message **"Enter file name, or type return for file ECS-TRAIN-B>"** will be displayed in the bottom left corner of the screen. Press **<enter>**. Verify that a new fault tree with the transfer symbol appears on the screen. Press the **right mouse button** to cancel the "Page" option.

18. Place the cursor on the "EXIT" option and press the **left mouse button** to return to the *Fault Tree Pager* screen. Verify that a new file name was added to the *Fault Tree Pager* screen.

19. Exit out of the *Fault Tree Pager* screen and return to the *Fault Tree Graphics System* menu. Place the cursor on the *Build Graphics Tree* menu and press **<enter>** to access the *Primary Fault Tree Graphics* menu.

20. Place the cursor on the "FILE" option and press the **left mouse button** to access the *FILE* menu. Then place the cursor on the "LIST" option and press **<enter>**. Verify that the new file name created by the pager was added to the list of files that appears on the screen.

21. Place the cursor on the file name ECS and press the **left mouse button** to access the fault tree. Return to the "FILE" option and press the **left mouse button** to access the *FILE* menu.

22. Place the cursor on the "Tran→" option and press the **left mouse button**. The message **"Pick the TRANSIN for transfer"** will be displayed in the bottom left corner of the screen. Place the cursor on the transin symbol and press the **left mouse button**. Verify that the transfer occurred to the new fault tree.

23. Exit and return to the *IRRAS* menu.

24. **IF:**   Steps 1 through 23 were performed successfully

       THEN: Go to Step 25.

       ELSE: Record which step(s) was unsuccessful. Generate an anomaly report and submit it to the SAPHIRE 4.0 V&V committee.

25. End of Section 7.

F-5. PROCEDURE 2—IRRAS NONVITAL EVENT TREE FEATURES

F-5.1 Introduction

This procedure tests the nonvital features associated with the event tree analysis functions included in IRRAS. These nonvital event tree analysis features include event tree construction, plot trees, load graphic trees, extract event trees, and link event trees. Table F-3 identifies the features, success criteria, and the specific options that are tested in each of the sections. Also, the success criteria for the specific event tree options tested are listed in the table in

Table F-3. IRRAS event tree test procedure specifications.

Sections	Feature group	Features	Success criteria	Options tested
1	Create Event Trees	File	To load and perform file manipulations	New, Save, Laser
		Edit	To modify the event tree	Add, Copy, and Delete
2	Create Event Trees	Ends	To add the sequence name, endstate name, and other fields	Edit Header, End States, Place Header, Edit Height, Edit Size, Font
	Plot Trees	Hpgl File Raster File	To develop a printout where the event tree diagram is identical in content and structure	Hpgl File Raster File
3	Create Event Trees	File	To load and perform file manipulations	List
		Tops	To make changes to the top events	Add, Del, Name, Width, Size, Font and DESC
		Show	To clear the screen and re-display the defined diagram	Show
4	Create Event Trees	File	To load and perform file manipulations	Load, Save
		Text	To add text to a diagram at any location in any size and color	Write, Move, Erase, Font, Copy, Color, Size, Just, Font
5	Create Event Trees	View	To change the position and size of the displayed diagram	Page ↑, Page ↓, Page ←, Page →, Scroll, Zout, Zres, Zin, Sh Text , No Text, Grid
		File	To load and perform file manipulations	Event, Save, File?
6	Graphics Load	Load	To load graphic trees	Load

Table F-3. IRRAS event tree test procedure specifications.

Sections	Feature group	Features	Success criteria	Options tested
	Link Event Trees	Generate Sequences	To generate sequence logic for event trees	Generate Sequences
7	Extract Event Trees	Display Extracted Trees	To display all the extracted trees	Display Extracted Trees
		Clear Extracted Trees	To clear all extracted trees from a file	Clear Extracted Trees
		Extracted Trees	To remove all extracted trees from data base	Extracted trees
8	Create Event Trees	Transfer	To add, delete, and modify transfer files names as well as transfer to and from files	Add, Tran→ ←Tran, Modify, Delete
		Edit	To modify the event tree.	Page, and Pass
		Line	To select a color for the lines in a diagram	Color Bar
		Text	To set a default color for text in a diagram	Laser
9	Link Event Trees	Sequence Editor	To edit sequence names, endstates and assigned frequencies after an event tree has been created	Header, Line Edit, and Frequency
10	Link Event Trees	Sequence Editor	To edit sequence names, endstates and assigned frequencies after an event tree has been created	Status @, Global Replace, Change Transfer, and Transfer
11	Link Event Trees	Link Editor	To specify linkage rules for an event tree	Add, Insert, Delete, Restore, and Copy

each of the sections . The steps in each section of this procedure should be followed. If any step is not completed successfully, an anomaly report should be generated and submitted to the SAPHIRE 4.0 V&V committee.

F-5.2 Section 1

1. Enter the \PRADATA.B1 directory. Type **IRRAS** and press <enter> to access the *IRRAS* menu. Place the cursor on the "MODIFY Data Base" option and press <enter> to access the *Modify DataBase* menu.

2. At the *Modify Database* menu, select the "Family" option and press <enter> to access the *Edit Family* screen.

3. At the *Edit Family* screen, type **A** and press <enter> to access the *Add Family* screen. Type **Event** as the new family name and press <enter>. Exit and return to the *IRRAS* menu.

4. Place the cursor on the "Select Family" option and press <enter> to access the *Select Family* screen. Place the cursor on the family named Event, and press <enter> to return to the *IRRAS* menu. On the top of the *IRRAS* menu, the family EVENT will be displayed.

5. Place the cursor on the "CREATE Event Trees" option and press <enter> to access the *Event Trees Graphics System* menu. Then place the cursor on the "Create Event Trees" option and press <enter> to access the *Primary Event Tree Graphics* menu.

6. Place the cursor on the "FILE" option and press the **left mouse button** to access the *FILE* menu.

7. The Large LOCA event tree shown in Figure F-3 will now be created. Place the cursor on the "New" option and press the **left mouse button**. The message "Enter Initiating Event or Top Name >" will be displayed in the bottom left corner of the screen and the menu will disappear.

8. Type A for the initiating event name and press <enter>. The message "**Is this an Initiating Event?** >" will be displayed in the bottom left corner of the screen. Type Y and press <enter>.

9. The message "**Enter Event Name #1:**" will be displayed in the bottom left corner of the screen. Type **RPS** and press <enter>.

10. The message "**Enter Event Name #2:**" will be displayed in the bottom left corner of the screen. Type **LPI** and press <enter>.

11. The message "**Enter Event Name #3:**" will be displayed in the bottom left corner of the screen. Type **LCS** and press <enter>.

12. The message "**Enter Event Name #4:**" will be displayed in the bottom left corner of the screen. Type **RHR** and press <enter>.

13. The message "**Enter Event Name #5:**" will be displayed in the bottom left corner of the screen. Type **VLJ** and press <enter>.

14. The message "**Enter Event Name #6:**" will be displayed in the bottom left corner of the screen. Press <enter> to end this option. Verify that the new event tree headings are successfully displayed on the screen.

15. From the *Primary Event Tree Graphics* menu, place the cursor on the "EDIT" option and press the **left mouse button** to access the *EDIT* menu.

16. Place the cursor on the "Add" option from the *EDIT* menu and press the **left mouse button**. The message "**Pick the intersecting point of the new branch.**" will be displayed in the bottom left corner of the screen.

17. Place the cursor on the top branch under the LPI event and press the **left mouse button**. The message "**Pick the vertical position of the new branch.**" will be displayed in the bottom left corner of the screen and a small outline box will appear on the selected branch. Press the **left mouse button** again. Verify that the branch was added.

18. Continue adding new branches until the event tree is the same as the event tree shown in Figure F-3. When completed, press the **right mouse button** to terminate the "Add" option.

19. Place the cursor on the "Copy" option and press the **left mouse button**. The message "**Pick beginning of the branch to copy**" will be displayed in the bottom left corner of the screen.

NUREG/CR-6145

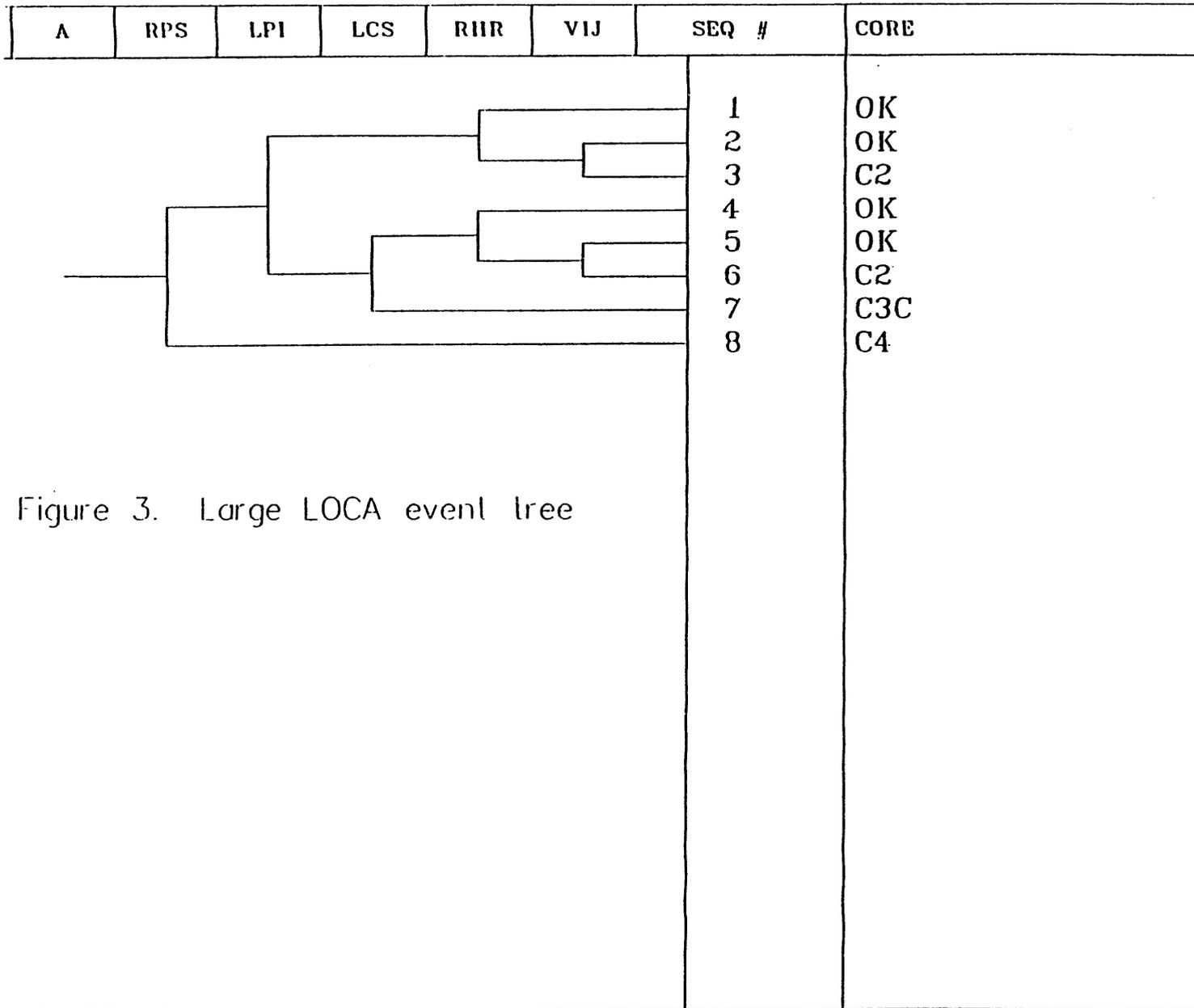| A	RPS	LPI	LCS	RHR	VIJ	SEQ #	CORE
						1	OK
						2	OK
						3	C2
						4	OK
						5	OK
						6	C2
						7	C3C
						8	C4

Figure 3.   Large LOCA event tree

Figure F-3. Large LOCA event tree.

20. Place the cursor on the top branch of the RPS event and press the **left mouse button**. Verify that the branch is highlighted and the message **"Pick copy location"** is displayed in the bottom left corner of the screen.

21. Place the cursor on the bottom branch of the RPS event and press the **left mouse button**. Verify that the branch that was selected is copied at the new location. The event tree displayed on the screen should now have 14 sequences. Press the **right mouse button** to terminate the "Copy" option.

22. Place the cursor on the "Del" option and press the **left mouse button**. The message **"Pick the branch to be deleted."** will be displayed in the bottom left corner of the screen.

23. Place the cursor on the lower branch of the RPS event and press the **left mouse button**. Verify that the branch is highlighted and the message **"Delete highlighted branch? Left = YES, Right = NO"** is displayed in the bottom left corner of the screen.

24. Press the **left mouse button**. Verify that the branch was deleted. Press the **right mouse button** to terminate the "Del" option.

25. Before preceding to the next step, make sure that the event tree is the same as the event tree shown in Figure F-3. Add or delete event tree branches as needed.

26. Place the cursor on the *EDIT* menu and press the **right mouse button** to terminate the *EDIT* menu. At the *Primary Event Tree Graphics* menu, place the cursor on the "FILE" option and press the **left mouse button** to access the *FILE* menu.

27. Place the cursor on the "Laser" option and press the **left mouse button** to obtain a printout of the event tree. Verify that the structure of the event tree on the printout is the same as the structure of the event tree displayed on the screen and shown in Figure F-3.

28. **IF:** Steps 1 through 27 were performed successfully,

    THEN: Go to Step 29.

    ELSE: Record which step(s) was unsuccessful. Generate an anomaly report and submit it to the SAPHIRE 4.0 V&V committee.

29. End of Section 1.

F-5.3 Section 2

1. Return to the *Primary Event Tree Graphics* menu. Place the cursor on the "ENDS" option and press the **left mouse button**.

2. Place the cursor on the "Edit Header" option and press the **left mouse button** to access a window. This window contains the following headings: Sequence names, Endstates-names, Frequency, and Extra-#1. The message **"Pick the box to be edited"** will be displayed in the bottom left corner of the screen.

3. Place the cursor on the box that contains the heading Sequence names and press the **left mouse button**. The message **"Enter text"** will be displayed in the bottom left corner of the screen. Type **CORE** and press <enter>. Verify that the window containing the header was revised.

4. Place the cursor on the first column with the "N" option and press the **left mouse button**. Verify that the "N" option changes to a "Y."

5. Verify that the columns for the remaining headers (i.e. End-States-Names, Frequency, and Extra-#1) have the "N" option. If they do not, place the cursor on the affected column and press the **left mouse button** to change to the "N" option.

6. Press the **right mouse button**. Verify that CORE header is displayed on the top of the screen.

7. Return to the *ENDS* menu. Place the cursor on the "End States" option and press the **left mouse button**. The message **"Pick the leaf corresponding to the desired endstate information"** will be displayed in the bottom left corner of the screen.

8. Move the cursor to the end of first sequence and press the **left mouse button**. Verify that the window appears on the screen and the Core, End-State-Names, Frequency, and Extra #1 headers are displayed in the window.

9. Place the cursor on the row under the CORE header and press the **left mouse button**. The message **"Enter text:"** will be displayed in the bottom left corner of the screen. Type **OK** and press <enter>. Verify the typed text in the window. Press the **right mouse button**. Verify the word OK for sequence 1.

10. Place the cursor at the end of the second sequence and press the **left mouse button** to access a window. The message **"Pick the box to be edited"** will be displayed in the bottom left corner of the screen.

11. Place the cursor on the row under the CORE header and press the **left mouse button**. The message **"Enter text:"** will be displayed on the bottom left corner of the screen. Type **OK** and press <enter>. Verify the typed text is displayed in the window. Press the **right mouse button**. Verify the word OK for sequence 2.

12. Repeat Step 11 for the remaining six sequences, but type the appropriate end-state as shown in Figure F-3. Press the **right mouse button** to cancel the "End States" option.

13. Place the cursor on the "Place Header" option and press the **left mouse button**. The message **"Pick the header that is to be relocated"** will be displayed in the bottom left corner of the screen.

14. Place the cursor on the Core header and press the **left mouse button**. The message **"Pick the placement point for the header"** will be displayed in the bottom left corner of the screen.

15. Move the cursor to a new location and press the **left mouse button**. Verify that the header was moved to the desired location.

16. Repeat Steps 14 and 15 and move the header back to its proper location. Press the **right mouse button** to terminate the "Place Header" option.

17. Place the cursor on the "Edit Height" option and press the **left mouse button**. The message **"Enter new node height: >"** will be displayed in the bottom left corner of the screen. Type **3** for the height value and press the **left mouse button**. Verify that the height or distance between the branches of the event tree have changed on the screen.

18. Place the cursor on the "Edit Size" option and press the **left mouse button**. The message **"Enter new end state text size: >:"** will be displayed in the bottom left corner of the screen. Enter a value of your choice and press the **left mouse button**. Verify that the end state text size has changed on the screen.

19. Place the cursor on the "FONT" option and press the **left mouse button** to access the *FONT* menu. Place the cursor on the desired font and press the **left mouse button**. Verify on the screen that the end-state text changed to the desired font.

20. Repeat Step 19 for the remaining fonts, but select the original font when completed. Then place the cursor on the *ENDS* menu and press the **right mouse button** to cancel this menu.

21. Return to the *Primary Event Tree Graphics* menu. Place the cursor on the "FILE" option and press the **left mouse button** to access the *FILE* menu.

22. Place the cursor on the "Save" option and press the **left mouse button**. The message "**Enter file name or CR for file NONAME.ETG**" will be displayed in the bottom left corner of the screen.

23. Type the file name **LOCA** and press the **left mouse button**. Verify that the message "**File Saved**" appears in the bottom left corner on the screen.

24. Place the cursor on the "EXIT" option and press the **left mouse button** to access the *Event Tree Graphics System* menu. Place the cursor on the "Plot Trees" option and press **<enter>** to access the *Plot Graphics* screen.

25. Type **H** and place the cursor on the LOSP event tree. Then press **<enter>** to generate a HPGL file. Type **R** and place the cursor on the LOSP event tree. Then press **<enter>** to generate a RAS file.

26. Exit the IRRAS program and go to the PRADATA.B1\XXXXX subdirectory. Note: *The subdirectory XXXXX is the family name.* Type **..\PLOTHPG *.HPG** and press **<enter>** to obtain a printout of the HPGL files. Note: *PLOTHPG will only work on an HP LaserJet III or IIIP.* When the printout is complete, type **..\PLOTRAS *.RAS** and press **<enter>** to obtain a printout of the RAS files. Note: *PLOTRAS will work on any laser printer.*

27. Verify that the LOCA event tree printouts correspond to the LOCA event tree shown in Figure F-3.

28. **IF:** Steps 1 through 27 were performed successfully

    THEN: Go to Step 29.

    ELSE: Record which step(s) was unsuccessful. Generate an anomaly report and submit it to the SAPHIRE 4.0 V&V committee.

29. End of Section 2.

F-5.4 Section 3

1. Return to the PRADATA.B1 directory, type **IRRAS**, and press **<enter>** to access the *IRRAS* menu. Select the "Create Event Trees" option and press **<enter>** to access the *Event Tree Graphics System* menu. Place the cursor on the "Create Event Tree" option and press **<enter>** to access the *Primary Event Tree Graphics* menu.

2. Place the cursor on the "FILE" option and press the **left mouse button** to access the *FILE* menu. Place the cursor on the "List" option and press the **left mouse button**. Verify that the file name LOCA is displayed on the list that appears on the screen. Then place the cursor on the file name and press the **left mouse button**. Verify that the event tree is displayed on the screen.

3. Place the cursor on the "TOPS" option and press the **left mouse button** to access the *TOPS* menu. Place the cursor on the "ADD" option and press the **left mouse button**. The message "**Pick the event that will follow the new even** ' will be displayed in the bottom left corner of the screen.

4. Place the cursor on LCS and press the **left mouse button**. The message "**Enter new event name**" will be displayed in the bottom left corner of the screen. Type a three letter acronym of your choice and press **<enter>**. Verify that the new event name appears after LPI.

5. Repeat Steps 3 and 4, but place the cursor on the event name LPI. Verify that the acronym appears before the event name LPI.

6. Return to the *TOPS* menu. Place the cursor on the "Del" option and press the **left mouse button**. The message **"Pick the event that is to be deleted"** will be displayed in the bo⋯⋯ ⋯ left corner of the screen. Place the cursor on the event name that was added in Step 4. Verify tnat the event name was deleted.

7. Repeat Step 6 for the remaining event name. Verify that the event name was deleted.

8. Place the cursor on the "Name" option on the *TOPS* menu and press the **left mouse button**. The message **"Pick an event name"** will be displayed in the bottom left corner of the screen.

9. Place the cursor on the event name and press the **left mouse button**. Verify that the event name appears in the bottom left corner on the screen. Press the **right mouse button** to cancel the "Name" option.

10. Place the cursor on the "Width" option and press the **left mouse button**. The message **"Enter top width size >"** and the width size will be displayed in the bottom left corner of the screen. Record the width size shown with the message.

11. Type **4** and press the **left mouse button**. Verify that the width of the event tree and event names has changed on the screen.

12. Repeat Steps 10 and 11, but enter the original size that was recorded in Step 10. Verify that the event tree and event names return to the original width.

13. Place the cursor on the "Size" option on the *Tops* menu and press the **left mouse button**. The message **"Enter new top name text size: >"** will be displayed in the bottom left corner of the screen.

14. Record the text size that is shown with the message. Then type **1.0** for the text size and press the **left mouse button**. Verify that the text size of the top event names on the screen has changed.

15. Repeat Step 13, but type the original text size that was recorded in Step 14. Verify that the text size of the top event names returns to the original size.

16. Place the cursor on the "FONT" option and press the **left mouse button** to access the *FONT* menu. Place the cursor on a desired font and press the **left mouse button**. Verify that the top event names change to the desired font.

17. Repeat Step 16 for the remaining fonts. After all the fonts have been selected, return to the original font by selecting the first font on the list.

18. Place the cursor on the "DESC" option and press the **left mouse button** to access the *DESC* menu.

19. Place the cursor on the "Line" option and press the **left mouse button**. The message **"Enter new top description line count: <old count>"** will be displayed in the bottom left corner of the screen. Type **4**. Verify that the line count for the event descriptions changed to the desired line count.

20. Place the cursor on the "EDIT" option from the *DESC* menu and press the **left mouse button**. The message **"Pick the event description to be edited"** will be displayed in the bottom left corner of the screen. Place the cursor on an event name and press <enter>. A small box will appear in the top left corner on the screen.

21. Type some text in the box and press <ESC> key. Verify that the text appears above the event names on the screen. Press the **right mouse button** to cancel the "Edit" option.

22. Place the cursor on the "Size" option and press the **left mouse button**. The message **"Enter new top description text size"** will be displayed in the bottom left corner of the screen. Enter a text size of **2.00**. Verify that the event description is displayed in its new size.

23. Place the cursor on the "FONT" option and press the **left mouse button** to access the *FONT* menu. Place the cursor on the desired font and press the **left mouse button**. Verify that the event description above the event names changed to the desired font.

24. Repeat Step 23 for the remaining fonts. When completed, return to the original font by selecting the first font on the list.

25. Return to the *Primary Event Tree Graphics* menu. Place the cursor on the "Show" option and press the **left mouse button**. Verify that the all menus are cleared from the screen.

26. Place the cursor on the "EXIT" option and press the **left mouse button**. The message **"ARE YOU SURE? Enter "Y" to Quit anyway"** will be displayed in the bottom left corner of the screen. Type **Y** to access the *Event Tree Graphics System* menu.

27. IF: Steps 1 through 26 were performed successfully

    THEN: Go to Step 28.

    ELSE: Record which step(s) was unsuccessful. Generate an anomaly report and submit it to the SAPHIRE 4.0 V&V committee.

28. End of Section 3.

F-5.5 Section 4

1. Place the cursor on the "Create Event Tree" option and press <enter> to return to the *Primary Event Tree Graphics* menu. Place the cursor on the "0.50" option and press the **left mouse button**. The message **"Enter new text size"** will be displayed in the bottom left corner of the screen. Type **1.5** and press <enter>. Verify the change on the *Primary Event Tree Graphics* menu.

2. Place the cursor on the "FILE" option and press the **left mouse button** to access the *FILE* menu. Place the cursor on the "Load" option and press the **left mouse button**. The message **"Enter file name>"** will be displayed in the bottom left corner of the screen. Type **LOCA** and press <enter>. Verify that the event tree appears on the screen.

3. Place the cursor on the "TEXT" option and press the **left mouse button** to access the *TEXT* menu.

4. Place the cursor on the "Write" option and press the **left mouse button**. The message **"Pick the text placement location"** will be displayed in the bottom left corner of the screen.

5. Place the cursor under the event tree and press the **left mouse button** to access the text window. Type **Figure F-3. Large LOCA event tree** and press the **<ESC>** key . Verify that the text appears on the screen. Press the **right mouse button** to cancel the "Write" option.

6. Return to the *Primary Event Tree Graphics* menu. Place the cursor on the "FILE" option and press the **left mouse button** to access the *FILE* menu. Place the cursor on the "Save" option and press the **left mouse button**. The message **"Enter file name or CR for LOCA.ETG"** will be displayed in the bottom left corner of the screen.

7. Press <enter> and the message "**File already exists! Replace file? <Y/N>**" will be displayed in the bottom left corner of the screen. Type **Y** to save the file. Place the cursor on the *FILE* menu and press the **right mouse button** to cancel this menu.

8. Place the cursor on the "FONT" option and press the **left mouse button** to access the *FONT* menu. Place the cursor on one of the font options and press the **left mouse button**.

9. Return to the *Text* menu. Place the cursor on the "Write" option and press the **left mouse button**. The message "**Pick the text placement location**" will be displayed in the bottom left corner of the screen.

10. Place the cursor at a location under the event tree. Press the **left mouse button** to access the text window. Type text of your choice and press the <ESC> key. Verify that the text changed to the desired font on the screen. Press the **right mouse button** to terminate the "Write" option

11. Repeat Steps 8 through 10 for several fonts. Verify that the text changed to the desired fonts.

12. Place the cursor on the "Move" option and press the **left mouse button**. The message "**Pick region to be moved - press Cancel to quit**" will be displayed in the bottom left corner of the screen.

13. Place the cursor near the text to be moved and press the **left mouse button**. Drag the cursor until the box surrounds the text and press the **left mouse button** again. The message "**Pick reference point - press CANCEL to quit**" will be displayed in the bottom left corner of the screen.

14. Place the cursor on the text and press the **left mouse button**. Position the box at a new location where the text can be moved and press the **left mouse button**. Verify that text has moved on the screen. Press the **right mouse button** twice to terminate the "Move" option.

15. Place the cursor on the "Copy" option and press the **left mouse button**. The message "**Pick region to be copied - press CANCEL to quit**" will be displayed in the bottom left corner of the screen.

16. Place the cursor near the text to be copied and press the **left mouse button**. Drag the cursor until the box surrounds the text and press the **left mouse button** again. The message "**Pick reference point - press CANCEL to quit**" will be displayed in the bottom left corner of the screen.

17. Place the cursor on the text and press the **left mouse button**. Position the box at a new location where the text can be copied and press the **left mouse button**. Verify that text was copied on the screen. Press the **right mouse button** twice to terminate the "Copy" option.

18. Place the cursor on the "Erase" option and press the **left mouse button**. The message "**Pick region to be deleted**" will be displayed in the bottom left corner of the screen.

19. Place the cursor near the text that was copied and press the **left mouse button**. Drag the cursor until the box surrounds the text and press the **left mouse button** again. The message "**Delete this region? left = YES, Right = NO**" will be displayed in the bottom left corner of the screen.

20. Press the **left mouse button**. Verify that the text on the screen was deleted. Repeat Step 19 to delete all the text that was typed in Step 10. When completed, press the **right mouse button** to cancel the "Erase" option.

21. Return to the *TEXT* menu. Place the cursor the "EDIT" option and press the **left mouse button** to access the *EDIT TXT* menu.

22. Place the cursor on the "Color" option and press the **left mouse button**. The message "**Box the text to be changed**" will be displayed in the bottom left corner of the screen.

23. Place the cursor near the text to be changed and press the **left mouse button**. Drag the cursor until the box surrounds the text and press the **left mouse button** again. The message **"Pick the new text color from the color bar"** will be displayed in the bottom left corner of the screen.

24. Place the cursor on one of the colors from the *Color Bar* menu and press the **left mouse button**. Verify that the text changed to the selected color. Press the **right mouse button** to terminate the "Color" option.

25. Repeat Steps 22 through 24 for the remaining colors.

26. Place the cursor on the "Size" option and press the **left mouse button**. The message **"Box the text to be changed"** will be displayed in the bottom left corner of the screen.

27. Place the cursor near the text to be modified and press the **left mouse button**. Drag the cursor until the box surrounds the text and press the **left mouse button** again. The message **"Enter the new text size >"** will be displayed in the bottom left corner of the screen.

28. Type a text size of your choice and press the **left mouse button**. Verify that the size of the selected text was changed.

29. Place the cursor on the "Just" option and press the **left mouse button**. The message **"Box the text to be changed"** will be displayed in the bottom left corner of the screen.

30. Place the cursor near the text to be modified and press the **left mouse button**. Drag the cursor until the box surrounds the desired text and press the **left mouse button** again. The message **"Enter text justification - ('L'=Left, 'R'=Right, 'C'=Center) >"** will be displayed in the bottom left corner of the screen.

31. Type **R** and press **<enter>**. Verify that the selected text on the screen has changed to right justification.

32. Type **L** and press **<enter>**. Verify that the selected text on the screen has changed to left justification.

33. Type **C** and press **<enter>**. Verify that the selected text on the screen has changed to center justification. Press the **right mouse button** to terminate the "Just" option.

34. Place the cursor on the "FONT" option and press the **left mouse button** to access the *Font* menu. Place the cursor at one of the desired fonts and press the **left mouse button**. The message **"Box the text to be changed"** will be displayed in the bottom left corner of the screen.

35. Place the cursor near the text to be modified and press the **left mouse button**. Drag the cursor until the box surrounds the desired text and press the **left mouse button** again. Verify that the text changed to the desired font. Press the **right mouse button** to terminate the "FONT" option. Press the **right mouse button** again to cancel the *TEXT* menu.

36. Return to the *Primary Event Tree Graphics* menu. Place the cursor on the "Exit" option and press **<enter>**. The message **"ARE YOU SURE? Enter Y to Quit anyway"** will be displayed in the bottom left corner of the screen. Type **Y** to access the *Event Tree Graphics System* menu.

37. <u>IF</u>:   Steps 1 through 36 were performed successfully,

    <u>THEN</u>: GO TO Step 38.

    <u>ELSE</u>: Record which step or steps were unsuccessful. Generate an anomaly report and submit it to the SAPHIRE 4.0 V&V committee.

38. End of Section 4.0.

F-5.6 Section 5

1. Place the cursor on the "Create Event Trees" option and press <enter> to return to the *Primary Fault Tree Graphics* menu.

2. Place the cursor on the "FILE" option and press the **left mouse button** to access the *FILE* menu. Place the cursor on the "List" option and press the **left mouse button**. Then place the cursor on the LOCA file name and press the **left mouse button** to display the event tree on the screen.

3. Place the cursor on the "VIEW" option and press the **left mouse button** to access the *VIEWS* menu.

4. Place the cursor on the "Page ↑" option and press the **left mouse button**. This will shift the event tree up a page. Verify that the screen is blank.

5. Place the cursor on the "Page ↓" and press the **left mouse button**. This will shift the event tree down a page. Verify that the event tree is displayed on the screen.

6. Place the cursor on the "Page ←" option and press the **left mouse button**. This will shift the event tree to the left one page. Verify that the screen is blank.

7. Place the cursor on the "Page →" option and press the **left mouse button**. This will shift the event tree to the right one page. Verify that the event tree is displayed on the screen.

8. Place the cursor on the "Scroll" option and press the **left mouse button**. Verify that a white outline box appears and a cursor is in the center of the box. Place the cursor at a desired location and press the **left mouse button**. Verify that the figure moves relative to the cursor position.

9. Place the cursor on the "Zout" option and press the **left mouse button**. Verify that the figure is reduced approximately 50%. Place the cursor on the "Zres" option and press the **left mouse button**. Verify that the display returns to the original display size.

10. Place the cursor on the "Zin" option and press the **left mouse button**. The message **"Pick first corner"** will be displayed in the bottom left corner of the screen.

11. Place the cursor near the figure and press the **left mouse button**. Drag the cursor until the box surrounds the figure and press the **left mouse button**. Verify that the figure is magnified. Place the cursor on the "Zres" option and press the **left mouse button**. Verify that the display returns to the original display size.

12. Place the cursor on the "Sh Text" option and press the **left mouse button**. Verify that the option toggles to the "No Text" option. Also verify that written text is no longer displayed on the screen.

13. Place the cursor on the "No Text" option and press the **left mouse button**. Verify that the option toggles to the "Sh Text" option and that the written text **"Figure F-3. Large LOCA event tree"** is displayed on the screen.

14. Place the cursor on the "Grid" option and press the **left mouse button**. Verify that a grid appears on the screen.

15. Place the cursor on the "Grid" option again and press the **left mouse button**. Verify that the original screen background is displayed. Place the cursor on the *VIEWS* menu and press the **right mouse button** to terminate this menu.

16. Return to the *Primary Event Tree Graphics* menu. Place the cursor on the "FILE" option and press the **left mouse button** to access the *FILE* menu.

17. Place the cursor on the "Event" option and press the **left mouse button**. The message **"Type in the New Name or <Return> for LOCA>"** will be displayed in the bottom left corner of the screen. Type **LOCA2** as the new file name and press **<enter>**.

18. Place the cursor on the "Save" option and press the **left mouse button**. The message **"Enter file name or CR for file LOCA ETG"** will be displayed on the screen. Press the **left mouse button** again. The message **"File and Event Tree names do not match! Use File Name for both? <Y/N>"** will be displayed in the bottom left corner of the screen.

19. Type N and press **<enter>**. The message **"File already exists! Replace file? <Y/N>"** will be displayed in the bottom left corner of the screen. Type **Y**, the message **"File Saved"** will be displayed in the bottom left corner of the screen.

20. Place the cursor on the "EXIT" option from the *Primary Event Tree Graphics* menu and press the **left mouse button** to access the *Event Tree Graphics System* menu. Place the cursor on the "eXtract Event Trees" option and press **<enter>** to access the *Extract Graphic Event Trees* screen. Verify that the file name LOCA2 was added to the file names displayed on the screen.

21. Type **X** and place the cursor on the LOCA2 file name to extract the file. Press **<enter>** and the message **"LOCA2.ETG successfully extracted"** will be displayed in the bottom left corner of the screen.

22. Exit the *Extract Graphic Event Trees* screen and return to the *Event Tree Graphics System* menu. Place the cursor on the "Create Event Trees" option and press **<enter>** to access the *Primary Event Graphics System* menu.

23. Place the cursor on the "FILE" option to access the *FILE* menu. Place the cursor on the "LIST" option and press the **left mouse button**. Verify that both file names appear on the screen. Place the cursor on the file name called LOCA2 and press the **left mouse button**. Verify that the event tree appears on the screen.

24. Return to the "FILE" option located at the *Primary Event Graphics System* menu and press the **left mouse button** to access the *FILE* menu. Place the cursor on the "File ?" option and press the **left mouse button**. Verify that the file name is displayed in the bottom left corner of the screen.

25. Place the cursor on the *FILE* menu and press the **right mouse button** to terminate the *FILE* menu.

26. **IF:** Steps 1 and 24 were performed successfully

  THEN: Go to Step 27

  ELSE: Record which step(s) was unsuccessful. Generate an anomaly report and submit it to the SAPHIRE 4.0 V&V committee.

27. End of Section 5.

F-5.7 Section 6

1. Exit from the IRRAS program and change to the to the PRADATA.B1\XXXXX subdirectory. Note: *The PRADATA.B1 subdirectory is the family name which is EVENT.*

2. Insert the disk that has been included for this procedure into the A drive. Type **Dir A:** and press <enter> to view the files on this disk. Verify that all of the following files are on the disk: LOSP.ETG, CCS.DLS, ECS.DLS, EVENT.BEA, EVENT.BED, and EVENT.BEI.

3. Copy all of these files to the C:\PRADATA.B1\EVENT subdirectory. Then change to the C:\PRADATA.B1 directory and type **IRRAS** to access the *IRRAS* menu.

4. Place the cursor on the "UTILITY Options" option and press <enter> to access the *IRRAS Utilities* menu. Place the cursor on the "Load MAR-D Data Files" option and press <enter> to access the *Select Data Source* screen. Place the cursor on the "MAR-D" option and press <enter> to access the *MAR-D Data* menu. Place the cursor on the "Basic Event" option and press <enter> to access the *Basic Event Data* menu.

5. Place the cursor on the "Descriptions" option and press <enter> to access the *Load* screen. Verify the name EVENT in the window. Type **L** and place the cursor on the file name and press <enter>. Verify that the message **"Load complete"** is displayed on the bottom left corner of the screen. Exit this screen and return to the *Basic Event Data* menu.

6. Place the cursor on the "Rate Information" option and press <enter> to access the *Load* screen. Verify the file name in the window. Type **L** and place the cursor on the file name and press <enter>. Verify that the message **"Load compete"** is displayed on the bottom left corner of the screen. Exit this screen and return to the *Basic Event Data* menu.

7. Place the cursor on the "Attributes" option and press <enter> to access the *Load* screen. Again verify the name EVENT in the window. Type **L** and place the cursor on the file name and press <enter>. Verify that the message **"Load compete"** is displayed on the bottom left corner of the screen. Exit this screen and return to the *Basic Event Data* menu. Exit the *Basic Event Data* menu and return to the *IRRAS* menu.

8. Place the cursor on the "BUILD Fault Trees" option and press <enter> to access the *Fault Tree Graphics System* menu. Place the cursor on the "Load Graphic Trees" option and press <enter> to access the *Load Fault Tree Graphics* screen. Verify that the system file names CCS and ECS appear on the screen.

9. Type **L** and press <enter>. The message **"Process all entries? (Y/N)"** will be displayed in the bottom left corner of the screen. Type **Y** and observe the message **"Fault tree successfully loaded"** in the bottom left corner on the screen. Exit from the *Load Fault Tree Graphics* screen and return to the *Fault Tree Graphics System* menu. Place the cursor on the "Build Graphics Trees" option and press <enter> to access the *Fault Tree Graphics* menu.

10. Place the cursor on the "FILE" option and press the **left mouse button** to access the *FILE* menu. Place the cursor on the "List" option and press the **left mouse button**. Verify that the files CCS and ECS appear on the screen. Press the **right mouse button** to cancel the "List" option. Exit the *Primary Fault Tree Graphics* menu and return to the *IRRAS* menu.

11. Place the cursor on the "Create Event Tree" option and press <enter> to access the *Event Tree Graphics System* menu. Place the cursor on the "Graphics Load" option and press <enter> to access the *Load Graphics Trees* screen.

12. Type **L**, place the cursor on the LOSP file name, and press <enter>. Exit the *Load Event Tree Graphics* screen and return to the *Event Tree Graphics System* menu. Place the cursor on the *Create Event Trees* menu and press <enter> to access the *Primary Event Tree Graphics* menu.

13. Place the cursor on the "FILE" option and press the **left mouse button** to access the *FILE* menu. Place the cursor on the "List" option and press the **left mouse button**. Verify that the file name LOCA

appears on the screen. Press the **right mouse button** to cancel the "List" option. Exit the *Primary Event Tree Graphics* menu and return to the *IRRAS* menu.

14. Place the cursor on the "GENERATE Event Data" option and press **<enter>** to access the *Change Sets* screen. Verify that the screen is blank. Type **G** and press **<enter>**. A window will be displayed in the bottom left corner of the screen.

15. Place the cursor on the "Propagate Event Failure (Y/N)" option. Type **Y** and press **<enter>**. The message **"Event data set to base case (no changes)"** will be displayed in the bottom left corner on the screen. Exit the *Change Sets* screen and return to the *IRRAS* menu.

16. Place the cursor on the "FAULT Tree Analysis" option and press **<enter>** to access the *Fault Tree Analysis* menu. Place the cursor on the "Analyze Systems" option and press **<enter>** to access the *Analyze Systems* screen.

17. Type **G** and place the cursor on the fault tree named CCS. Press **<enter>** to access the *CUT SET GENERATION CUTOFF VALUES* screen. Press **<enter>** again. Verify the total number of cutsets and the cutset value. Record the cut-set value. Press **<enter>** to return to the *Analyze Systems* screen.

18. Place the cursor on fault tree ECS and press **<enter>** to access the *CUT SET GENERATION CUTOFF VALUES* screen. Press **<enter>** again. Verify the total number of cutsets and the cutset value. Record the cut-set value. Exit and return to the *IRRAS* menu.

19. Place the cursor on the "CREATE Event Trees" option and press **<enter>** to access the *Event Tree Graphics System* menu. Place the cursor on the "Link Event Trees" option and press **<enter>** to access the *Link Event Tree* screen.

20. Type **G** and place the cursor on the LOSP file name to generate the sequence logic. Press **<enter>** to access the *Sequence Generation Information* screen. Place the cursor in the space for Report Name. Delete LOSP.LST, type **CON**, and press **<enter>**. Verify the list of sequences on the screen. Note: *The LOSP event tree should have three sequences.* Press **<enter>** to return to the *Link Event Tree* screen.

21. Exit the *Link Event Tree* screen and return to the *Event Tree Graphics System* menu.

22. <u>IF</u>: Steps 1 through 21 were performed successfully

   <u>THEN</u>: Go to Step 23.

   <u>ELSE</u>: Record which step(s) was unsuccessful. Generate an anomaly report and submit it to the SAPHIRE 4.0 V&V committee.

23. End of Section 6.

F-5.8 Section 7

1. From the *Event Tree Graphics System* menu, place the cursor on the "Create Event Trees" option and press **<enter>** to access the *Primary Event Tree Graphics* menu. Place the cursor on the "FILE" option and press the **left mouse button** to access the *FILE* menu. Place the cursor on the "LIST" option and press the **left mouse button**. Record the list of file names that are displayed on the screen. Then press the **right mouse button** to cancel the "LIST" option.

2. Place the cursor on the "EXIT" option and press the **left mouse button** to return to the *Event Tree Graphics System* menu. Place the cursor on the "eXtract Event Trees" option and press **<enter>** to access the *Extract Graphic Event Trees* screen.

3. Type **D** to display extracted trees and press **<enter>**. Verify that a different *Extract Graphic Event Tree* screen appears that contains only the file names. Press **<enter>** to return to the previous *Extract Graphic Event Trees* screen.

4. Type **C** to clear extracted trees and press **<enter>**. Observe the following message in a box on the screen: **"WARNING All existing "*.ETG" files will be deleted!! Do you wish to proceed?"** Type **Y** and press **<enter>**. The message **"All extracted files cleared"** will be displayed in the bottom left corner of the screen.

5. Exit from the *Extract Graphics Event Trees* screen and return to the *Event Tree Graphics System* menu. Place the cursor on the "Create Event Trees" option and press **<enter>** to access the *Primary Event Tree Graphics* menu. Place the cursor on the "FILE" option and press the **left mouse button** to access the *FILE* menu.

6. Place the cursor on the "LIST" option and press the **left mouse button**. Verify that no file names appear on the screen.

7. Place the cursor on the "EXIT" option and press the **left mouse button** to return to the *Event Tree Graphics System* menu. Place the cursor on the "eXtract Event Trees" option and press **<enter>** to access the *Extract Graphic Event Trees* screen.

8. Type **X** and press **<enter>** to extract trees. The message **"Process all entries? (Y/N)"** will be displayed on the bottom left corner on the screen. Type **Y**.

9. Exit the *Extract Graphics Event Tree* screen and return to the *Event Tree Graphics System* menu. Place the cursor on the "Create Event Trees" option and press **<enter>** to access the *Primary Event Tree Graphics* menu. At the *Primary Event Tree Graphics* menu, place the cursor on the "FILE" option and press the **left mouse button** to access the *FILE* menu.

10. Place the cursor on the "LIST" option and press the **left mouse button**. Verify that a list of files appears on the screen. Place the cursor on one of the files and press the **left mouse button**. Verify that the selected event tree appears on the screen.

11. Place the cursor on the "EXIT" option and return to the *Event Tree Graphics System* menu.

12. **IF:**   Steps 1 through 11 were performed successfully,

    THEN: Go to Step 13.

    ELSE: Record which step(s) was unsuccessful. Generate an anomaly report and submit it to the SAPHIRE 4.0 V&V committee.

13. End of Section 7.

F-5.9 Section 8

1. At the *Event Tree Graphic System* menu. Place the cursor on the "Create Event Trees" option and press **<enter>** to access the *Primary Event Tree Graphics* menu.

2. Place the cursor on the "FILE" option and press the **left mouse button** to access the *FILE* menu. Place the cursor on the "List" option and press the **left mouse button**. Place the cursor on the LOCA file name and press the **left mouse button**. Verify that the LOCA event tree appears on the screen.

3. Return to the *Primary Event Tree Graphics* menu. Place the cursor on the "TRAN" option and press the **left mouse button** to access the *TRANS* menu.

4. Place the cursor on the "Add" option and press the **left mouse button**. The message "**Pick the leaf where the transfer is to be placed**" will be displayed in the bottom left corner of the screen.

5. Place the cursor on the sixth sequence and press the **left mouse button**. The message "**Enter transfer file name**" will be displayed in the bottom left corner of the screen. Type **LOSP** and press <enter>. Verify that the letter T is displayed at the end of sequence six. Press the **right mouse button** to cancel the "Add" option.

6. Return to the *Primary Event Tree Graphics* menu. Place the cursor on the "FILE" option and press the **left mouse button** to access the *FILE* menu. Place the cursor on the "Save" option and press the **left mouse button**. The message "**Enter file name or CR for file LOCA.ETG**" will be displayed in the bottom left corner of the screen. Press the **left mouse button** again and the message "**File already exists! Replace file? <Y/N>**" will be displayed in the bottom left corner of the screen. Type Y to save the file.

7. Place the cursor on the *FILE* menu and press the **right mouse button** to cancel the *FILE* menu.

8. Return to the *TRANS* menu. Place the cursor on the "Tran→" option and press the **left mouse button**. The message "**Pick the leaf corresponding to the desired transfer file name**" will be displayed in the bottom left corner of the screen.

9. Place the cursor on the sixth sequence and press the **left mouse button**. Verify that the file transfers to the LOSP event tree.

10. Return to the *Primary Event Tree Graphics* menu. Place the cursor on the "TRAN" option to access the *TRANS* menu again. Place the cursor on the "←Tran" option and press the **left mouse button**. A list of file names will be displayed on the screen. Also the message "**Pick the file to transfer to**" will be displayed in the bottom left corner of the screen.

11. Place the cursor on the LOCA file name and press the **left mouse button**. Verify that the LOCA event tree appears on the screen.

12. Return to the *Primary Event Tree Graphics* menu. Place the cursor on the "TRAN" option and press the **left mouse button** to access the *TRANS* menu again.

13. Place the cursor on the "Modify" option and press the **left mouse button**. The message "**Pick the leaf corresponding to the transfer to be edited**" will be displayed in the bottom left corner of the screen.

14. Place the cursor on the sixth sequence. Press the **left mouse button** and observe the message "**Type in the New Name of <Return> for LOSP>**" displayed in the bottom left corner of the screen.

15. Type a new file name of your choice and press <enter>. In order to verify the new file name, place the cursor on the sixth sequence again and press the **left mouse button**. Verify that the file name is displayed in the bottom left corner of the screen. Press the **right mouse button** twice to terminate "Modify" option.

16. Place the cursor on the "Delete" option and press the **left mouse button**. Observe the message "**Pick the leaf corresponding to the transfer to be deleted**" displayed in the bottom left corner of the screen.

17. Place the cursor on the sixth sequence and press the **left mouse button**. Verify that the transfer letter T is no longer displayed on the screen. Press the **right mouse button** to terminate the "Delete" option.

18. Return to the *Primary Event Tree Graphics* menu. Place the cursor on the "FILE" option and press the **left mouse button** to access the *FILE* menu.

19. Place the cursor on the "Save" option and press the **left mouse button**. Observe the message "**Enter file name or CR for file LOCA.ETG >**" displayed in the bottom left corner of the screen. Press the **left mouse button** again and observe the message "**File already exists! Replace file? <Y/N>**" displayed in the bottom left corner of the screen. Type **Y** to save the file.

20. Return to the *Primary Event Tree Graphics* menu. Place the cursor on the "EDIT" option and press the **left mouse button** to access the *EDIT* menu.

21. Place the cursor on the "Page" option and press the **left mouse button**. The message "**Pick beginning of the branch to page**" will be displayed in the bottom left corner of the screen.

22. Place the cursor on the branch under the LPI event name and press the **left mouse button**. Verify that the branch is highlighted and that the message "**Create Transfer here? [Y]**" is displayed in the bottom left corner of the screen. Type **Y** and press **<enter>**. The message "**Enter the transfer file name >**" will be displayed in the bottom left corner of the screen.

23. Type a file name of your choice and press **<enter>**. Verify that the event tree appears on the screen with the transfer letter T at the end of the sequence. Press the **right mouse button** to terminate the "Page" option.

24. Return to the *Primary Event Tree Graphics* menu. Place the cursor on the "TRAN" option and press the **left mouse button** to access the *TRANS* menu.

25. Place the cursor on the "Tran→" option and press the **left mouse button**. The message "**Pick the leaf corresponding to the desired transfer file name**" will be displayed in the bottom left corner of the screen.

26. Place the cursor on the sequence with the letter T and press the **left mouse button**. The message "**Current drawing not saved. Enter "T" to transfer anyway >**" will be displayed in the bottom left corner of the screen. Type **T**. Verify the new event tree.

27. Return to the "TRAN" option and press the **left mouse button** to access the *TRANS* menu. Place the cursor on the "←Tran" option and press the **left mouse button**. A list will appear on the screen with the new event tree file name. Also, the message "**Pick the file to transfer to**" will be displayed in the bottom left corner of the screen.

28. Place the cursor on the LOCA file name and press the **left mouse button**. Verify on the screen that the file transfers to the event tree.

29. Return to the *Primary Event Tree Graphics* menu. Place the cursor on the "Line" option and press the **left mouse button**. The message "**Pick a new color from the color bar**" will be displayed in the bottom left corner of the screen.

30. Place the cursor on the first color and press the **left mouse button**. Then place the cursor on the "Show" option and press the **left mouse button**. Verify that the event tree changes to the desired color.

31. Repeat Steps 29 and 30 for the remaining colors.

32. Return to the *Primary Event Tree Graphics* menu. Place the cursor on the "TEXT" option and press the **left mouse button**. The message "**Pick a new color from the color bar**" will be displayed in the bottom left corner of the screen.

33. Place the cursor on the first color and press the **left mouse button**. Then place the cursor on the "Show" option and press the **left mouse button**. Verify that the text for event tree changes to the desired color.

34. Repeat Steps 32 and 33 for the remaining colors.

35. Return to the *Primary Event Tree Graphics* menu. Place the cursor on the "↑↓" option and press the **left mouse button.** Verify that the color bar toggles to the new color bar.

36. Return to the *Primary Event Tree Graphics* menu. Place the cursor on the "EDIT" option and press the **left mouse button** to access the *EDIT* menu.

37. Place the cursor on the "Pass" option and press the **left mouse button.** The message **"Pick the branch that is to be made a pass"** will be displayed in the bottom left corner of the screen.

38. Under the VIJ event name, place the cursor on the branch that supports the second and third sequences and press the **left mouse button.** Verify that the branch is highlighted and that the message **"Make highlight branch a pass? Left = YES, Right = NO"** is displayed in the bottom left corner of the screen.

39. Press the **left mouse button** again. Verify that the branch under the VIJ event name was deleted. Press the **right mouse button** to cancel the "Pass" option.

40. Return to the *Primary Event Tree Graphics* menu. Place the cursor on the "FILE" option and press the **left mouse button** to access the *FILE* menu.

41. Place the cursor on the "Save" option and press the **left mouse button.** The message **"Enter file name or CR for file LOCA.ETG >"** will be displayed in the bottom left corner of the screen. Press the **left mouse button** again and the message **"File already exists! Replace file? <Y/N>"** will be displayed in the bottom left corner of the screen. Type **Y** to save the file.

42. Place the cursor on the "Laser" option and press the **left mouse button** to obtain a printout of the modified event tree. Verify from the printout that the event name VIJ for sequences 2 and 3 has been passed.

43. Place the cursor on the "EXIT" option and press the **left mouse button** to access the *Event Tree Graphics System* menu. Place the cursor on the "Link Event Trees" option and press <enter> to access the *Link Event Tree* screen.

44. Type **G** and place the cursor on the LOCA file name in the window. Press <enter> to access the *Sequence Generation Information* screen. Verify that the cursor is in the *Report Name* screen.

45. Delete the report name LOCA.LST, type **Con,** and press <enter>. Verify the sequence list on the screen. Compare the printout obtained from Step 42 and to the figure displayed on the screen. Verify that the event name VIJ was not included in sequences 2 and 3. Press <enter> to return to the *Link Event Tree* screen.

46. IF: Steps 1 through 45 were performed successfully,

    THEN: Go to Step 47.

    ELSE: Record which step(s) was unsuccessful. Generate an anomaly report and submit it to the SAPHIRE 4.0 V&V committee.

47. End of Section 8.

F-5.10 Section 9

1. At the *Link Event Tree* screen, type S and place the cursor on the LOSP file name. Place the cursor on the LOSP file name and press <enter> to access the *Sequence Editor* screen.

2. Type **H** and press <enter> to edit the headers. Observe that there are four headers that can be changed. Type **NAMES** to change the first header. Use the **<right arrow>** key to move the cursor to the second header. Type **STATES** for the second header.

3. Let the third and fourth headers remain as FREQUENCY and EXTRA-#1. Press <enter>. Verify that the changes were made to the NAMES, STATES, FREQUENCY, and EXTRA-#1 headers displayed on the screen.

4. Type **L** and place cursor on the first sequence. Press <enter> and observe the four different blocks under each header. Under the first header type **1-LOSP-ECS**. Use the **<right arrow>** key to move the cursor to the second column under STATES. Delete the numerical value, type **OK**, and press <enter>. Verify that the change for sequence 1 occurred on the screen.

5. Place the cursor on the second sequence and press <enter>. Observe the four different columns under each header. Type **2-LOSP-ECS-CCS** for the first header. Use the **<right arrow>** key to move the cursor to the second column under the word STATES. Type **SMALL-RELEASE** and press <enter>. Verify that the change occurred on the screen for sequence 2.

6. Place the cursor on the third sequence and press <enter>. Type **3-LOSP-ECS-CCS** under the first header NAMES. Use the **<right arrow>** key to move the cursor to the second column under the STATES header, type **LARGE-RELEASE**, and press <enter>. Verify that the change for sequence 3 occurred on the screen.

7. Type **F** and press <enter> to view the frequencies. The message "**Process all entries? (Y/N)**" will be displayed in the bottom left corner of the screen. Type **Y**. Verify the values under the frequency column for sequences 2 and 3.

8. Type **E** and press <   exit. The message "**Save Changes before Exiting? (Y/N)**" will be displayed in the middle   screen. Type **Y** and press <enter> to access the *Link Event Tree* screen. Exit and return to the *Event Tree Graphics System* menu.

9. Place the cursor on the "Create Event Trees" option and press <enter> to access the *Primary Event Tree Graphics* menu. Place the cursor on the "FILE" option and press the **left mouse button** to access the *FILE* menu.

10. Place the cursor on the "List" option and press the **left mouse button**. Place the cursor on the LOSP file name and press the **left mouse button**. Verify that the changes made in the previous steps are correctly displayed the screen.

11. Place the cursor on the "Exit" option and press the **left mouse button** to return to the *Event Tree Graphics System* menu.

12. <u>IF</u>:   Steps 1 through 11 were performed successfully

    <u>THEN</u>: Go to Step 13.

    <u>ELSE</u>: Record which step(s) was unsuccessful. Generate an anomaly report and submit it to the SAPHIRE 4.0 V&V committee.

13. End of Section 9.

F-5.11 Section 10

1. Place the cursor on the "Link Event Trees" option and press <enter> to access the *Link Event Tree* screen. Type S, place the cursor on the LOCA file name, and press <enter> to access the *Sequence Editor* screen.

2. Type @, place the cursor on sequence 2, and press <enter>. Verify that the @ symbol appears on the screen in the END-STATES-NAMES column. Place the cursor on Sequence 3 and press <enter>. Again, verify that the @ symbol appears on the screen in the END-STATES-NAMES column.

3. Type E and press <enter>. The message "Save Changes before Exiting? (Y/N)" will be displayed in the middle of the screen. Type Y and press <enter> to access the *Link Event* screen.

4. Type G and place the cursor on the LOCA file name in the window. Then press <enter> to access the *Sequence Generation Information* window. Observe that the cursor is in the *Report Name* screen.

5. Delete the report name LOCA.LST, type CON, and press <enter>. Verify the sequence list on the screen. Verify that sequences 2 and 3 are not displayed on the screen. Pres · <enter> to return to the *Link Event Tree* screen.

6. Type S and place the cursor on the LOCA file name and press <enter> to access the *Sequence Editor* screen.

7. Type @, place the cursor on sequence 2, and press <enter>. Verify that the @ symbol is deleted from the screen. Place the cursor on sequence 3 and press <enter>. Verify that the @ symbol is deleted from the screen.

8. Type E and press <enter>. The message "Save Changes before Exiting? (Y/N)" will be displayed in the middle of the screen. Type Y and press <enter> to access the *Link Event* screen.

9. Type G and place the cursor on the LOCA file in the window. Then press <enter> to access the *Sequence Generation Information* screen. Observe that the cursor is in the *Report Name* screen.

10. In the *Report Name* screen, delete LOCA.LST, type CON, and press <enter>. Verify the sequence list on the screen. Verify that sequences 2 and 3 are displayed on the screen. Press <enter> and return to the *Link Event Tree* screen.

11. Type S and place the cursor on the LOCA file name and press <enter> to access the *Sequence Editor* screen.

12. Type C, place the cursor on sequence 2, and press <enter>. Verify the letter T on the screen. Type E and press <enter>. The message "Save Changes before Exiting? (Y/N)" will be displayed in the middle of the screen. Type Y and press <enter> to access the *Link Event* screen. Exit the *Link Event* screen and return to the *Event Tree Graphics System* menu.

13. Place the cursor on the "Create Event Trees" option and press <enter> to access the *Primary Event Tree Graphics* menu. Place the cursor on the "FILE" option and press the **left mouse button** to access the *FILE* menu.

14. Place the cursor on the "LIST" option and press the **left mouse button**. Place the cursor on the LOCA file name and press the **left mouse button**. Verify that the LOCA event tree is displayed on the screen and that the transfer letter T is displayed at the end of sequence 2.

15. Return to the *Primary Event Tree Graphics* menu. Place the cursor on the "TRAN" option and press the **left mouse button** to access the *TRANS* menu.

16. Place the cursor on the "Modify" option and press the **left mouse button**. The message **"Pick the leaf corresponding to the transfer to be edited"** will be displayed in the bottom left corner of the screen.

17. Position the cursor on sequence 2 and press the **left mouse button** again and the message **"Type in the New Name or <Return> for >"** will be displayed in the bottom left corner of the screen. Type **LOSP** and press <enter>. Then press the **right mouse button** to terminate the "Modify" option.

18. Return to the *Primary Event Tree Graphics* menu. Place the cursor on the "FILE" option and press the **left mouse button** to access the *FILE* menu.

19. Place the cursor on the "Save" option and press the **left mouse button**. The message **"Enter file name or CR for file name LOCA.ETG>"** will be displayed in the bottom left corner of the screen. Press the **left mouse button** again and the message **"File already exists! replace file? <Y/N>"** will be displayed in the bottom left corner of the screen. Type Y to save the file.

20. Return to the *TRANS* menu. Place the cursor on the "Tran→" option and press the **left mouse button**. The message **"Pick the leaf corresponding to the desired transfer file name"** will be displayed in the bottom left corner of the screen.

21. Place the cursor on sequence 2 and press the **left mouse button**. Verify on the screen that the transfer to the LOSP event tree occurred. Exit the *Primary Event Tree Graphics* menu and return to the *Event Tree Graphics System* menu.

22. Place the cursor on the "Link Event Tree" option and press <enter> to access the *Link Event Tree* screen. Type **S** and place the cursor on the LOCA file name and press <enter> to access the *Sequence Editor* screen.

23. Type **T**, place the cursor on sequence 2, and press <enter>. Verify that the transfer occurred to the LOSP file. Press <enter> and return to the LOCA file.

24. Exit the *Sequence Editor* screen and return to the *Link Editor* screen. Place the cursor on the LOSP file name and press <enter> to access the *Sequence Editor* screen.

25. Type **G** and place the cursor on sequence 2. Verify that 2-LOSP-ECS-CCS and Small-Release appear under the NAMES and STATE headers for sequence 2. Press <enter>. Verify that a window is displayed in the middle of the screen with two blocks entitled Search for and Replace with.

26. Type **2-LOSP-ECS-CCS** in the Search for space. Move the cursor to the Replace with space and type **LOSP-ECS-CCS**. Move the cursor to the In Column space and type 1. Press <enter>. Verify that the changes are made in the *Sequence Editor* screen.

27. Remain on sequence 2 and press <enter> again. Type **Small-Release** in the Search for space. Move the cursor to the Replace with space and type **Large-Release**. Move the cursor to the In Column space, type 2, and press <enter>. Verify that the changes were made on the *Sequence Editor* screen.

28. Type **E** and press <enter>. The message **"Save Changes before Exiting? (Y/N)"** will be displayed in the middle of the screen. Type **Y** and press <enter>. Exit the *Link Event* screen and return to the *Event Tree Graphics System* menu.

29. Place the cursor on the "Create Event Trees" option and press <enter> to access the *Primary Event Tree Graphics* menu. Place the cursor on the "FILE" option and press the **left mouse button** to access the *FILE* menu.

30. Place the cursor on the "LIST" option and press the **left mouse button**. Place the cursor on the LOSP file name and press the **left mouse button**. Verify that the LOSP event tree is displayed on the screen. Also verify the changes made on the screen.

31. Place the cursor on "EXIT" option and return to the *Event Tree Graphics System* screen.

32. IF: Steps 1 through 31 were performed successfully,

    THEN: Go to Step 33.

    ELSE: Record which step(s) were unsuccessful. Generate an anomaly report and submit it to the SAPHIRE 4.0 V&V committee.

33. End of Section 10.

F-5.12 Section 11

1. Exit the *Event Tree Graphics System* menu and return to the *IRRAS* menu. Place the cursor on the "MODIFY DataBase" option and press <enter> to access the *Modify Database* menu.

2. Place the cursor on the "SYstems" option and press <enter> to access the *Edit Systems* screen. Type **A** and press <enter> to access the *Add System* screen. Place the cursor in the space for Name. Type **Syst1** and press <enter>. Verify on the *Edit System* screen that the word Syst1 was added.

3. Press <enter> again to access the *Add System* screen. Type **Syst2** and press <enter>. Verify on the *Edit System* screen that the name Syst2 was added. Exit the *Edit System* screen and return to the *IRRAS* menu.

4. Place the cursor on the "CREATE Event Trees" option and press <enter> to access the *Event Tree Graphic System* screen. Place the cursor on the "Link Event Trees" option and press <enter> to access the *Link Event Tree* screen.

5. Type **G** and place the cursor on the LOCA file name. Press <enter> to access the *Sequence Generation Information* screen. Observe that the cursor is in the *Report Name* screen.

6. Delete LOCA.LST and type **PRN**. Press <enter> to obtain a printout of the sequences and return to the *Link Event Tree* screen.

7. Type **L** and place the cursor on the LOCA file name. Press <enter> to access the *Link Editor* screen. Locate the cursor under the # symbol.

8. Type **A**. Verify the number 1 under the # symbol. Press <enter> to move the cursor to the IF column. Type **A** again. Verify the list of systems in the *Tops* menu.

9. Type **/LPI** and press <enter>. Use the arrow keys to move the cursor to the Then column. Type **A** to access the list of systems in the *Tops* menu. Type **RHR** and press <enter>. The cursor should now be in the Is column. Type **A** again to access the *Systems* window. Type **Syst1** and press <enter>.

10. Type **A**. Verify the number 2 under the # symbol. Press <enter> to move the cursor to the next column. Type **A** to access the *Tops* menu. Type **LPI** and press <enter>. The cursor should now be in the And column.

11. Type **A** and **/LCS**. Move the cursor to the next column. Type **A** and **RHR**. Press <enter> to move the cursor to the Is column. Type **A** and **Syst2**.

12. Type **E** to exit the *Link Editor* screen. Observe the message "**Save Rules, then Exit (Y/N)**" displayed c the screen. Type **Y** and press **<enter>** to return to the *Link Event Tree* screen.

13. Type **G** and place the cursor on the LOCA file name in the window. Then press **<enter>** to access the *Sequence Generation Information* window. Observe that the cursor is in the *Report Name* screen.

14. Delete the report name LOCA.LST and type **PRN**. Press **<enter>** to obtain a printout of the sequences and return to the *Link Event Tree* screen. Compare this printout to the printout obtained in Step 6. Verify that the changes were made to the sequences.

15. Type **L** and place the cursor on the LOCA file name. Press **<enter>** to access the *Link Editor* screen. At the first sequence, place the cursor on the /LPI system name and type **I**. Verify that a new space was inserted on the screen. Type **A** to access the *Tops* menu, type **VIJ**, and press **<enter>**. Verify that system name VIJ was added on the screen.

16. Place the cursor on the system name VIJ and type **D** in order to delete. Verify that the system name was removed from the screen. Type **R** in order to restore the system name VIJ. Verify that the system name VIJ is restored on the screen.

17. Place the cursor on the system name LPI in sequence 2 and type **C**. The message "**Copy Rule 2 after Rule 0 (0=First, 9999= Last)**" will be displayed at the top on the screen. Place the cursor in the After Rule space, type **2**, and press **<enter>**. Verify that the sequence was copied and that the sequence is now sequence 3.

18. Exit the *Link editor* screen and return to the *Link Event Tree* screen. Exit the *Link Event Tree* screen and Return to the *IRRAS* menu.

19. **IF:**   Steps 1 through 18 were performed successfully

    THEN: GO TO Step 20.

    ELSE: Record which step(s) was unsuccessful. Generate an anomaly report and submit it to the SAPHIRE 4.0 V&V committee.

20. End of Section 11.

F-6. PROCEDURE 3—FEP NONVITAL FEATURES

F-6.1 Introduction

FEP is a program developed to provide common user access to the suite of graphical tools developed for performing risk analysis. These tools include the fault tree, event tree, and piping & instrumentation diagram (P&ID) graphical editors. The fault tree and event tree editors are available through IRRAS, and allow the analyst to construct and modify graphical fault trees and event trees, respectively. The P&ID editor is accessible only through FEP and allows the user to construct and modify graphical plant drawings. These plant drawings can be used to document the modeling used in a PRA. FEP allows the analyst to generate a graphical display, and avoid using the more complex and time consuming IRRAS program.

This procedure tests the nonvital features associated with the P&ID editor of FEP. The nonvital features of the fault tree and event tree graphical editors are not included because they are tested in Procedures 1 and 2. The nonvital P&ID features include extracting P&ID graphics, plotting P&IDs, and generating symbols plus the P&ID editor. Table F-4 identifies the nonvital features of the P&ID editor, the success criteria, and the specific options that are tested in each section. The steps in each section of this procedure should be followed. If any step is not completed successfully, an anomaly report should be generated and submitted to the SAPHIRE 4.0 V&V committee.

Table F-4. FEP test procedure specifications.

Sections	Feature group	Features	Success criteria	Options tested
1	P&ID Editor	Edit	To make changes to the P&ID	Symbols, Line Ortho Y, Ortho N, Flip ←→, Flip ↑↓, Rotate, Scale, Move, Copy, Delete, and Adjust
		Utility	To specify program defaults	Rotate, Skip
		Text	To add and manipulate text	Write
		Prnt	To send the current file to a Laser printer	Laser
		File	To load and save drawings and perform file manipulations	Save, File? and New
2	P&ID Editor	Edit	To make changes to the P&ID	Symbols, Line Attribute, Name
		Text	To add and manipulate text	Write
		File	To load and save drawings and perform file manipulations	Save, and File?
	Utilities	HP Plotter	To send P&ID to the laser printer	Hpgl File Raster File
3	P&ID Editor	File	To load and save drawings and perform file manipulations	List
		Text	To add and manipulate text	Font, Write Move, Copy, Edit, Text, Just, Color, Size, Erase
		View	To change the position and size of the displayed diagram	Zout, Zres
4	P&ID Editor	File	To load and save drawings and perform file manipulations	Load, +Tran, Save, Xfer→ ←Xfer
		Edit	To make changes to the P&ID	Attributes, Fill Color, Name Color, Line Color, Name Size, Name Font
5	P&ID Editor	View	To change the position and size of the displayed diagram	Page ↑,↓ Page ←,→ Scroll, Zout, Zin, Zprev, Zres

Table F-4. FEP test procedure specifications.

Sections	Feature group	Features	Success criteria	Options tested
		Utility	To specify program defaults	Grid Off, Grid On, Show Name, No Name, Skip, Snap On, Snap Off, and Scale
6	Utility	Symbol Generator	To make a change to an existing symbol file or create a new symbol file	Symbol Generator
	P&ID Editor	FILE	To load and save drawings and perform file manipulations	SName
		Edit	To make changes to the P&ID	Copy, Paste
7		Text	To set a default color for the text for a diagram	Color Bar
		Cntr/Left/Right	Change the text justification	Center, Right, Left
		Fill	To select color for symbols	Color Bar
		Line	To select a color for lines to used in a diagram	Color Bar
		On/Off	To toggle to fill or not to fill symbols when placed on the drawings	On, Off
8	Utility	Extract Graphics/P&IDS	To extract or delete P&IDS	Extract P&IDs, and Delete Extracted P&IDs

F-6.2 Section 1

1. Enter the PRADATA.B1 directory, type **FEP** and press <enter> to access the *Fault Tree, Event Tree, and Piping & Instrumentation Diagram Editors* menu.

2. Place the cursor on the "P&ID Editor" option and press <enter> to access the *P&ID Editor Graphics* menu. Place the cursor on the "EDIT" option and press the **left mouse button** to access the *EDIT* menu. Then place the cursor on the "SYMBLS" option and press the **left mouse button** to access the *Symbol* menu.

3. Place the cursor on the pump symbol and press the **left mouse button**. Drag the symbol and press the **left mouse button** to position the symbol on the screen. Press the **right mouse button** to cancel the "SYMBLS" option.

4. Repeat Step 3 for the remaining symbols. Verify that each symbol is successfully placed on the screen. Press the **right mouse button** to cancel the *Symbol* menu.

5. From the *EDIT* menu, place the cursor on the "LINE" option and press the **left mouse button** to access the *LINE* menu.

6. From the *LINE* menu, place the cursor on the first line located in the first row, and press the left mouse button. The message "**Enter points in order of direction of flow**" will be displayed at the bottom of the screen.

7. Press the **left mouse button**. Drag the cursor to a desired location and press the **left mouse button**. Press the **right mouse button** to terminate the line. Verify that the line appears on the screen. Press the **right mouse button** again to cancel the "LINE" option.

8. Repeat steps 6 and 7 for the remaining lines.

9. From the *LINE* menu, place the cursor on the "Ortho Y" option and press the **left mouse button**. Verify that the orthogonal line function toggles to the "Ortho N" option.

10. Place the cursor on the line located in the fifth row of the *LINE* menu and press the **left mouse button**. The message "**Enter points in order of direction of flow**" will be displayed in the bottom left corner of the screen.

11. Press the **left mouse button**. Drag the cursor to a desired location and press the **left mouse button** again to display a line on the screen. Drag the cursor to a new location, at a different angle, and press the **left mouse button** again. Verify that a diagonal line has been drawn. Press the **right mouse button** twice to cancel the "Ortho N" option.

12. From the *LINE* menu, place the cursor on the "Ortho N" option and press the **left mouse button**. Verify that the orthogonal line function toggles to the "Ortho Y" option.

13. Place the cursor on the line located on the fifth row of the *LINE* menu and press the **left mouse button**. The message "**Enter points in order of direction of flow**" will be displayed in the bottom left corner of the screen.

14. Press the **left mouse button**. Drag the cursor to a desired location and press the **left mouse button** to display a line on the screen. Drag the cursor to a new location, at a different angle, and press the **left mouse button** again. Verify that a line at a right angle has been drawn. Press the **right mouse button** twice to cancel the "Ortho Y" option. Press the **right mouse button** again to cancel the *LINE* menu.

15. Return to the *EDIT* menu. Place the cursor on the "←—→Flip" option and press the left mouse button. The message "**Box Symbols to flipped**" will be displayed in the bottom left corner of the screen. Place the cursor near the butterfly valve symbol and press the **left mouse button**. Drag the cursor until the box surrounds one of the symbols and press the **left mouse button** again. Verify that the symbol was inverted horizontally. Press the **right mouse button** to cancel the "←—→Flip" option.

16. Place the cursor on the "↑↓Flip" option and press the **left mouse button**. The message "**Box Symbols to flipped**" will be displayed in the bottom left corner of the screen. Place the cursor near a control valve and press the **left mouse button**. Drag the cursor until the box surrounds the symbol and press the **left mouse button** again. Verify that the symbol has been inverted vertically. Press the **right mouse button** to cancel the "↑↓Flip" option.

17. Return to the *Primary P&ID Graphics* menu. Place the cursor on the "UTIL" option and press the left mouse button to access the *UTIL* menu. Place the cursor on the "Rotate" option and press the left mouse button. The message "**Group rotations will be done at center of each symbol/line - change? >**" will be displayed in the bottom left corner of the screen. Type N.

18. Return to the *EDIT* menu. Place the cursor on the "Rotate" option and press the **left mouse button**. The message "**Pick Symbol to rotate**" will be displayed in the bottom left corner of the screen.

19. Place the cursor near a valve symbol and press the **left mouse button**. Drag the cursor until the box surrounds a valve symbol and press the **left mouse button** to access a window. Place the cursor on the "Angle =" option and press the **left mouse button**. The message **"Enter the rotation angle (0-360)"** will be displayed at the bottom on the screen.

20. Type **90** and press the **left mouse button**. Verify that the window displays the numerical value of 90 degrees and that the other window displays a cursor indicating the position of the angle. Place the cursor on the "OK" option and press the **left mouse button**. Verify on the screen that the position of the valve symbol changed to the desired angle.

21. Place the cursor near the same symbol and press the **left mouse button**. Drag the cursor until the box surrounds the symbol. Press the **left mouse button** again to access a window. In the left window, notice the symbol that looks like an analog clock without numbers. Place the cursor in the window and press the **left mouse button**. Verify that the angle indicator in the window changes position to represent the new angle.

22. Place the cursor on the "OK" option and press the **left mouse button**. Verify that the symbol rotates to the specified angle. Press the **right mouse button** to cancel the "Rotate" option.

23. Return to the *UTIL* menu. Place the cursor on the "rotate" option and press the **left mouse button**. The message **"Group rotations will be done at center of each symbol/line - change? >"** will be displayed in the bottom left corner of the screen. Type **Y**.

24. Return to the *EDIT* menu. Place the cursor on the "Rotate" option and press the **left mouse button**. The message **"Pick Symbol to rotate"** will be displayed at the bottom of the screen.

25. Place the cursor near a valve symbol and press the **left mouse button**. Drag the cursor until the box surrounds the valve symbol and press the **left mouse button**. The message **"Pick center of rotation (point to remain unchanged)"** will be displayed in the bottom left corner of the screen.

26. Move the cursor to a location on the screen. Press the **left mouse button** to access the window. Record the angle value displayed on the screen. _____

27. Place the cursor on the "OK" option and press the **left mouse button**. Verify on the screen that the position of the valve symbol changed to the recorded angle in Step 26. Also, verify that the symbol changed relative to the cursor position according to the angle recorded in Step 26.

28. From the *EDIT* menu, place the cursor on the "Scale" option and press the **left mouse button**. The message **"Box Symbols to scale"** will be displayed at the bottom of the screen.

29. Place the cursor near one of the symbols and press the **left mouse button**. Drag the cursor until the box surrounds the symbol and press the **left mouse button**. The message **"Select edge/corner to expand-corners give even scale"** will be displayed in the bottom left corner of the screen.

30. Place the cursor on the corner of the box and press the **left mouse button**. Drag the mouse until the box expands. Press the **left mouse button** again. Verify that the symbol expanded to the size of the box.

31. Place the cursor on the corner of the box and press **the left mouse button**. Drag the mouse until the box shrinks to a smaller size. Press the **left mouse button**. Verify that the symbol shrinks to the size of the box. Press the **right mouse button** twice to cancel the "Scale" option.

32. Place the cursor on the "Move" option and press the **left mouse button**. The message **"Pick region to be moved- press CANCEL to quit"** will be displayed at the bottom of the screen. Place the cursor

near the symbol and press the **left mouse button**. Drag the cursor until the box surrounds the symbol and press the **left mouse button** again.

33. The message **"Pick reference point - press CANCEL to quit"** will be displayed in the bottom left corner of the screen. Pick the reference point by placing the cursor on the symbol and pressing the **left mouse button**. Move the box to a new location and press the **left mouse button** again. Verify that the selected symbol was moved to the desired location. Press the **right mouse button** to cancel the "Move" option.

34. Place the cursor on the "Copy" option and press the **left mouse button**. The message **"Pick region to be copied - cancel to quit"** will be displayed at the bottom of the screen. Place the cursor near the symbol and press the **left mouse button**. Drag the cursor until the box surrounds the symbol and press the **left mouse button** again.

35. The message **"Pick reference point-cancel to quit"** will be displayed in the bottom left corner of the screen. Pick the reference point by placing the cursor on the symbol and pressing the **left mouse button**. Move the box to a new location and press the **left mouse button** again. Verify that the selected symbol was copied to the desired location. Press the **right mouse button** to cancel the "Copy" option.

36. Return to the *P&ID Editor Graphics* menu. Place the cursor on the "PRNT" option and press the **left mouse button** to access the *PRINT* menu. Place the cursor on the "Laser" option and press the **left mouse button**. Verify that the printout is identical to the display shown on the screen.

37. From the *Primary P&ID Graphics* menu, place the cursor on the "TEXT" option and press the **left mouse button** to access the *TEXT* menu. Place the cursor on the "Write" option and press the **left mouse button**. The message **"Pick text placement location"** will be displayed in the bottom left corner of the screen.

38. Press the **left mouse button** to access the text window. Type **text of your choice** and press the <ESC> key. Verify that the typed text appears on the screen.

39. Repeat Step 38 several times, but place the text at different locations. Press the **right mouse button** to cancel the "Write" option. Press the **right mouse button** again to cancel the *TEXT* menu.

40. From the *EDIT* menu, place the cursor on the "Delete" option and press the **left mouse button**. The message **"Pick region to be deleted"** will be displayed at the bottom of the screen.

41. Place the cursor near the figure on the screen and press the **left mouse button**. Drag the cursor until the box surrounds a combination of symbols, lines, and text. Press the **left mouse button** to access the *Delete* menu. The message **"Select type of deletion-CANCEL to terminate"** will be displayed in the bottom left corner of the screen.

42. Place the cursor on the "Symbols" option and press the **left mouse button**. Verify that only the symbols were deleted.

43. Repeat Step 41. Place the cursor on the "Text" option and press the **left mouse button**. Verify that only the text was deleted.

44. Repeat Step 41. Place the cursor on the "Lines" option and press the **left mouse button**. Verify that only the line or lines were deleted.

45. Repeat Step 41. Place the cursor on the "All" option and press the **left mouse button**. Verify that the symbols, text, and lines were all deleted. Press the **right mouse button** to cancel the delete "symbols" option.

· NUREG/CR-6145

46. From the *EDIT* menu, place the cursor on the "SYMBLS" option and press the **left mouse button** to access the *Symbol* menu.

47. Place the cursor on the valve symbol and press the **left mouse button**. Move the valve symbol to a desired position on the screen and press the **left mouse button** again.

48. Move the valve symbol to a position near the first valve symbol already displayed on the screen and press the **left mouse button**.

49. Repeat Step 48 to place a third and fourth valve symbol on the screen. Press the **right mouse button** to cancel the "Symbols" option. Press the **right mouse button** again to cancel the *Symbol* menu.

50. Place the cursor on the "UTIL" option and press the **left mouse button** to access the *UTIL* menu. Place the cursor on the "Skip" option and press the **left mouse button**. The message **"Enter grid snap value (default= 1.0, range = [0,120])"** will be displayed in the bottom left corner of the screen. Type **3** and press **<enter>**.

51. Return to the *EDIT* menu. Place the cursor on the "Adjust" option and press the **left mouse button**. The message **"Box Symbols to be adjusted"** will be displayed in the bottom left corner of the screen.

52. Place the cursor near the symbols and press the **left mouse button**. Drag the cursor until a box surrounds the four valve symbols and press the **left mouse button** again. Verify that the valve symbols are in alignment. Press the **right mouse button** to cancel the "Adjust" option. Press the **right mouse button** again to cancel the *EDIT* menu.

53. Return to the *UTIL* menu. Place the cursor on the "Skip" option and press the **left mouse button**. The message **"Enter grid snap value (default= 1.0, range = [0,120])"** will be displayed in the bottom left corner of the screen. Type **1** and press **<enter>**. Press the **right mouse button** to cancel the *UTIL* menu.

54. From the *Primary P&ID Graphics* menu, place the cursor on the "FILE" option and press the **left mouse button** to access the *FILE* menu. Place the cursor on the "Save" option and press the **left mouse button**. The message **"Enter file name or CR for file [NONAME.PID]"** will be displayed in the bottom left corner of the screen.

55. Type the file name **SYMBOL** and press **<enter>**. Verify that the message **"File Saved"** appears in the bottom left corner on the screen.

56. From the *FILE* menu, place the cursor on the "File?" option and press the **left mouse button**. Verify that the file name SYMBOL.PID is displayed in the bottom left corner of the screen.

57. Place the cursor on the "New" option and press the **left mouse button**. Verify that a new screen appears on the screen.

58. IF: Steps 1 through 57 were performed successfully

    THEN: Go to Step 59.

    ELSE: Record which step(s) was unsuccessful. Generate an anomaly report and submit it to the SAPHIRE 4.0 V&V committee.

59. End of Section 1.

F-6.3 Section 2

1. The Containment Spray P&ID, shown in Figure F-4 will now be created. From the *Primary P&ID Graphic* menu, place the cursor on the "EDIT" option and press the **left mouse button** to access the *EDIT* menu.

2. Place the cursor on the "SYMBLS" option and press the **left mouse button** to access the *Symbol* menu. Locate and place each symbol at the appropriate location on the screen. When completed, press the **right mouse button**. Press the **right mouse button** again to cancel the *Symbol* menu.

3. From the *EDIT* menu, place the cursor on the "Line" option and press the **left mouse button** to access the *LINE* menu. Place the cursor on the line located in the fifth row and press the **left mouse button**. The message **"Enter points in order of direction of flow"** will be displayed at the bottom of the screen.

4. Connect the line to each symbol until the P&ID is completed. Press the **right mouse button** to cancel the "Line" option. Press the **right mouse button** to cancel the *LINE* menu.

5. Return to the *P&ID Editor Graphics* menu. Place the cursor on the "0.50" option and press the **left mouse button**. The message **"Enter new text size>"** will be displayed in the bottom left corner of the screen. Type **1.5** and press **<enter>**. Verify the change on the *Primary P&ID Graphics* menu.

6. Place the cursor on the "TEXT" option and press the **left mouse button** to access the *TEXT* menu.

7. Place the cursor on the "Write" option and press the **left mouse button**. The message **"Pick the text placement location"** will be displayed on the bottom left corner of the screen.

8. Place the cursor under the P&ID diagram and press the **left mouse button** to access the text window. Type **Figure F-1. Containment spray system.** and then press the **<ESC>** key. Verify that the text appears on the screen. Press the **right mouse button** to cancel the "Write" option.

9. From the *Primary P&ID Graphics* menu, place the cursor on the "EDIT" option and press the **left mouse button** to access the *EDIT* menu.

10. Place the cursor on the "ATTRIB" option and press the **left mouse button** to access the *ATTRIBUTES* menu. Place the cursor on the "Name" option and press the **left mouse button**. The message **"Box the symbols to be changed"** will be displayed in the bottom left corner of the screen.

11. Place the cursor near the pump symbol and press the **left mouse button**. Drag the cursor until the box surrounds the symbol and press the **left mouse button**. The message **"Enter new symbol name"** will be displayed in the bottom left corner of the screen. Type **MDP-1** and press **<enter>**. Verify that the component name is displayed on the screen.

12. Continue naming the remaining components until the P&ID is the same as the P&ID shown in Figure F-4. When completed, press the **right mouse button** to cancel the "Name" option.

13. Return to the *Primary P&ID Graphics* menu. Place the cursor on the "FILE" option and press the **left mouse button** to access the *FILE* menu. Place the cursor on the "Save" option and press the **left mouse button**. The message **"Enter file name or CR for file [NONAME.PID]"** will be displayed in the bottom left corner of the screen.

14. Type **Cont** for the file name and press the **left mouse button**. From the *FILE* menu, place the cursor on the "File?" option. Verify that the file name CONT.PID is displayed in the bottom left corner of the screen.
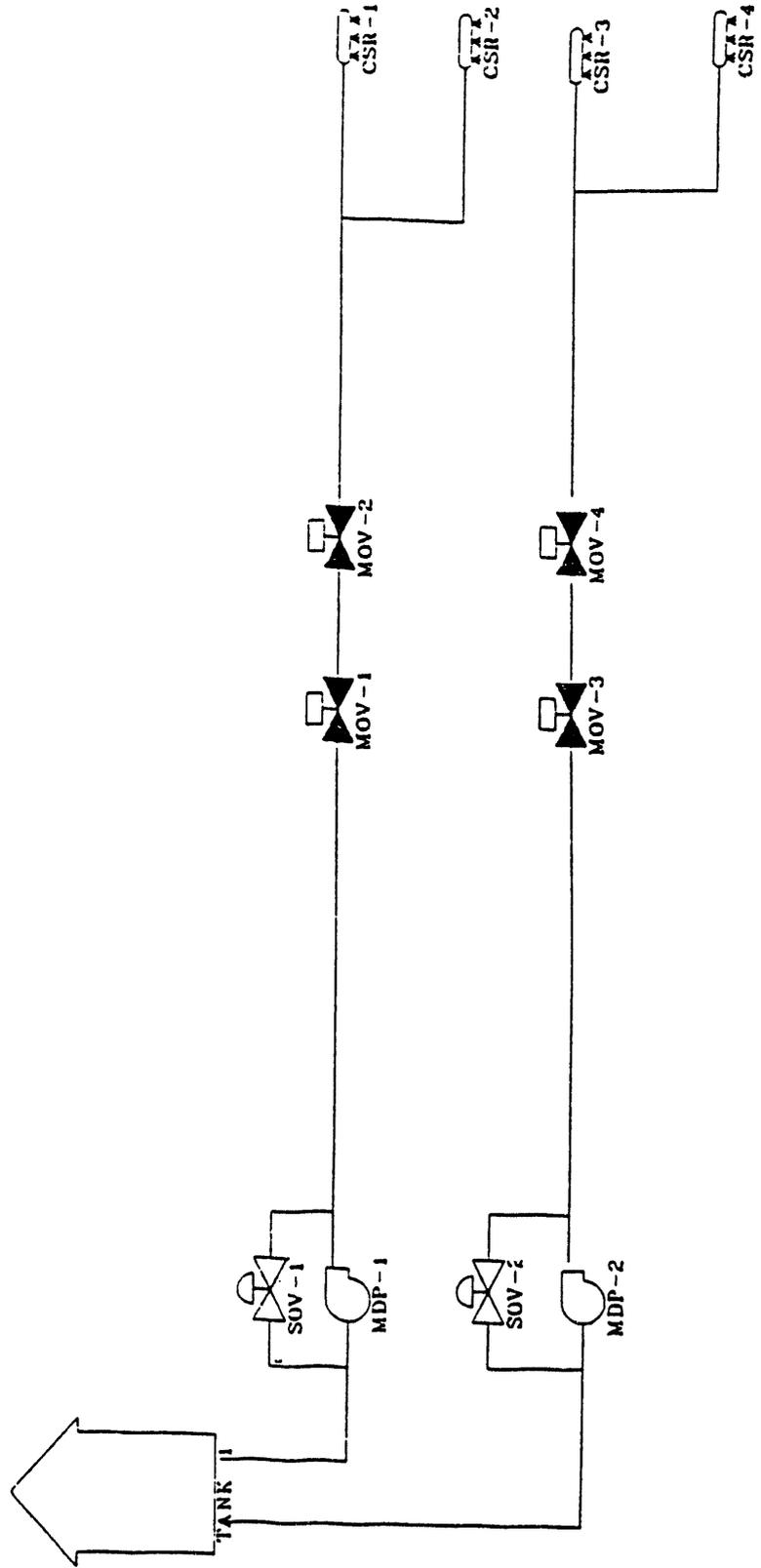
Figure F-4. Containment spray system.

15. Place the cursor on the "EXIT" option and press the **left mouse button** to access the *FEP* main menu. Place the cursor on the "UTILITY" option and press <enter> to access the *Utility* menu.

16. Place the cursor on the *HP Plotter* menu and press <enter> to access the *Plotter* menu. Place the cursor on the "P&ID Drawings" option and press <enter> to access the *Plot Graphics* screen.

17. Type **H**, place the cursor on the Cont file name, and press <enter> to generate an HPGL file. Type **R**, place the cursor on the CONT file name, and press <enter> to generate an RAS file.

18. Exit the FEP program and go the PRADATA.B1\XXXXX subdirectory. Note: *The subdirectory XXXXX is the family name.* Type ..\PLOTHPG*.HPG and press <enter> to obtain a printout of the HPGL file. Note: *PLOTHPG will only work on an HP LaserJet III or IIIP.*

19. Type ..\PLOTRAS *.RAS and press <enter> to obtain a printout of the RAS files. Note: *The PLOTRAS will work on any laser printer.* Verify that the P&ID printout corresponds to the P&ID in Figure F-4.

20. **IF**: Steps 1 through 19 were performed successfully

    THEN: Go to Step 21.

    ELSE: Record which step(s) was unsuccessful. Generate an anomaly report and submit it to the SAPHIRE 4.0 V&V committee.

21. End of Section 2.

F-6.4 Section 3

1. Return to the PRADATA.B1 directory, type **FEP**, and press <enter> to access the *FEP* main menu. Select the "P&ID Editor" option and press <enter> to access the *Primary P&ID Graphics* menu.

2. Place the cursor on the "FILE" option and press the **left mouse button** to access the *FILE* menu. Place the cursor on the "List" option and press the **left mouse button**. Verify that the file name CONT is displayed on the screen. Then place the cursor on CONT and press the **left mouse button**. Verify that the P&ID is displayed on the screen.

3. From the *Primary P&ID Graphics* menu, place the cursor on the "0.50" option and press the **left mouse button**. The message "**Enter new text size**" will be displayed in the bottom left corner of the screen. Type **2.0** and press <enter>. Verify the change on the *Primary P&ID Graphics* menu.

4. Place the cursor on the "TEXT" option and press the **left mouse button** to access the *TEXT* menu.

5. Place the cursor on the "FONT" option and press the **left mouse button** to access the *FONT* menu. Place the cursor on one of the Font options and press the **left mouse button**.

6. From the *Text* menu place the cursor on the "Write" option and press the **left mouse button**. The message "**Pick the text placement location**" will be displayed in the bottom left corner of the screen.

7. Place the cursor at a blank location on the screen and press the **left mouse button** to access the text window. Type the text **of your choice** and press the <ESC> key. Verify on the screen that the text matches the Font that was chosen. Press the **right mouse button** to terminate the "Write" option.

8. Repeat Steps 6 and 7 for several fonts. Verify that the text changed to the desired fonts.

9. Place the cursor on the "Move" option and press the **left mouse button**. The message **"Pick region to be moved - press Cancel to quit."** will be displayed in the bottom left corner of the screen.

10. Place the cursor near the text to be moved and press the **left mouse button**. Drag the cursor until the box surrounds the text and press the **left mouse button**. The message **"Pick reference point - press CANCEL to quit"** will be displayed in the bottom left corner of the screen.

11. Pick the reference point by positioning the cursor on the text. Then press the **left mouse button** and position the box at a new location where text can be moved. Press the left mouse button again. Verify that the text has moved to a new location on the screen. Press the **right mouse button** to terminate the "Move" option.

12. Place the cursor on the "Copy" option and press the **left mouse button**. The message **"Pick region to be copied- press CANCEL to quit"** will be displayed on the bottom left corner on the screen.

13. Place the cursor near the text to be copied and press the **left mouse button**. Drag the cursor until the box surrounds the text and press the **left mouse button**. The message **"Pick reference point - press cancel to quit"** will be displayed in the bottom left corner of the screen.

14. Pick a reference point by placing the cursor on the text and pressing the **left mouse button**. Place the box at a new location where the text can be copied and press the **left mouse button**. Verify that the text was copied on the screen. Press the **right mouse button** twice to terminate the "Copy" option.

15. Place the cursor on the "EDIT" option and press the **left mouse button** to access the *EDIT* menu.

16. Place the cursor on the "Text" option and press the **left mouse button**. The message **"Pick the text to be changed"** will be displayed at the bottom of the screen.

17. Place the cursor near the text that was copied and press the **left mouse button**. Drag the cursor until the box surrounds the text and press the **left mouse button** to access a window with the written text. Make revisions to the text and press the <ESC> key. Verify that the changes were made to the text displayed on the screen. Press the **right mouse button** to cancel the "Text" option.

18. Place the cursor on the "Colr" option and press the **left mouse button** to edit the text color. The message **"Pick a new color from the color bar"** will be displayed on the bottom left corner of the screen. Place the cursor on a color located in the *Primary P&ID Graphics* menu and press the **left mouse button**. The message **"Pick the text to be changed"** will be displayed in the bottom left corner of the screen.

19. Place the cursor near the text and press the **left mouse button**. Drag the cursor until the box surrounds the text and press the **left mouse button**. Verify that the text changed to the color that was chosen from the color bar. Press the **right mouse button** to cancel the "Colr" option.

20. Repeat Steps 18 and 19 for several colors.

21. Place the cursor on the "Size" option and press the **left mouse button**. The message **"Enter new Text size >"** will be displayed in the bottom left corner of the screen. Type 1.0 and press <enter>. The message **"Pick the text to be changed"** will be displayed in the bottom left corner of the screen.

22. Place the cursor near some text and press the **left mouse button**. Drag the cursor until the box surrounds the text and press the **left mouse button** again. The message **"Pick the text to be changed - Press cancel to quit"** will be displayed in the bottom left corner of the screen. Verify on the screen that the text size changed. Press the **right mouse button** to cancel the "Size" option.

23. Return to the *Primary P&ID Graphics* menu. Place the cursor on the "VIEW" option and press the **left mouse button** to access the *VIEW* menu.

24. Place the cursor on the "Zout" option and press the **left mouse** button. Verify that the figure is reduced approximately 50%. Press the **right mouse button** to cancel the *VIEW* menu.

25. From the *Primary P&ID Graphics* menu, place the cursor on the "TEXT" option and press the **left mouse button** to access the *TEXT* menu. Place the cursor on the "EDIT" option and press the **left mouse button** to access the *EDIT* menu.

26. Place the cursor on the "Just" option and press the **left mouse button**. The message **"Enter text justification - ('L'=Left, 'R'=Right, 'C'=Center) >"** will be displayed at the bottom of the screen.

27. Type **C** and the message **"Pick the text to be changed"** will be displayed in the bottom left corner of the screen. Place the cursor near the text and press the **left mouse button**. Drag the cursor until the box surrounds the text and press the **left mouse button**. Verify that the text displayed on the screen changed to the selected justification. Press the **right mouse button** to cancel the "Just" option.

28. Repeat steps 26 and 27, but type **R**.

29. Repeat steps 26 and 27, but type **L**.

30. Return to the *Primary P&ID Graphics* menu. Place the cursor on the "VIEW" option and press the **left mouse button** to access the *VIEW* menu.

31. Place the cursor on the "Zres" option and press the **left mouse button**. Verify that the display returns to the original display size. Press the **right mouse button** to cancel the *VIEW* menu.

32. From the *Primary P&ID Graphics* menu, place the cursor on the "TEXT" option and press the **left mouse button** to access the *TEXT* menu. Place the cursor on the "EDIT" option and press the left **mouse button** to access the *EDIT* menu.

33. Place the cursor on the "FONT" option, and press the **left mouse button** to access the *FONT* menu. Place the cursor at one of the desired fonts and press the **left mouse button**. The message **"Pick the text to be changed."** will be displayed in the bottom left corner of the screen.

34. Place the cursor near the text to be modified and press the **left mouse button**. Drag the cursor until the box surrounds the desired text and press the **left mouse button** again. Verify that the text changed to the desired font on the screen. Press the **right mouse button** to terminate the "FONT" option. Press the **right mouse button** to cancel the *EDIT* menu.

35. Return to the *Text* menu. Place the cursor on the "Erase" option and press the **left mouse button**. The message **"Pick region to be deleted"** will be displayed on the bottom left corner of the screen.

36. Place the cursor near some text and press the **left mouse button**. Drag the cursor until the box surrounds the text and press the **left mouse button**. Verify that text on the screen was deleted.

37. Place the cursor on the "EXIT" option and press the **left mouse button**. The message **"Current drawing not saved! Exit anyway? <Y/N>"** will be displayed in the bottom left corner of the screen. Type **Y** to return to the *FEP* main menu.

38. **IF:**   Steps 1 through 37 were performed successfully

    THEN: Go to Step 39

ELSE: Record which step(s) was unsuccessful. Generate an anomaly report and submit it to the SAPHIRE 4.0 V&V committee.

39.   End of Section 3.

F-6.5 Section 4

1.   Place the cursor on the "P&ID Editor" option and press <enter> to access the *Primary P&ID Graphics* menu.

2.   Place the cursor on the "FILE" option and press the **left mouse button** to access the *FILE* menu. Place the cursor on the "Load" option and press the **left mouse button**. The message **"Enter file name >"** will be displayed at the bottom of the screen. Type **Cont** and press <enter>. Verify that the P&ID diagram is displayed on the screen.

3.   From the *Primary P&ID Graphics* menu, place the cursor on the "FILE" option and press the **left mouse button** to access the *FILE* menu. Place the cursor on the "+Tran" option and press the **left mouse button**. Verify that a triangle symbol appears on the screen.

4.   Place the triangle at a convenient location on the screen and press the **left mouse button**. The message **"Enter Transfer file name >"** will be displayed in the bottom left corner of the screen. Type the file name **symbol** and press <enter>. Verify the triangle with the SYMBOL name on the screen. Press the **right mouse button** to cancel the "+Tran" option.

5.   Place the cursor on the "Save" option. The message **"Enter file name or CR for file. [CONT.PID] >"** will be displayed in the bottom left corner of the screen. Press the **left mouse button** again and the message **"File already exists. Replace file? <Y?N>"** will be displayed in the bottom left corner of the screen. Type **Y** and press <enter>.

6.   From the *FILE* menu, place the cursor on the "Xfer→" option and press the **left mouse button**. The message **"Pick transfer symbol to transfer"** will be displayed in the bottom left corner of the screen. Place the cursor on the triangle symbol and press the **left mouse button**. Verify on the screen that the symbol transferred.

7.   From the *Primary P&ID Graphics* menu, place the cursor on the "FILE" option and press the **left mouse button** to access the *FILE* menu. Place the cursor on the "←Xfer" option and press the **left mouse button**. Verify that a list of file names are displayed on the screen. The message **"Pick the file to transfer to"** will be displayed in the bottom left corner of the screen.

8.   Place the cursor on the file named CONT and press the **left mouse button**. Verify that the P&ID is displayed on the screen.

9.   Place the cursor on the "EDIT" option and press the **left mouse button** to access the *EDIT* menu. Place the cursor on the "ATTRIB" option and press the **left mouse button** to access the *ATTRIBUTES* menu.

10.   Place the cursor on the "Fill col" option and press the **left mouse button**. The message **"Pick a new color from the color bar"** will be displayed in the bottom left corner of the screen.

11.   Place the cursor on a color from the *Color Bar* menu and press the **left mouse button**. The message **"Box the symbols to be changed"** will be displayed in the bottom left corner of the screen.

12.   Place the cursor near one of the symbols to be modified and press the **left mouse button**. Drag the cursor until the box surrounds the symbol and press the **left mouse button** again. Verify that the symbol changed to the selected color. Press the **right mouse button** to cancel the "Fill col" option.

13. Repeat steps 11 and 12 for the remaining colors.

14. From the *ATTRIBUTES* menu, place the cursor on the "Name Col" option and press the **left mouse button**. The message **"Pick a new color from the color bar"** will be displayed at the bottom on the screen.

15. Place the cursor on a color from the *Color Bar* menu and press the **left mouse button**. The message **"Box the symbols to be changed"** will be displayed in the bottom left corner of the screen.

16. Place the cursor near one of the symbols to be modified and press the **left mouse button**. Drag the cursor until the box surrounds the symbol and press the **left mouse button** again. Verify that the symbol name changed to the selected color. Press the **right mouse button** to cancel the "Name col" option.

17. Repeat steps 15 and 16 for the remaining colors.

18. From the *ATTRIBUTES* menu, place the cursor on the "Line col" option and press the **left mouse button**. The message **"Pick a new color from the color bar"** will be displayed at the bottom of the screen.

19. Place the cursor on a color located on the *Color Bar* menu and press the **left mouse button**. The message **"Place a box around the lines to be changed - Press CANCEL to quit"** will be displayed in the bottom left corner of the screen.

20. Place the cursor near the line to be modified and press the **left mouse button**. Drag the cursor until the box surrounds the line and press the **left mouse button** again. Verify that the line changes to the selected color. Press the **right mouse button** to cancel the "Line col" option.

21. Repeat steps 19 and 20 for the remaining colors.

22. From the *ATTRIBUTES* menu, place the cursor on the "Name size" option and press the **left mouse button**. The message **"Enter new text size"** will be displayed in the bottom left corner of the screen.

23. Type a symbol *name size of your choice* and press <enter>. The message **"Box the symbols to be changed"** will be displayed in the bottom left corner of the screen.

24. Place the cursor near one of the symbols to be modified and press the **left mouse button**. Drag the cursor until the box surrounds the symbol and press the **left mouse button** again. Verify on the screen that the symbol name size was modified. Press the **right mouse button** to cancel the "Name Size" option.

25. From the *ATTRIBUTES* menu, place the cursor on the "Name FONT" option and press the **left mouse button** to access the *FONT* menu. Place the cursor on one of the fonts and press the **left mouse button**. The message **"Box the symbols to be changed"** will be displayed in the bottom left corner of the screen.

26. Place the cursor near one of the symbols to be modified and press the **left mouse button**. Drag the cursor until the box surrounds the symbol and press the **left mouse button** again. Verify on the screen that the symbol name changed to the selected font. Press the **right mouse button** to cancel the "Name Font" option.

27. Place the cursor on the "EXIT" option and press the **left mouse button**. The message **"Current drawing not saved! Exit anyway? <Y?N>"** will be displayed on the bottom left corner of the screen. Type **Y** to return to the *FEP* main menu.

28. <u>IF</u>:   Steps 1 through 27 were performed successfully

       <u>THEN</u>: Go to Step 29.

       <u>ELSE</u>: Record which step(s) was unsuccessful. Generate an anomaly report and submit it to the SAPHIRE 4.0 V&V committee.

29.   End of Section 4.

F-6.6 Section 5

1. Place the cursor on the "P&ID Editor" option and press <enter> to return to the *Primary P&ID Graphics* menu.

2. Place the cursor on the "FILE" option and press the **left mouse button** to access the *FILE* menu. Place the cursor on the "List" option and press the **left mouse button**. The message **"Pick the file to load"** will be displayed in the bottom left corner of the screen. The file names CONT and SYMBOL will be displayed on the screen. Place the cursor on the CONT file name and press the **left mouse button**. The P&ID will be displayed on the screen.

3. Place the cursor on the "VIEW" option and press the **left mouse button** to access the *VIEW* menu.

4. Place the cursor on the "Page ↑" option and press the **left mouse button**. This will shift the P&ID one page up. Verify that the screen is blank.

5. Place the cursor on the "Page ↓" and press the **left mouse button**. This will shift the P&ID one page down a page. Verify that the diagram returns on the screen.

6. Place the cursor on the "Page ←" option and press the **left mouse button**. This will shift the P&ID one page to the left. Verify that the screen is blank.

7. Place the cursor on the "Page →" option and press the **left mouse button**. This will shift the P&ID one page to the right . Verify that the diagram returns on the screen.

8. Place the cursor on the "Scroll" option and press the **left mouse button**. Verify that a white outline box appears and a cursor is in the center of the box. Place the cursor at a desired location and press the **left mouse button**. Verify that the figure moves relative to the cursor position.

9. Place the cursor on the "Zout" option and press the **left mouse button**. Verify that the figure is reduced approximately 50%. Place the cursor on the "Zres" option and press the **left mouse button**. Verify that the display returns to its original size.

10. Place the cursor on the "Zin" option and press the **left mouse button**. The message **"Pick first corner"** will be displayed in the bottom left corner of the screen.

11. Place the cursor near a symbol and press the **left mouse button**. Drag the cursor until the box surrounds the symbol and press the **left mouse button**. Verify that the figure is enlarged.

12. Place the cursor on the "Zprev" option and press the **left mouse button**. Verify that the display returns to its original size. Place the cursor on the "Zprev" option again and press the **left mouse button**. Verify that the magnified figure is displayed.

13. Now place the cursor on the "Zres" option and press the **left mouse button**. Verify that the display returns to its original size. Press the **right mouse button** to cancel the *VIEW* menu.

14. From the *Primary P&ID Graphics* menu, place the cursor on the "UTIL" option and press the left mouse button to access the *UTIL* menu.

15. From the *UTIL* menu, place the cursor on the "GRD Off" option and press the left mouse button. Verify that a grid and the P&ID is displayed on the screen. Verify that the "GRD Off" option toggles to the "GRD On" option.

16. Place the cursor on the "GRD On" option and press the left mouse button. Verify that the grid is turned off, and that the original screen background and the P&ID are displayed on the screen. Also verify that the "GRD On" option toggles back to the "GRD Off" option.

17. Place the cursor on the "Sh Name" option and press the left mouse button. Verify that the symbol names are no longer displayed on the screen. Also verify that the "Sh Name" option toggles to the "No Name" option.

18. Place the cursor on the "No Name" option and press the left mouse button. Verify that the symbol names are displayed on the screen. Also verify that the "No Name" option toggles to the "Sh Name" option.

19. Place the cursor on the "Skip" option and press the left mouse button. The message **"Enter grid snap value (default = 1.0, range = [0,120]"** will be displayed at the bottom of the screen. Type **25.0** and press **<enter>**.

20. Place the cursor on the "EDIT" option and press the left mouse button to access the *EDIT* menu. Place the cursor on the "LINE" option and press the left mouse button to access the *LINE* menu.

21. Place the cursor on one of the line options and press the left mouse button. Move the cursor to a desired starting point and press the left mouse button. Drag the cursor to a new location and press the left mouse button to create a new line. Verify that the cursor skipped spaces as the line was created. Press the right mouse button twice to cancel the "Line" option. Press the right mouse button to cancel the *LINE* menu.

22. Return to the *UTIL* menu. Place the cursor on the "Snap On" option and press the left mouse button. Verify that the "Snap On" option toggles to the "Snp Off" option.

23. Return to the *EDIT* menu. Place the cursor on the "Line" option and press the left mouse button to access the *LINE* menu. Place the cursor on one of the line options and press the left mouse button. Move the cursor to a desired starting point and press the left mouse button. Drag the cursor to create the line. Verify that the cursor does not skip any spaces as the line is created. Press the right mouse button twice to cancel the "Line" option.

24. Return to the *UTIL* menu. Place the cursor on the "Snp Off" option and press the left mouse button. Verify that the "Snp Off" option toggles to the "Snap On" option.

25. Place the cursor on the "Skip" option and press the left mouse button. The message **"Enter grid snap value (default = 25.0, range = [0,120]"** will be displayed at the bottom of the screen. Type **0** and press **<enter>**.

26. Return to the *LINE* menu. Place the cursor on one of the line options and press the left mouse button. Move the cursor to the desired starting point and press the left mouse button. Drag the cursor to a new location and press the left mouse button to create a new line. Verify that the cursor did not skip spaces as the line was created. Press the right mouse button twice to cancel the "Skip" option. Press the right mouse button to cancel the *LINE* menu.

27. Return to the *UTIL* menu. Place the cursor on the "Scale" option and press the **left mouse button**. The message "**Enter symbol scale value (default = 2.0, range = [0,10]>**" will be displayed in the bottom left corner of the screen. Type 5 and press <enter>.

28. Return to the *EDIT* menu. Place the cursor on the "SYMBLS" option and press the **left mouse button** to access the *Symbols* menu.

29. Place the cursor on one of symbols and press the **left mouse button**. Drag the symbol to a convenient place on the screen and press the **left mouse button**. Verify that the symbol was magnified. Press the **right mouse button** to cancel the symbols "Scale" option. Press the **right mouse button** again to cancel the *Symbol* menu.

30. From the *Primary P&ID Graphics* menu, place the cursor on the "Show" option and press the **left mouse button**. Verify that the menus are cleared from the screen.

31. Place the cursor on the "EXIT" option and press the **left mouse button**. The message "**Current drawing not saved! Exit anyway ? <Y/N>**" will be displayed in the bottom left corner of the screen. Type **Y** and return to the *FEP* menu.

32. <u>IF</u>:   Steps 1 through 31 were performed successfully

    <u>THEN</u>: Go to Step 33.

    <u>ELSE</u>: Record which step(s) was unsuccessful. Generate an anomaly report and submit it to the SAPHIRE 4.0 V&V committee.

33. End of Section 5.0.

F-6.7 Section 6

Note: *The DOS text editor will be used to test the Symbol Generator feature.*

1. Exit out of the *FEP* menu and return to the PRADATA.B1 directory. Change from the current directory to PRATOOLS.B1 directory.

2. From the PRATOOLS.B1 directory, type **copy smblfile.gen test.gen** and press <enter>. Then type **Edit** and press <enter> to access the MS-DOS editor screen. Press the <ESC> key to clear the dialog box.

3. Place the cursor on the "File" option located in the top left corner of the screen and press the **left mouse button** to access the *File* menu. Place the cursor on the "Open" option and press <enter> to access the screen.

4. Del *.TXT and type *.gen for the file name. Press <enter>. In the files window, verify the SMBLFILE.GEN and TEST.GEN file names.

5. Place the file on the TEST.GEN file name and press the **left mouse button**. Press the **left mouse button** again to access the *TEST.GEN* screen.

6. Delete the first eight lines in order to delete the pump symbol.

7. Place the cursor on the "File" option located in the top left corner of the screen and press the **left mouse button** to access the *File* menu. Place the cursor on the "Exit" option and press the **left mouse button**. The message "**Loaded file is not saved. Save now?**" will be displayed in a window. Place the cursor on the "<yes>" option and press the **left mouse button** to return to the PRATOOL.B1 directory.

8. Type **FEP** and press **<enter>** to access the *FEP* menu. Place the cursor on the "Utility" option and press **<enter>** to access the *Utilities* menu. Place the cursor on the "Symbol Generator" option and press **<enter>** to access the *Symbol Table Generator screen.*

9. Type **Test.gen** and press **<enter>**. The message "**Successful compilation of TEST.STB**" will be displayed at the bottom of the screen. Exit from the *Utilities* menu and return to the *FEP* main menu.

10. Place the cursor on the "P&ID Editor" option and press **<enter>** to access the *Primary P&ID Graphics* menu.

11. From the *Primary P&ID Graphics* menu place the cursor on the "FILE" option and press the **left mouse button** to access the *FILE* menu. Place the cursor on the "SName" option and press the **left mouse button**. The message "**Symbol table name or <or> for [syblfile.stb] :**" will be displayed in the bottom left corner of the screen.

12. Type **Test.stb** and press **<enter>**. Place the cursor on the "EDIT" option and press the **left mouse button** to access the *EDIT* menu. Place the cursor on the "SYMBLS" option and press the **left mouse button**. Verify that the pump symbol is no longer displayed on the screen.

13. Place the cursor on the "EXIT" option and press the **left mouse button** to return to the main *FEP* menu. Then exit from the main *FEP* menu and return to the PRATOOLS.B1 directory.

14. Type **Edit** and press **<enter>** to access the MS-DOS editor screen. Press the **<ESC>** key to clear the dialog box.

15. Place the cursor on the "File" option located on the top left corner of the screen and press the **left mouse button** to access a menu. Place the cursor on the "Open" option and press **<enter>** to access the *Open* screen.

16. Del *.TXT, type *.gen for File Name and press **<enter>**. In the files window, verify the SMBLFILE.GEN and TEST.GEN file names. Place the cursor on the SMBLFILE.GEN and press **<enter>**.

17. Press the **left mouse button** and highlight the first 8 lines of the SMBLFILE.GEN file. Place the cursor on the "Edit" option located in the top left corner of the screen and press the **left mouse button** to access the Edit window.

18. Place the cursor on the "Copy" option and press the **left mouse button**. Place the cursor on the "File" option and press the **left mouse button** to access the *File* menu. Place the cursor on the "Open" option and press the **left mouse button** to access the *Open* screen.

19. Del *.TXT, type *.GEN for the space designated for File Name and press **<enter>**. Place the cursor on the TEST.GEN file name and press **<enter>** to access the *TEST.GEN* screen.

20. Place the cursor on the "Edit" option and press the **left mouse button** to access the *Edit* menu. Place the cursor on the "Paste" option and press **<enter>**. Verify that the eight lines of text from the SYBLFILE.GEN were copied to the *TEST.GEN* screen.

21. Place the cursor on the "File" option located on the top left corner on the screen and press the **left mouse button** to access the *File* menu. Place the cursor on the "Exit" option and press the **left mouse button**. The message "**Loaded file is not saved. Save now?**" will be displayed in a window on the screen. Place the cursor on the "<yes>" option and press the **left mouse button** to return to the PRATOOLS.B1 directory.

22. Repeat Steps 8 though 11.

23. Type **Test.stb** and press <enter>. Place the cursor on the "EDIT" option and press the **left mouse button** to access the *EDIT* menu. Place the cursor on the "SYMBLS" option and press the **left mouse button**. Verify that the pump symbol is displayed on the screen.

24. Place the cursor on the "EXIT" option and press the **left mouse button** to return to the *FEP* main menu. Exit from the *FEP* menu and return to the PRATOOLS.B1 directory. Then change the current directory to the PRADATA.B1 directory.

25. <u>IF</u>:   Steps 1 through 24 were performed successfully

    <u>THEN</u>:   Go to Step 26.

    <u>ELSE</u>:   Record which step(s) was unsuccessful. Generate an anomaly report and submit it to the SAPHIRE 4.0 V&V committee.

26. End of Section 6.

F-6.8 Section 7

1. Type **FEP** and return to the *FEP* main menu. Place the cursor on the "P&ID "Editor" option and press <enter> to access the *Primary P&ID Graphics* menu.

2. Place the cursor on the "Text" option and press the **left mouse button**. The message **"Pick a new color from the color bar"** will be displayed in the bottom left corner of the screen.

3. Place the cursor on one of the colors from the *color bar* menu and press the **left mouse button**. Verify that the "Text" option changes to the color selected.

4. Place the cursor on the "TEXT" option and press the **left mouse button** to access the *TEXT* menu. Then place the cursor on the "Write" option and press the **left mouse button**. The message **"Pick text placement location:"** will be displayed in the bottom left corner of the screen.

5. Move the cursor to a convenient location on the screen and press the **left mouse button** to access the Text window. Type the *text of your choice* and press the <ESC> key. Verify that the typed text is the color that was selected. Press the **right mouse button** to cancel the "Write" option.

6. From the *Primary P&ID Graphics* menu, place the cursor on the "left" option and press the **left mouse button**. Verify that the "left" option toggles to the "cntr" option.

7. Return to the *TEXT* menu. Place the cursor on the "Write" option and press the **left mouse button**. The message **"Pick text placement location"** will be displayed in the bottom left corner of the screen.

8. Move the cursor to a convenient location on the screen and press the **left mouse button** to access the Text window. Type *text of your choice* and press the <ESC> key. Verify that the text on the screen is centered around the placement point. Press the **right mouse button** to cancel the "Write" option.

9. From the *Primary P&ID Graphics* menu, place the cursor on the "cntr" option and press the **left mouse button**. Verify that the "cntr" option toggles to the "rght" option.

10. Return to the *TEXT* menu. Place the cursor on the "Write" option and press the **left mouse button**. The message **"Pick text placement location"** will be displayed in the bottom left corner of the screen.

11. Move the cursor to a convenient location on the screen and press the **left mouse button** to access the Text window. Type the *text of your choice* and press the <ESC> key. Verify that the text on the screen is set to the right of the placement point. Press the **right mouse button** to cancel the "Write" option.

12. From the *Primary P&ID Graphics* menu, place the cursor on the "rght" option and press the **left mouse button.** Verify that the "rght" option toggles to the "left" option.

13. Return to the *TEXT* menu. Place the cursor on the "Write" option and press the **left mouse button.** The message **"Pick text placement location"** will be displayed in the bottom left corner of the screen.

14. Move the cursor to a convenient location on the screen and press the **left mouse button** to access the Text window. Type the *text of your choice* and press the **<ESC>** key. Verify that the text on the screen is set to the left of the placement point. Press the **right mouse button** to cancel the "Write" option. Press the right mouse button again to cancel the *TEXT* menu.

15. From the *Primary P&ID Graphics* menu, place the cursor on the "Fill" option and press the **left mouse button.** The message **"Pick a new color from the color bar"** will be displayed in the bottom left corner of the screen.

16. Place the cursor on one of the colors on the color bar and press the **left mouse button.** Verify that the "Fill" option changes to the color that was selected.

17. From the *Primary P&ID Graphics* menu, place the cursor on the "EDIT" option and press the **left mouse button** to access the *EDIT* menu. Place the cursor on the "SYMBLS" option and press the **left mouse button** to access the *Symbol* menu.

18. Place the cursor on one of the symbols and press the **left mouse button.** Move the symbol to a convenient location on the screen and press the **left mouse button** again. Verify that the fill color matches the color that was selected. Press the **right mouse button** twice to cancel the *Symbol* menu.

19. Return to the *Primary P&ID Graphics* menu. Place the cursor on the "On" option and press the **left mouse button.** Verify that the "On" option toggles to the "Off" option.

20. Return to the *EDIT* menu. Place the cursor on the "SYMBLS" option and press the **left mouse button** to access the *Symbol* menu. Place the cursor on one of the symbols and press the **left mouse button.** Move the symbol to a convenient location on the screen and press the **left mouse button.** Verify that the fill color is no longer displayed with the symbol. Press the **right mouse button** twice to cancel the *Symbol* menu.

21. Return to the *Primary P&ID Graphics* menu. Place the cursor on the "Line" option and press the **left mouse button.** The message **"Pick a new color from the color bar"** will be displayed in the bottom left corner of the screen.

22. Place the cursor on one of the colors on the color bar and press the **left mouse button.** Verify that the "Line" option changed to the color that was selected.

23. Return to the *EDIT* menu. Place the cursor on the "Line" option and press the **left mouse button** to access the *Line* menu. Place the cursor on one of the line options and press the **left mouse button.** The message **"Enter points in order of direction of flow"** will be displayed in the bottom left corner of the screen.

24. Move the cursor to a desired location and press the **left mouse button.** Drag the cursor to a new location and press the **left mouse button.** Verify that the line matches the color that was selected. Press the **right mouse button** twice to cancel the "Line" option.

25. Place the cursor on the "EXIT" option and press the **left mouse button.** The message **"Current drawing not saved! Exit anyway? <Y/N>"** will be displayed in the bottom left corner of the screen. Type **Y** to return to the *FEP* main menu.

26. <u>IF</u>:   Steps 1 through 25 were performed successfully

    <u>THEN</u>:   Go to Step 27.

    <u>ELSE</u>:   Record which step(s) was unsuccessful. Generate an anomaly report and submit it to the SAPHIRE 4.0 V&V committee.

27. End of Section 7.

F-6.9 Section 8

Note: *The purpose of this section is to test the Extract P&ID feature. However, this set of instructions would not be the preferred way to test the Extract P&ID feature.*

1.   Exit from the *FEP* menu and return to the PRADATA.B1 directory. Type **MARD** and press <enter> to access the *MARD* menu.

2.   Place the cursor on the "Load Data" option and press <enter> to access the *Load* screen. Place the cursor on the "MAR-D" option and press <enter> to access the *MAR-D Data* menu. Then place the cursor on the "SYstem" option and press <enter> to access the *Select System Data* menu.

3.   Place the cursor on the "Piping & Instr. PID" option and press <enter> to access the *Load* screen. Verify the file names CONT and SYMBOL. Type L and press <enter> load the files. The message **"File name NOT selected. Do All? [Y,N] Y"** will be displayed at the bottom on the screen. Type Y. Verify that the message **"Load complete"** appears in the bottom left corner of the screen. Exit from this screen and return to the *MARD* menu.

4.   Exit from the *MARD* menu and return to the PRADATA.B1 directory. Type **FEP** to access the *FEP* menu. Then place the cursor on the "Utilities" option and press <enter> to access the *Utilities* screen. Place the cursor on the "eXtract Graphics" option and press <enter> to access the *Extract* menu. Then place the cursor on the "P&ID" option and press <enter> to access the *Extract P&ID* screen. Verify that the file names CONT and SYMBOL are displayed on the *Extract P&ID* screen.

5.   Type **D** and press <enter> to access the *Delete P&IDs* screen. Type **D** and press <enter> again. The message **"Process all entries (Y/N)"** will be displayed at the bottom of the screen. Type Y. Verify that the file names CONT and SYMBOL are no longer displayed on the *Delete P&IDs* screen.

6.   Exit from the *P&ID* screen and return to the *FEP* menu. Place the cursor on the "P&ID editor" option and press <enter> to access the *Primary P&ID Graphics* menu.

7.   Place the cursor on the "FILE" option and press the **left mouse button** to access the *FILE* menu. Place the cursor on the "List" option and press the **left mouse button**. Verify that the file names CONT and SYMBOL are no longer displayed on the screen.

8.   Place the cursor on the "EXIT" option and press the **left mouse button** to return to the *FEP* menu. Place the cursor on the "UTILITY" option and press <enter> to access the *Utilities* menu. Place the cursor on the "EXtract Graphics" option and press <enter> to access the *Extract* menu.

9.   Place the cursor on the "P&ID" option and press <enter> to access the *Extract P&ID* screen. Type **X** and press <enter> to extract the files. The message **"process all entries"** will be displayed at the bottom on the screen. Type Y.

10. Exit from the *Extract P&ID* screen and return to the *FEP* menu. Place the cursor on the "P&ID editor" option and press <enter> to access the *Primary P&ID Graphics* menu.

11. Place the cursor on the "FILE" option and press the **left mouse button** to access the *FILE* menu. Place the cursor on the "List" option and press the **left mouse button**. Verify that the file names CONT and SYMBOL are displayed on the screen.

12. Place the cursor on the file name CONT and press the **left mouse button**. Verify that the P&ID is displayed on the screen. Then place the cursor on the "EXIT" option and press the **left mouse button** to return to the *FEP* menu.

13. IF: Steps 1 through 12 were performed successfully

    THEN: Go to Step 14.

    ELSE: Record which step(s) was unsuccessful. Generate an anomaly report and submit it to the SAPHIRE 4.0 V&V committee.

14. End of Section 8.

F-7. PROCEDURE 4—MAR-D NONVITAL FEATURES

F-7.1 Introduction

The Models and Results Database (MARD-D) is a program that is used primarily for PRA data loading. This program defines a common relational database structure that is used by the entire suite of programs. This structure allows all of the software to access and manipulate data created by other software in the system without performing a lengthy conversion. These created by IRRAS are immediately available to SARA for sensitivity analysis. MAR-D also provides the facilities for loading and unloading of PRA data from the relational database structure used to store data. A simple ASCII data format is used for interchange with other PRA software not included in NRC's suite of programs. This feature allows for compatibility with previously developed software systems and allows for maximum data interchange. Elements of this software are included with both IRRAS and SARA to allow these programs to load and unload data in MAR-D format. Normally, the entire MAR-D software is used only by those performing a data loading function and is not required by the user.

This test procedure tests the nonvital features associated with MAR-D. The minor features tested from the MAR-D program are Family, Load Data, Extract Data, Modify Database, Reports, and Utility. Table F-5 lists the features, the success criteria and the specific options that are tested in each of the sections. The steps in each section of this procedure should be followed. If any step is not completed successfully, an anomaly report should be generated and submitted to the SAPHIRE 4.0 V&V committee.

Table F-5. MAR-D test procedure specifications.

Sections	Feature group	Features	Success criteria	Options tested
1	Load Data	Family	To load FAD, FAA, and FAT data files	Description·FAD Attributes·FAA, Text·FAT
		Attributes	To load, CAD, CTD, FMD, LCD, and STD data files	Class Attributes·CAD, Basic Event Types·CTD, Failure Modes·FMD, Location Description·LCD, System Types·STD

Appendix F

Table F-5. MAR-D test procedure specifications.

Sections	Feature group	Features	Success criteria	Options tested
		Basic Event	To load BED, BEI, and BEA data files	Description·BED, Rate Information·BEI, Attributes·BEA
		Systems	To load FTD, FTL, DLS, FTT, PID, FTC, FTA	Description·FTD Logic·FTL Graphic·DLS Piping & Inst·PID Cutsets·FTC Attributes·FTA
		Event Trees	To load ETD, ETG, ETL, ETA, ETR and ETT data files	Description·ETD, Graphics·ETG, Logic·ETL Attributes·ETA, Rules·ETR Text·ETT
		End State	To load ESD, ESI and EST data files	Description·ESD, Information·ESI, Text·EST
		Sequence	To load SQD, SQA, SQL, SQC, and SQT data files	Description·SQD, Attributes·SQA, Logic·SQL Text·SQT, Cutsets·SQC
		Gate	To load GTD, and GTA data files	Description·GTD, Attributes·GTA
		Change Sets	To load CSD and CSI data files	Description·CSD, Information·CSI
2	Extract Data	Family	To extract FAD, FAA, and FAT data files from the database	Description·FAD, Attributes·FAA, Text·FAT
		Attributes	To extract, CAD, CTD, FMD, LCD, and STD data files from the database	Class Attributes·CAD, Basic Event Types·CTD, Failure Modes·FMD, Location·LCD System Types·STD

Table F-5. MAR-D test procedure specifications.

Sections	Feature group	Features	Success criteria	Options tested
		Basic Event	To extract, BED, BEI, and BEA data files from the database	Description·BED, Rate Information·BEI, Attributes·BEA
		Systems	To extract FED, FTL. DLS, FTC, FTA, FTT, and PID files from the database	Description·FTD, Logic·FTL, Graphics·DLS, Cutsets·FTC, Attributes·FTA, Text·FTA, Piping & Instr·FTA
		Event Trees ·	To extract ETD, ETG, ETL, ETA, ETR and ETT data files from the database	Description·ETD, Graphics·ETG, Logic·ETL Rules·ETR, Text·ETT Attributes·ETT
	Extract Data	Endstate	To extract ESD, ESI and EST data files from the database	Description·ESD, Text·EST
	Extract Data	Sequence	To extract SQD, SQA, SQL, SQC, and SQT data files from the database	Description·SQD, Attributes·SQA, Logic·SQL Cutset·SQC, Text·SQT
		Gate	To extract GTD, and GTA data files from the database	Description·GTD, Attributes·GTA
		Change sets	To extract CSD and CSI data files from the database	Description·CSD, Information·CSI
3	Modify Data Base	Event Trees	To modify event tree data records	Add, Modify, Text, Delete and Sequences.
		Systems	To modify system data records	Add, Modify, Text, Delete
4		Endstates	To modify end- states data records	Add, Modify, Text, Delete
		Basic Events	To modify basic events data records	Add, Modify, Delete, Remove Unused Events

Appendix F

Table F-5. MAR-D test procedure specifications.

Sections	Feature group	Features	Success criteria	Options tested
		Attributes 1.System 2. Location 3. Failure Mode 4. Class Attributes 5. Basic Event Types 6. Trains	To modify the six attributes	Add, Modify, Delete
5		Gates	To modify gate records	Add, Modify, Delete
		Graphics 1. Fault Tree 2. Event Tree 3. P&ID	To modify three graphics data records	Add, Modify, Delete
6	Reports	Summary	To list the data available for a selected family	Console, Printer, File
		Counts	To provide a count of the number of records in each data type currently in the database	Console, Printer, File
	Reports	Detail 1. All 2. Basic Events 3. System 4. Endstates	To display the details of the family data stored in the database	Console, Printer, File
	Utility	Recover Database	To restructure the database and re-index the data	Recover Everything

F-7.2 Section 1

1. Enter the PRADATA.B1 directory. Type **mard** and press <enter> to access the *MAR-D* menu. Place the cursor on the "MODIFY Database" option and press <enter> to access the *Modify Database* menu.

2. Place the cursor on the "Family" option and press <enter> to access the *Edit Family* screen.

3. Type **A** to add and press <enter> to access the *Add family* screen. Type **pbtrain** as the new family name and press <enter> twice to return to the *Edit Family* screen. Exit and return to the *MAR-D* menu.

4. Place the cursor on the "Select Family" option and press <enter> to access the *Select Family* screen. Place the cursor on the family named PBTRAIN and press <enter>. The name PBTRAIN will be displayed in the top left corner of the screen.

5. Exit from the MAR-D program and return to the PRADATA.B1 directory. Then access the PRADATA.B1\PBTRAIN subdirectory.

6. Copy the files from diskette A to the subdirectory. When all the files are loaded, return to the PRADATA.B1 directory.

7. Make a subdirectory called Temp and repeat Step 6.

8. Type **MARD** and press <enter> to access the *MAR-D* menu. Place the cursor on the "Load Data" option and press <enter> to access the *Load* screen. Place the cursor on the "MAR-D" option and press <enter> to access the *Select MAR-D Data* menu.

9. Place the cursor on the "Family" option and press <enter> to access the *Select Family Data* menu.

10. Place the cursor on the "Description.FAD" option and press <enter>. Verify that the file name PBTRAIN is displayed on the screen and press <enter>. Verify that the message **"Load Complete"** is displayed at the bottom of the screen. Exit and return to the *Select Family Data* menu.

11. Repeat Step 10 for the "Attributes. FAA" option and the "Text. FAT" option. When complete, exit and return to the *Select MAR-D Data* menu.

12. Place the cursor on the "Attributes" option and press <enter> to access the *Select Attribute Data* menu.

13 Place the cursor on the "Class Attributes.CAD" option and press <enter>. Verify that the family name PBTRAIN is displayed on the screen and press <enter>. Verify that the message **"Load Complete"** is displayed on the bottom of the screen. Exit and return to the *Select Attribute* menu.

14. Repeat step 13 for the following options.

   a. Basic Event Types.CTD

   b. Failure Modes.FMD

   c. Location Description.LCD

   d. System Types.STD

   When complete, exit and return to the *Select MAR-D Data* menu.

15. From the *Select MAR-D Data* menu, place the cursor on the "Basic Event" option and press <enter> to access the *Select Basic Event Data* menu.

16. Place the cursor on the "Description.BED" option and press <enter>. Verify that the family name PBTRAIN is displayed on the screen and press <enter>. Also, verify that the message **"Load Complete"** is displayed at the bottom of the screen. Exit and return to the *Select Basic Event* menu.

17. Repeat Step 16 for the "Rate Information.BEI" option and the "Attributes.BEA" option. Then exit and return to the *Select MAR-D Data* menu.

18. From the *Select MAR-D Data* menu place the cursor on the "SYstems" option and press <enter> to access the *Select System Data* menu.

19. Place the cursor on the "Descriptions.FTD" option and press <enter>. Verify that the file names are displayed on the screen and then press <enter>. Verify that the message **"Load Complete"** is displayed at the bottom of the screen. Exit and return to the *Select System Data* menu.

20. Repeat step 19 for the following options:

        a. Logic.FTL

        b. Graphics.DLS

        c. Text.FTT

        d. Piping & Instr.PID

21. From the *Select System Data* menu, place the cursor on the "CutSets.FTC" option and press <enter>. Verify that the file name PBTRAIN is displayed on the screen and press <enter>. Verify the message **"Load Complete"** is displayed at the bottom of the screen. Exit and return to the *Select System Data* menu.

22. Repeat Step 21 for the "Attributes.FTA" option. Then exit and return to the *Select MAR-D Data* menu.

23. From the *Select MAR-D Data* menu, place the cursor on the "Event Tree" option and press <enter> to access the *Select Event Tree Data* menu.

24. Place the cursor on the "Descriptions.ETD" option and press <enter>. Verify that the file name PBTRAIN is displayed on the screen and press <enter>. Verify that the message **"Load Complete"** is displayed at the bottom of the screen. Exit and return to the *Select Event Tree Data* menu.

25. Repeat step 24 for the following options:

        a. Graphics.ETG

        b. Logic.ETL

        c. Attributes.ETA

        d. Rules.ETR

        e. Text.ETT

    Exit and return to the *Select MAR-D Data* menu.

26. From the *Select MAR-D Data* menu, place the cursor on the "ENDstate" option and press <enter> to access the *Select Endstate Data* menu.

27. Place the cursor on the "Descriptions.ESD" option and press <enter>. Verify that the file name PBTRAIN is displayed on the screen and press <enter>. Verify that the message **"Load Complete"** is displayed at the bottom of the screen. Exit and return to the *Select MAR-D Data* menu.

28. Repeat Step 27 for the "Information .ESI" option and the "Text" option. When complete, exit and return to the *Select MAR-D Data* menu.

29. From the *Select MAR-D Data* menu, place the cursor on the "Sequence" option and press <enter> to access the *Select Sequence Data* menu.

30. Place the cursor on the "Descriptions.SQD" option and press <enter>. Verify that the file names are displayed on the screen. Type L for load and press <enter>. The message **"File name NOT selected. Do All? [Y/N] Y."** Type Y. Verify the message **"Load Complete"** at the bottom of the screen. Exit and return to the *Select Sequence Data* menu.

31. Repeat Step 30 for the following options.

   a. Attributes.SQA

   b. Logic.SQL

   c. Text.SQT

   Exit and return to the *Select Sequence Data* menu.

32. Place the cursor on the "Cutsets . SQC" option and press <enter>. Verify that the file names are on the screen. Type **B** and press <enter>. The message "**File name NOT selected. Do All? [Y/N] Y.**" Type **Y**. Verify the message "**Load Complete**" at the bottom of the screen. Exit and return to the *Select MAR-D Data* menu.

33. From the *Select MAR-D Data* menu, place the cursor on the "Gate" option and press <enter> to access the *Select Gate Data* menu.

34. Place the cursor on the "Descriptions.GTD" option and press <enter>. Verify the file name PBTRAIN on the screen and press <enter>. Verify the message "**Load Complete**" at the bottom of the screen. Exit and return to the *Select Gate Data* menu.

35. Repeat Step 34 for the "Attributes.GTA" option. Then exit and return to the *Select MAR-D Data* menu.

36. From the *Select MAR-D* menu, place the cursor on the "Change Set" option and press <enter> to access the *Select Change Set Data* menu.

37. Place the cursor on the "Description.CSD" option and press <enter>. Verify that the file name PBTRAIN is on the screen and press <enter>. Verify that the message "**Load Complete**" is displayed at the bottom of the screen. Exit and return to the *Select Change Set Data* menu.

38. Repeat Step 37 for the "Information.CSI" option. Exit and return to the *Select MAR-D Data* menu. Then exit from the *Select MAR-D Data* menu and return to the *MAR-D* menu.

39. IF: Steps 1 through 38 were performed successfully

   THEN: Go to Step 40

   ELSE: Record which step(s) was unsuccessful. Generate an anomaly report and submit it to the SAPHIRE 4.0 V&V committee.

40. End of Section 1.

F-7.3 Section 2

1. Exit from the *MAR-D* menu and return to the PRADATA.B1 directory. Then access the PRADATA.B1\TEMP subdirectory.

2. Type **dir/w** and press <enter> to obtain a list of those files that were copied. Then press the Print Screen key to obtain a hard copy of the files.

3. Change from the PRADATA.B1\TEMP subdirectory to the PRADATA.B1\PBTRAIN subdirectory and delete those files that are shown on the printout. When all the files are deleted, return to the PRADATA.B1 directory.

4. Type **MARD** and press <enter> to access the *MAR-D* menu. Place the cursor on the "EXTRACT Data" option and press <enter> to access the *Extract* screen. Place the cursor on the "MAR-D" option and press <enter> to access the *Select MAR-D Data* menu.

5. Place the cursor on the "Family" option and press <enter> to access the *Select Family Data* menu.

6. From the *Select Family Data* menu, place the cursor on the "Description.FAD" option and press <enter>. The message **"Output Data for Whole Family [Y,N] Y"** will be displayed at the bottom of the screen. Type **Y**. Verify the message **"Output Complete"** displayed at the bottom of the screen.

7. Repeat Step 6 but place the cursor on the "Attributes.FAA" option and then the "Text.FAT" option. When complete, exit and return to the *Select MAR-D Data* menu.

8. Place the cursor on the "Attributes" option and press <enter> to access the *Select Attribute Data* menu.

9. Place the cursor on the "Class Attributes.CAD" option and press <enter>. The message **"Output Data for Whole Family [Y,N] Y"** will be displayed at the bottom of the screen. Type **Y**. Verify the message **"Output Complete"** displayed at the bottom of the screen.

10. Repeat Step 9. for the following options:

    a. Basic Event Types.CTD

    b. Failure Modes.FMD

    c. Location Description.LCD

    d. System Types.STD

    When complete, exit and return to the *Select MAR-D Data* menu.

11. From the *Select MAR-D Data* menu, place the cursor on the "Basic Event" option and press <enter> to access the *Select Basic Event Data* menu.

12. Place the cursor on the "Description.BED" option and press <enter>. The message **"Output Data for Whole Family [Y,N] Y"** will be displayed at the bottom of the screen. Type **Y**. The message **"Output Complete"** will be displayed at the bottom of the screen.

13. Repeat Step 12 for the "Rate Information.BEI" option and the then "Attributes . BEA" option. When complete, exit and return to the *Select MAR-D Data* menu.

14. From the *Select MAR-D Data* menu, place the cursor on the "SYstems" option and press <enter> to access the *Select System Data* menu.

15. Place the cursor on the "Descriptions.FTD" option and press <enter>. The message **"Output Data for Whole Family [Y,N] Y"** will be displayed at the bottom of the screen. Type **Y**. The message **"Output Complete"** will be displayed at the bottom of the screen.

16. Place the cursor on the "Logic.FTL" option and press <enter> to access the *Extract* screen. Type **X** and press <enter>. The message **"Name not selected. Do all? [Y/N] Y."** will be displayed at the bottom of the screen. Type **Y**. The message **"Output Complete"** will be displayed at the bottom of the screen. Exit and return to the *Select System Data* menu.

17. From the *Select System Data* menu, place the cursor on the "Graphics.DLS" option and press <enter> to access the *Extract* screen. Type **X** and press <enter>. The message **"Name not selected.Do all?**

[Y,N] Y" will be displayed at the bottom of the screen. Type Y. The message "Output Complete" will be displayed at the bottom of the screen. Exit and return to the *Select System Data* menu.

18. Place the cursor on the "Cut Sets.FTC" option and press <enter> to access the *Extract* screen. Type B and press <enter>. The message "Name not selected Do all? [Y,N] Y" will be displayed at the bottom of the screen. Type Y. The message "Output Complete" will be displayed at the bottom of the screen. Exit and return to the *Select System Data* menu.

19. Place the cursor on the "Attributes.FTA" option and press <enter>. The message "Output Data for Whole Family [Y,N] Y" will be displayed at the bottom of the screen. Type Y. The message "Output which type. B = Base Case, A= Alternate." will be displayed at the bottom of the screen. Type B. The message "Output Complete" will be displayed at the bottom of the screen.

20. Place the cursor on the "Text.FTT" option and press <enter> to access the *Extract* screen. Type X and press <enter>. The message "Name not selected Do all? [Y,N] Y" will be displayed at the bottom of the screen. Type Y. The message "Output Complete" will be displayed at the bottom of the screen. Exit and return to the *Select System Data* menu.

21. Place the cursor on the "Piping & Instr.PID" option and press <enter> to access the *Extract* screen. Type X and press <enter>. The message "Name not selected Do all? [Y,N] Y" will be displayed at the bottom of the screen. Type Y. The message "Output Complete" will be displayed at the bottom of the screen. Exit and return to the *Select System Data* menu. Exit from the *Select System Data* menu and return to the *Select MAR-D Data* menu.

22. From the *Select MAR-D Data* menu, place the cursor on the "EVent Tree" option and press <enter> to access the *Select Event Tree Data* menu.

23. Place the cursor on the "Descriptions.ETD" option and press <enter>. The message "Output Data for Whole Family [Y,N] Y" will be displayed at the bottom of the screen. Type Y. The message "Output Complete" will be displayed at the bottom of the screen. Place the cursor on the "Graphics.ETG" option and press <enter> to access the *Extract* screen.

24. Type X and press <enter>. The message "Name not selected. Do all? [Y,N] Y" will be displayed at the bottom of the screen. Type Y. The message "Output Complete" will be displayed at the bottom of the screen. Exit and return to the *Select Event Tree Data* menu.

25. Repeat step 24 for the following options:

   a. Logic.ETL

   b. Rules.ETR

   c. Text.ETT

   Exit and return to the *Select Event Tree Data* menu.

26. Place the cursor on the "Attributes.ETA" option and press <enter>. The message "Output Data for Whole Family [Y,N] Y" will be displayed at the bottom of the screen. Type Y. The message "Output Complete" will be displayed at the bottom of the screen. Exit from the *Select Event Tree Data* menu and return to the *Select MAR-D Data* menu.

27. From the *Select MAR-D Data* menu, place the cursor on the "ENDstate" option and press <enter> to access the *Select Endstate Data* menu.

28. Place the cursor on the "Descriptions.ESD" option and press <enter>. The message "**Output Data for Whole Family [Y,N] Y**" will be displayed at the bottom of the screen. Type Y. The message "**Output Complete**" will be displayed at the bottom of the screen.

    Note: *The "Information.ESI" option has not been implemented.*

29. Place the cursor on the "Text.EST" option and press <enter> to access the Extract screen. Type X and press <enter>. The message "**Name not selected Do all? [Y,N] Y**" will be displayed at the bottom of the screen. Type Y. The message "**Output Complete**" will be displayed at the bottom of the screen. Exit and return to the *Select Endstate Data* menu. Then exit from the *Select Endstate Data* menu and return to the *Select MAR-D Data* menu.

30. From the *Select MAR-D Data* menu, place the cursor on the "Sequence" option and press <enter> to access the *Select Sequence Data* menu.

31. Place the cursor on the "Descriptions.SQD" option and press <enter>. The message "**Output Data for Whole Family [Y,N] Y**" will be displayed at the bottom of the screen. Type Y. The message "**Output Complete**" will be displayed at the bottom of the screen.

32. Place the cursor on the "Cutsets.SQC" option and press <enter> to access the *Extract* screen. Type B and press <enter>. The message "**Name not selected Do all? [Y,N] Y**" will be displayed at the bottom of the screen. Type Y. The message "**Output Complete**" will be displayed at the bottom of the screen. Exit and return to the *Select Sequence Data* menu.

33. Place the cursor on the "Attributes.SQA" option and press <enter>. The message "**Output Data for Whole Family [Y,N] Y**" will be displayed at the bottom of the screen. Type Y. The message "**Output which type. B = Base Case, A= Alternate. B**" will be displayed at the bottom of the screen. Type B. The message "**Output Complete**" will be displayed at the bottom of the screen.

34. Place the cursor on the "Logic.SQL" option and press <enter> to access the *Extract* screen. Type X and press <enter>. The message "**Name not selected Do all? [Y,N] Y**" will be displayed at the bottom of the screen. Type Y. The message "**Output Complete**" will be displayed at the bottom of the screen. Exit and return to the *Select Sequence Data* menu.

35. Place the cursor on the "Text.SQT" option and press <enter> to access the *Extract* screen. Type X and press <enter>. The message "**Name not selected Do all? [Y,N] Y**" will be displayed at the bottom of the screen. Type Y. The message "**Output Complete**" will be displayed at the bottom of the screen. Exit and return to the *Select Sequence Data* menu. Then exit and return to the *Select MAR-D Data* menu.

36. From the *Select MAR-D Data* menu, place the cursor on the "Gate" option and press <enter> to access the *Select Gate Data* menu.

37. Place the cursor on the "Descriptions.GTD" option and press <enter>. The message "**Output Data for whole Family [Y,N] Y**" will be displayed at the bottom of the screen. Type Y. The message "**Load Complete**" will be displayed at the bottom of the screen.

38. Repeat Step 37 for the "Attributes.GTA" option. Then exit and return to the *Select MAR-D Data* menu.

39. From the *Select MAR-D* menu, place the cursor on the "Change Set" option and press <enter> to access the *Select Change Set Data* menu.

40. Place the cursor on the "Descriptions.CSD" option and press <enter>. The message "**Output Data for whole Family [Y,N] Y**" will be displayed at the bottom of the screen. Type **Y**. The message "**Load Complete**" will be displayed at the bottom of the screen.

41. Repeat Step 40 for the "Information .CSI" option. Exit and return to the *Select MAR-D Data* menu. Then exit the *Select MAR-D* menu and return to the *MAR-D* menu.

42. Exit from the *MAR-D* menu and return to the PRADATA.B1 directory. Then access the PRADATA.B1\TEMP subdirectory. Type **DIR | SORT>LPT1** to obtain a printout of all the files.

    Note: *The following instructions are intended to determine whether the files were successfully extracted by comparing the files in the PRADATA.B1\TEMP subdirectory to the files in PRADATA.B1\PBTRAIN subdirectory byte by byte.*

43. Type **comp** and press <enter> The message "**Name of first file to compare:**" will be displayed on the screen. From the first file from the printout, type **A.etg** and press <enter>. The message "**Name of second file to compare:**" will be displayed on the screen.

44. Type **C:\PRADATA.B1\PBTRAIN\A.ETG** and press <enter>. The message "**Option:**" will be displayed on the screen. Then type **/a** and press <enter> twice. If the files are the same, the message "**Files compare OK**" will be displayed on the screen.

45. Follow the DOS prompt and Repeat Steps 43 and 44 for the remaining files listed on the print-out

46. **IF:**   Steps 1 through 45 were performed successfully

    THEN: Go to Step 47.

    ELSE: Record which step(s) was unsuccessful. Generate an anomaly report and submit it to the SAPHIRE 4.0 V&V committee.

47. End of Section 2.

F-7.4 Section 3

1. Access the PRADATA.B1 directory and type **mard** to access the *MAR-D* menu. Place the cursor on the "Modify Database" option to access the *Modify Database* menu. Place the cursor on the "EVent Trees" option and press <enter> to access the *Edit Event Trees* screen. A list of event trees along with descriptions will be displayed on the screen.

2. From the *Edit Event Trees* screen, type **A** to Add. The press <enter> to access the *Add Event Tree* screen.

3. Type **test2** in the space for Name. Move the cursor in the Description space and type **This is a test**. Move the cursor in the space designated for the Initiating Event Name. A list of initiating events will be displayed in a window. Press the <F1> key to place the cursor in the window. Then pick an initiating event and press <enter>. The selected Initiating Event name will be displayed. Press <enter> to return to the *Edit Event Trees* screen. Verify that the event tree name along with the description were added to the *Edit Event Trees* screen.

4. Type **M** for modify and place the cursor on the name called **TEST2**. Press <enter> to access the *Modify Event Tree* screen.

5. Place the cursor in the space for Description and type **new information**. Press <enter> to return to the *Edit Event Trees* screen. Verify that the descriptive information was modified.

6. From the *Edit Event Trees* screen, type T for text. Place the cursor on the name called test2, and press <enter> to access the *Edit Text* screen. A blank screen will be displayed.

7. Press <Alt-A> and type the **text of your choice**. Verify that the typed text was displayed on the screen.

8. Press <Alt-B>. Verify that a line is added before the current line. Type **text of your choice**.

9. Press <Alt-D>. Verify that the line of text is deleted. Then press <Alt-R>. Verify that the line of text is restored to the screen.

10. Place the cursor at the beginning of the text that was typed in Step 7. Press <Ctrl-End>. Verify that text was deleted from the screen.

11. Press <ESC> key to return to the *Edit Event Trees* screen. The message "Text Record not modified" will be displayed. Press <enter> again to return to the *Edit Text* screen. A blank screen will be displayed.

12. Press <Alt-A> and type the **text of your choice**. Verify that the typed text is displayed on the screen.

13. Pressing the <Del> key to delete a character. Verify that the character was deleted. Press the <Ins> key to insert a space. Verify that a space was created.

14. Press <Ctrl-Z> to return to the *Edit Event Trees* screen. The message "Text record modified" will be displayed at the bottom of the screen. Press <enter> again to access the *Edit Text* screen. Verify that the revisions made were saved. Exit and return to the *Edit Event Trees* screen.

15. Type D for delete and place the cursor on event tree **TEST2**. Press <enter> to access the *Delete Event Trees* screen. Type D again and press <enter>. A WARNING message will be displayed on the screen. Type Y and press <enter>. The message "Record deleted" will be displayed at the bottom of the screen. Verify that the name TEST2 was deleted from the *Edit Event Trees* screen.

16. Type S for Sequence and place the cursor on the event tree named A. Then press <enter> to access the *Edit Sequences* screen.

17. From the *Edit Sequences* screen, type A to Add and press <enter> to access the *Add Sequence* screen.

18. Type A-V5V6 in the space for Name. Move the cursor in the Description space and type **This is a test**. Then move the cursor in the space designated for EndState Name. A list of EndStates will be displayed in a window. Press the <F1> key to place the cursor inside the window. Chose an endstate name and press <enter>. Verify that the selected end State name is displayed in the appropriate space. Press <enter> again to return to the *Edit Sequence* screen. Verify that the sequence name along with the description were added to the *Edit Sequence* screen.

19. Type M for modify and place the cursor on the sequence name called **A-V5V6**. Press <enter> to access the *Modify Sequence* screen.

20. Place the cursor in the space for Description and type **new information**. Press <enter> to return to the *Edit Event Trees* screen. Verify that the descriptive information was modified.

21. From the *Edit Sequences* screen, type T for text, place the cursor on the name called A-V5V6, and press <enter> to access the *Edit Text* screen. A blank screen will be displayed.

22. Repeat Steps 7 through 14. When complete, exit and return to the *Edit Sequences* screen.

23. Type **D** for delete and place the cursor on sequence name **A-V5V6**. Press <enter> to access the *Delete Sequence* screen. Type **D** again and press <enter>. The message **"Record deleted"** will be displayed at the bottom of the screen. Verify that the sequence name **A-V5V6** was deleted from the *Edit Sequences* screen. Exit and return to the *Edit Event Trees* Screen. Then return to the *Modify Database* menu.

24. Place the cursor on the "SYstems" option and press <enter> to access the *Edit Systems* screen.

25. From the *Edit System* screen, type **A** to add and press <enter> to access the *Add System* screen.

26. Type **SYS2** in the space for Name. Move the cursor in the Description space and type **This is a test.** Press <enter> to return to the *Edit Systems* screen. The message "Record Added" will be displayed at the bottom of the screen. Verify that the system name along with the description were added to the *Edit Systems* screen.

27. Type **M** for modify and place the cursor on the system name **SYS2**. Press <enter> to access the *Modify Systems* screen.

28. Place the cursor in the space for Description and type **new information**. Press <enter> to return to the *Edit Systems* screen. The message **"Record modified"** will be displayed at the bottom of the screen. Verify that the descriptive information was modified.

29. From the *Edit Systems* screen type **T** for text and place the cursor on the name **SYS2** and press <enter> to access the *Edit Systems* screen. The screen should be blank.

30. Repeat Steps 7 through 14. Then exit and return to the *Edit Systems* screen.

31. Type **D** for delete and place the cursor on the name **SYS2**. Press <enter> to access the *Delete System* screen. Type **D** again and press <enter>. The message **"Record deleted"** will be displayed at the bottom of the screen. Verify that the system named **SYS2** was deleted from the *Edit Systems* screen. Exit and return to the "Modify Database" option.

32. **IF:**   Steps 1 through 31 were performed successfully

    THEN: Go to Step 33.

    ELSE: Record which step(s) was unsuccessful. Generate an anomaly report and submit it to the SAPHIRE 4.0 V&V committee.

33. End of Section 3.

F-7.5 Section 4

1. From the *Modify Database* menu, place the cursor on the "ENd States" option and press <enter> to access the *Edit EndState* screen. The endstates with their descriptions will be listed.

2. From the *Edit EndStates* screen, type **A** for add and press <enter> to access the *Add EndState* screen.

3. Type **T1-TEST** in the space for Name. Move the cursor in the Description space, and type **This is a test.** Press <enter> to return to the *Edit EndState* screen. Verify that the end state name along with the description were added to the *Edit EndState* screen.

4. Type **M** for modify and place the cursor on the name called **T1-TEST**. Press <enter> to access the *Modify EndState* screen.

5. Place the cursor in the space for Description and type **new information**. Press <enter> to return to the *Edit EndState* screen. Verify that the descriptive information was modified.

6. From the *Edit EndState* screen, type **T** for text, place the cursor on the name called T1-TEST, and press <enter> to access the *Edit Text* screen. A blank screen will be displayed.

7. Press <Alt-A> and type the **text of your choice**. Verify that the typed text is displayed on the screen.

8. Press the <Alt-B> keys. Verify that a line is added before the current line. Type **text of your choice**.

9. Press <Alt-D>. Verify that the line of text is deleted. Then press <Alt-R>. Verify that the line of text is restored to the screen.

10. Place the cursor at the beginning of the text that was typed in Step 7. Press <Ctrl-End>. Verify that text is deleted from the screen.

11. Press the <ESC> key to return to the *Edit Event Trees* screen. The message "**Text record not modified**" will be displayed. Press <enter> again to return to the *Edit Text* screen. A blank screen will be displayed.

12. Press <Alt-A> and type the **text of your choice**. Verify that the typed text is displayed on the screen.

13. Press the <Del> key to deleted a character. Verify that the character was deleted. Press the <Ins> key to insert a space. Verify that a space was created.

14. Press <Ctrl-Z> to return to the *Edit EndState* screen. The message "**Text record modified**" will be displayed at the bottom of the screen. Press <enter> again to access the *Edit Text* screen. Verify that the revisions made were saved. Exit and return to the *Edit Endstate* screen.

15. Type **D** to delete and place the cursor on endstate name **T1-TEST**. Press <enter> to access the *Delete Endstate* screen. Type **D** again and press <enter>. The message "**Record deleted**" will be displayed at the bottom of the screen. Verify that the name T1-TEST was deleted from the *Edit Endstate* screen. Then exit and return to the *Modify Database* menu.

16. From the *Modify Database* menu, place the cursor on the "Basic Events" option and press <enter> to access the *Edit Events* screen. A list of basic event names will be displayed on the screen.

17. From the *Edit Events* screen, type **A** for add and press <enter> to access the *Add Event* screen.

18. Type **ACP-MOV-TEST** in the space for Name. Move the cursor in the Description space. Type **This is a test** and press <enter> to return to the *Edit Events* screen. Verify from the *Edit Events* screen that the new basic event name and description were added to the database.

19. Repeat Steps 17 and 18 but type **ACP-MOV-TEST2** in the space for name. In the space for description, type **This is the final test**.

20. Type **M** for modify and place the cursor on the name called ACP-MOV-TEST. Press <enter> to access the *Modify Event* screen.

21. Place the cursor in the space for Description and type **new information**. Press <enter> to return to the *Edit Events* screen. The message "**Record modified**" will be displayed at the bottom of the screen. Verify that the descriptive information was modified.

22. Type **D** for delete and place the cursor on basic event name **ACP-MOV-TEST**. Press <enter> to access the *Delete Event* screen. Type **D** to delete and press <enter>. The message "**Record deleted**" will be displayed at the bottom of the screen. Verify from the *Edit Event* screen that the basic event name ACP-MOV-TEST was deleted.

23 Type **R** for remove place the cursor on basic event name **ACP-MOV-TEST2** to remove. Press <enter>. A warning message will be displayed on the screen. Type **Y** and press <enter>. The message "**Record deleted**" will be displayed at the bottom of the screen. Verify from the *Edit Events* screen that the basic event name ACP-MOV-TEST2 was deleted. Then exit and return to the *Modify Database* menu.

24. Place the cursor on the "Attributes" option and press <enter> to access the *Edit Attributes* menu. From the *Edit Attributes* menu, place the cursor on the "System" option and press <enter> to access the *Edit Attributes screen.*

25. From the *Edit Attributes* screen, type **A** for add and press <enter> to access the *Add Attributes* screen.

26. Type **RPS** in the space for Attribute Name. Move the cursor in the Description space. Type **This is a test** in the Description space and press <enter> to return to the *Edit Attributes* screen. The message "**Record added**" will be displayed at the bottom of the screen. Verify that the attribute name along with the description were added to the *Edit Attributes* screen.

27. Type **M** for modify and place the cursor on the attribute name **RPS**. Press <enter> to access the *Modify Attribute* screen.

28. Place the cursor in the space for Description and type **new information**. Press <enter> to return to the *Edit Attributes* screen. The message "**Record modified**" will be displayed at the bottom of the screen. Verify that the descriptive information was modified.

29. Type **D** for delete and place the cursor on the attribute name **RPS**. Press <enter> to access the *Delete Attribute* screen. Type **D** for delete and press <enter> again. The message "**Record deleted**" will be displayed at the bottom of the screen. Verify that the system name RPS was deleted from the *Edit Attributes* screen. Exit and return to the *Edit Attributes* menu.

30. From the *Edit Attributes* menu, place the cursor on the "Location" option and press <enter> to access the *Edit Attributes* screen. From the *Edit Attributes* screen, type **A** to add and press <enter> to access the *Add Attributes* screen.

31. Type **CONT** in the space for Attribute Name. Move the cursor in the Description space. Type **This is a test** in the Description space and press <enter> to return to the *Edit Attributes* screen. The message "**Record added**" will be displayed at the bottom of the screen. Verify that the location name **CONT** along with the description were added to the *Edit Attributes* screen.

32. Type **M** for modify and place the cursor on the location name CONT. Press <enter> to access the *Modify Attribute* screen.

33. Place the cursor in the space for Description and type **new information**. Press <enter> to return to the *Edit Attributes* screen. The message "**Record modified**" will be displayed at the bottom of the screen. Verify that the descriptive information was modified.

34. Type **D** for delete and place the cursor on location name CONT. Press <enter> to access the *Delete Attribute* screen. Type **D** again and press <enter>. The message "**Record deleted**" will be displayed at the bottom of the screen. Verify that the location name CONT was deleted from the *Edit Attributes* screen. Exit and return to the *Edit Attributes* menu.

NUREG/CR-6145

35. From the *Edit Attributes* menu, place the cursor on the "Failure Mode" option and press <enter> to access the *Edit Attributes* screen. From the *Edit Attributes* screen, type A to add and press <enter> to access the *Add Attributes* screen.

36. Type **FTS** in the space for Attribute Name. Move the cursor in the Description space. Type **Pump failed to start** in the Description space and press <enter> to return to the *Edit Attributes* screen. The message **"Record added"** will be displayed at the bottom of the screen. Verify that the failure mode name FTS along with the description were added to the *Edit Attributes* screen.

37. Type **M** for modify. Place the cursor on the failure mode name **FTS**. Then press <enter> to access the *Modify Attribute* screen.

38. Place the cursor in the space for Description and type **new information**. Press <enter> to return to the *Edit Attributes* screen. The message **"Record modified"** will be displayed at the bottom of the screen. Verify that the descriptive information was modified.

39. Type **D** for delete and place the cursor on the failure mode named **FTS**. Press <enter> to access the *Delete Attribute* screen. Type **D** for delete and press <enter>. The message **"Record deleted"** will be displayed at the bottom of the screen. Verify that the failure mode name FTS was deleted from the *Edit Attributes* screen. Exit and return to the *Edit Attributes* menu.

40. From the *Edit Attributes* menu, place the cursor on the "Class Attributes" option and press <enter> to access the *Edit Attributes* screen. From the *Edit Attributes* screen, type A and press <enter> to access the *Add Attributes* screen.

41. Type the number **10** in the space for Attribute Name. Move the cursor in the Description space. Type **This is a test** in the Description space and press <enter> to return to the *Edit Attributes* screen. The message **"Record added"** will be displayed at the bottom of the screen. Verify that the attribute name and description were added to the *Edit Attributes* screen.

42. Type **M** for modify and place the cursor on the class attribute name. Press <enter> to access the *Modify Attribute* screen.

43. Place the cursor in the space for Description and type **new information**. Press <enter> to return to the *Edit Attributes* screen. The message **"Record modified"** will be displayed at the bottom of the screen. Verify that the descriptive information was modified.

44. Type **D** to delete and place the cursor on the class attribute name. Press <enter> to access the *Delete Attribute* screen. Type **D** again and press <enter>. The message **"Record deleted"** will be displayed at the bottom of the screen. Verify that the class attribute name was deleted from the *Edit Attributes* screen. Exit and return to the *Edit Attributes* menu.

45. From the *Edit Attributes* menu, place the cursor on the "Basic Event Type" option and press <enter> to access the *Edit Attributes* screen. A list of attribute names with descriptions will be displayed on the screen. From the *Edit Attributes* screen, type A to add and press <enter> to access the *Add Attributes* screen.

46. Type **ABC** in the space for Attribute Name. Move the cursor in the Description space. Then type **This is a test** in the Description space and press <enter> to return to the *Edit Attributes* screen. The message **"Record added"** will be displayed at the bottom of the screen. Verify that the basic event name **ABC** along with the description were added.

47. Type **M** for modify. Place the cursor on the basic event named **ABC**. Press <enter> to access the *Modify Attribute* screen.

48. Place the cursor in the space for Description and type **new information**. Press <enter> to return to the *Edit Attributes* screen. The message **"Record modified"** will be displayed at the bottom of the screen. Verify that the descriptive information was modified.

49. Type **D** for delete and place the cursor on the basic event type name ABC. Press <enter> to access the *Delete Attribute* screen. Type **D** and press <enter> to return to the *Edit Attributes* screen. The message **"Record deleted"** will be displayed at the bottom of the screen. Verify that the basic event type was deleted from the *Edit Attributes* screen. Exit and return to the *Edit Attributes* menu.

50. From the *Edit Attributes* menu, place the cursor on the "Trains" option and press <enter> to access the *Edit Attributes* screen. From the *Edit Attributes* screen, type **A** and press <enter> to access the *Add Attributes* screen.

51. Type **TRN-A** in the space for Attribute Name. Move the cursor in the Description space. Then type **Test train A of the HPCI system** in the Description space and press <enter> to return to the *Edit Attributes* screen. The message **"Record added"** will be displayed at the bottom of the screen. Verify that the name along with the description were added to the *Edit Attributes* screen.

52. Type **M** for modify and place the cursor on the name TRN-A. Press <enter> to access the *Modify Attribute* screen.

53. Place the cursor in the space for Description and type **new information**. Press <enter> to return to the *Edit Attributes* screen. The message **"Record modified"** will be displayed at the bottom of the screen. Verify that the descriptive information was modified.

54. Type **D** for delete and place the cursor on the name **TRN-A**. Press <enter> to access the *Delete Attribute* screen. Type **D** and press <enter> to return to the *Edit Attributes* screen. The message **"Record deleted"** will be displayed at the bottom of the screen. Verify that the basic event type name was deleted from the *Edit Attributes* screen. Exit and return to the *Edit Attributes* menu. Then exit and return to the *Modify Database* menu.

55. **IF:**   Steps 1 through 54 were performed successfully

      THEN: Go to Step 56.

      ELSE: Record which step(s) was unsuccessful. Generate an anomaly report and submit it to the SAPHIRE 4.0 V&V committee.

56. End of Section 4.

F-7.6 Section 5

1. From the *Modify Database* menu, place the cursor on the "GaTe" option and press <enter> to access the *Edit Gates* screen. A list of gate names will be displayed on the screen. From the *Edit Gates* screen, type **A** and press <enter> to access the *Add Gate* screen.

2. Type **ADS-M** in the space for Name. Move the cursor in the Description space. Then type **Train A of the HPCI unavailable due to maintenance** in the Description space. Move the cursor in the space for Type. Type **AND** and press <enter> to return to the *Edit Gates* screen. The message **"Record added"** will be displayed at the bottom of the screen. Verify that the name along with the description were added to the database.

3. Type **M** for modify and place the cursor on the name **ADS-M**. Press <enter> to access the *Modify Gate* screen.

4. Place the cursor in the space for Description and type **new information**. Press <enter> to return to the *Edit Gates* screen. The message **"Record modified"** will be displayed at the bottom of the screen. Verify that the descriptive information was modified on the *Edit Gates* screen.

5. Type **D** for delete and place the cursor on the gate name **ADS-M**. Press <enter> to access the *Delete Attribute* screen. Type **D** again and press <enter> to return to the *Edit Gates* screen. The message **"Record deleted"** will be displayed at the bottom of the screen. Verify that the gate name was deleted from the *Edit Gates* screen. Exit from the *Edit Gates* screen and return to the *Modify Database* menu.

6. From the *Modify Database* menu, place the cursor on the "Graphics" option and press <enter> to access the *Edit Graphics* menu. Place the cursor on the "Fault Tree Graphics" option and press <enter> to access the *Edit Graphics* screen. A list of gate names will be displayed on the screen.

7. From the *Edit Gates* screen, type **A** and press <enter> to access the *Add Graphic Picture* screen. Type **ADS-T** in the space for Name. Move the cursor in the Description space. Then type **Train A of the HPCI unavailable due to test** in the Description space. Press <enter> to return to the *Edit Graphics* screen. The message **"Record added"** will be displayed at the bottom of the screen. Verify that the name and description were added to the *Edit Graphics* screen.

8. Type **M** for modify and place the cursor on the name **ADS-T**. Press <enter> to access the *Modify Graphic Picture* screen.

9. Place the cursor in the space for Description and type **new information**. Press <enter> to return to the *Edit Graphics* screen. The message **"Record modified"** will be displayed at the bottom of the screen. Verify that the descriptive information was modified on the *Edit Graphics* screen.

10. Type **D** for delete and place the cursor on the name **ADS-T**. Press <enter> to access the *Delete Graphic Picture* screen. Type **D** again and press <enter> to return to the *Edit Graphics* screen. The message **"Record deleted"** will be displayed at the bottom of the screen. Verify that the gate name was deleted from the *Edit Graphics* screen. Exit from the *Edit Graphics* screen and return to the *Edit Graphics* menu.

11. From the *Edit Graphics* menu, place the cursor on the "EVent Tree Graphics" option and press <enter> to access the *Edit Graphics* screen. A list of event tree names will be displayed on the screen.

12. From the *Edit Graphics* screen, type **A** and press <enter> to access the *Add Graphic Picture* screen. Type **T1** in the space for Name. Move the cursor in the Description space. Then type **Loss of offsite power** in the Description space. Press <enter> to return to the *Edit Graphics* screen. The message **"Record added"** will be displayed at the bottom of the screen. Verify that the name along with the description were added to the database.

13. Type **M** for modify and place the cursor on the event tree named **T**. Press <enter> to access the *Modify Graphic Picture* screen.

14. Place the cursor in the space for Description and type **new information**. Press <enter> to return to the *Edit Graphics* screen. The message **"Record modified"** will be displayed at the bottom of the screen. Verify that the descriptive information was modified.

15. Type **D** for delete and place the cursor on the event tree name **T**. Press <enter> to access the *Delete Graphic Picture* screen. Type **D** again, and press <enter> to return to the *Edit Graphics* screen. The message **"Record deleted"** will be displayed at the bottom of the screen. Verify that the gate name was deleted from the *Edit Graphics* screen. Exit from the *Edit Graphics* screen and return to the *Edit Graphics* menu.

16.　From the *Edit Graphics* menu, place the cursor on the "P&ID Graphics" option and press <enter> to access the *Edit Graphics* screen. A list of P&ID names will be displayed on the screen.

17.　From the *Edit Graphics* screen, type **A** and press <enter> to access the *Add Graphic Picture* screen. Type **RPS** in the space for Name. Move the cursor in the Description space. Then type "**Reactor Protection System diagram**" in the Description space. Press <enter> to return to the *Edit Graphics* screen. The message "**Record added**" will be displayed at the bottom of the screen. Verify that the name along with the description were added to the *Edit Graphics* screen.

18.　Type **M** for modify and place the cursor on the P&ID name **RPS**. Press <enter> to access the *Modify Graphic Picture* screen.

19.　Place the cursor in the space for Description and type **new information**. Press <enter> to return to the *Edit Graphics* screen. The message "**Record modified**" will be displayed at the bottom of the screen. Verify that the descriptive information was modified.

20.　Type **D** for delete and place the cursor on the P&ID name **RPS**. Press <enter> to access the *Delete Graphic Picture* screen. Type **D** again and press <enter> to return to the *Edit Graphics* screen. The message "**Record deleted**" will be displayed at the bottom of the screen. Verify that the P&ID name was deleted from the *Edit Graphics* screen. Exit from the *Edit Graphics* screen and return to the *Edit Graphics* menu. Then exit and return to the *MAR-D* menu.

21.　IF:　　Steps 1 through 20 were performed successfully

　　　THEN: Go to Step 22

　　　ELSE: Record which step(s) was unsuccessful. Generate an anomaly report and submit it to the SAPHIRE 4.0 V&V committee.

22.　End of Section 5.

F-7.7 Section 6

1.　Place the cursor on the "Select Family" option and press <enter> to access the Select Family screen. Place the cursor on the Family named EVENT and press <enter>.

2.　Place the cursor on the "REPORTS" option and press <enter> to access the *Report Type* menu. Place the cursor on the "Summary" option and press <enter> to access the *Data Selection* menu. Place the cursor on the "Summary" option again and press <enter> to access the *Output Selection* menu.

3.　Place the cursor on the "PRINTER" option and press <enter> to obtain a printout.

　　　Note: *After the printer has completed printing, the cursor will return to the Data Selection menu.*

4.　Place the cursor on the "Summary" option and press <enter> to access the *Output selection* menu. Place the cursor on the "CONSOLE" option and press <enter> to access the *Data Summary Report* screen. Compare the printout to this report. Verify that the information is consistent between the two reports. When complete, press <enter> to return to the *Data Selection* menu.

5.　Place the cursor on the "Summary" option and press <enter> again to access the *Output selection* menu. Place the cursor on the "FILE" option and press <enter>. The message "**Report completed. Summary data report written to Data.RPT**" will be displayed at the bottom of the screen. Exit from the *Data Selection* menu and return to the *MAR-D* menu.

6. Exit from the *MAR-D* menu to access the PRADATA.*B1 directory. Then access the PRADATA.B1\EVENT directory.

7. Use a text editor and access the Data.RPT file. Then compare the printout to the contents displayed in the text editor. Verify that the information is consistent between the two reports.

8. Return to the PARADATA.B1 directory, type **mard** and press <enter> to access the *MAR-D* menu. Place the cursor on the "REPORTS" option and press <enter> to access the *Report Type* menu.

9. Place the cursor on the "Summary" option and press <enter> to access the *Data Selection* menu. Place the cursor on the "Counts" option and press <enter> to access the *Output Selection* menu.

10. Repeat Steps 2 through 7.

11. From the *Report Type* menu, place the cursor on the "Detail" option and press <enter> to access the *Data Selection* menu.

12. Place the cursor on the "All" option and press <enter> to access the "Output selection" option. Then repeat Steps 2 through 7. Place the cursor on the "Detail" option and press <enter> to return to the *Data Selection* menu.

13. Repeat Steps 2 through 7 for the following options:

    1. Basic Event

    2. System

    3. EndStates

    Note: *The following steps are intended to test and verify the "Recover Database" option from the Utility option menu.*

14. Exit and return to the *MAR-D* menu. Then exit from the MAR-D program and access the PRADATA.B1 directory. Access the PRADATA.B1\EVENT subdirectory. Then type **dir *.idx** and press <enter> to obtain a list of IDX files. Press the **Print Screen** key to obtain a print-out of the files.

15. Access the PRADATA.B1 directory. Make a new directory called TESTIDX. Then access the PRADATA.B1\EVENT subdirectory. Copy all of the IDX files into the new directory.

16. Delete all of the IDX files from the PRADATA.B1\EVENT subdirectory. Then return to the PRADATA.B1 directory and type **mard** to access the *MAR-D* menu.

17. From the *MAR-D* menu, place the cursor on the "UTILITY" option and press <enter> to access the *Utility Options* menu. Place the cursor on the "Recover Database" option and press <enter> to access the *Recovery Utilities* screen. A warning message will be displayed on the screen. Type Y and press <enter> to return to the *Recovery Utilities* screen. The message "**Successful Completion**" will be displayed at the bottom of the screen.

    Note: *Invoking the "Recover Everything" option will recover all key indexes that pertain to the other options shown on the Recovery Utilities menu.*

18. Exit from the *Recovery Utilities* menu and return to the MAR-D menu. Exit from the *MAR-D* menu and access the PRADATA.BI directory. Change from the current directory to the PRADATA.B1\EVENT subdirectory. Type **dir *.idx** and press <enter>. Verify that a list of IDX files

is displayed on the screen. Compare the list of IDX files on the screen to the printout obtained in Step 13.

Note: *The following instructions are intended to electronically compare the files in the PRADATA.B1\EVENT subdirectory to the files in PRADATA.B1\TESTIDX subdirectory byte by byte.*

19.   Type **comp** and press **<enter>**. The message **"Name of first file to compare:"** will be displayed on the screen. Type the first file name from the printout and press **<enter>**. The message **"Name of second file to compare:"** will be displayed on the screen.

20.   Type **C:\PRADATA.B1\TESTIDX\FILE NAME** and press **<enter>**. The message **"Option:"** will be displayed on the screen. Then type **/a** and press **<enter>** twice. Verify that the message **"Files compare OK"** is displayed on the screen.

21.   Follow the DOS prompt and repeat Steps 19 and 20 for the remaining files listed on the printout.

22.   IF: Steps 1 through 21 were performed successfully

       THEN: Go to Step 23.

       ELSE: Record which step(s) was unsuccessful. Generate an anomaly report and submit it to the SAPHIRE 4.0 V&V committee.

23.   End of Section 6.

Appendix G

Non-Vital Features
Verification and Validation Anomaly Reports

Appendix G

Non-Vital Features
Verification and Validation Anomaly Reports

The anomaly report number is of the form

AR-VFN-MMYY-XXX

where

AR is the acronym for anomaly report
VFN is the two-digit non-vital feature procedure number
MM is the two-digit code for the month of the test
YY is the two-digit code for the year of the test
XXX is the three-digit sequential number assigned by the tester.

The test record number is of the form

IVVPRO-PN-SN-STN

where

IVVPRO stands for integrated V&V procedure
PN is the two-digit code for the procedure number
SN is the two-digit code for the section number
STN is the two-digit code for the step number.

ANOMALY REPORT

Anomaly Report No.   : AR-01-1092-001
Test Record No.   : IVVPRO-01-07-18
Date   : 10/8/92
Originator   : TWB

Source/Location: Procedure 1, Section 7, Step 18.

Description: Step 18 states that the new tree would be added to the fault tree pager screen—it was not. After getting out and back into Graphical Pager, the file was there.

Cause:

Impact: Non-critical.

Appendix G

ANOMALY REPORT

Anomaly Report No.   :   AR-02-1092-002
Test Record No.   :   IVVPRO-02-03-23,24
Date   :   10/8/92
Originator   :   TWB

Source/Location: Procedure 2, Section 3, Steps 23 and 24

Description: Changing fonts in the DESC menu: The fonts will not change the first time it is selected. It will the second time the font option is selected.

Cause: Unknown

Impact: Non-critical.

ANOMALY REPORT

Anomaly Report No.   :   AR-02-1092-003
Test Record No.   :   IVVPRO-02-09-07
Date   :   10/9/92
Originator   :   TWB

Source/Location: Procedure 2, Section 9, Step 7.

Description: Frequency remained 0.

Cause: Unknown.

Impact: Critical.

ANOMALY REPORT

Anomaly Report No.   :   AR-03-1092-004
Test Record No.   :   IVVPRO-03-01-55
Date   :   10/9/92
Originator   :   TWB

Source/Location: Procedure 3, Section 1, Step 55.

Description: The test states that, "File Saved" should appear on the bottom on the screen. This does not happen.

Cause: Unknown.

Impact: Non-critical.

ANOMALY REPORT

Anomaly Report No.   : AR-03-1092-005
Test Record No.   : IVVPRO-03-06-08
Date   : 10/12/92
Originator   : TWB

Source/Location: Procedure 3, Section 6, Step 8.

Description: Run FEP in PRATOOLS.B1 directory—will not run—error message states FEPTOP.EXE not executed. The symbol generator feature does not change files.

Cause: Unknown.

Impact: Non-critical.

ANOMALY REPORT

Anomaly Report No.   : AR-04-1092-006
Test Record No.   : IVVPRO-04-04-42
Date   : 10/13/92
Originator   : TWB

Source/Location: Procedure 4 (MAR-D), Section 4, Modify Database, Step 42, Edit Attributes, Class Attribute, Modify Attribute.

Description: Want to modify the description—the class attribute is 10—error message comes up that says class attribute name must be a number between 1 and 16 and won't modify the description—have to use escape to get out.

Cause: Unknown.

Impact: Non-critical.

.

2. TITLE AND SUBTITLE

Verification and Validation of the SAPHIRE Version 4.0
PRA Software Package

3. DATE REPORT PUBLISHED

MONTH	YEAR
February	1994

4. FIN OR GRANT NUMBER

L2483

5. AUTHOR(S)

T.W. Bolander, M.B. Calley, E.L. Capps, W.J. Galyean,
J.L. Jones, S.D. Novak, C.L. Smith, A.D. Trusty,
R.L. VanHorn, L.M. Wolfram, S.L. Zeigler

6. TYPE OF REPORT

Technical

7. PERIOD COVERED *(Inclusive Dates)*

8. PERFORMING ORGANIZATION — NAME AND ADDRESS *(If NRC, provide Division, Office or Region, U.S. Nuclear Regulatory Commission, and mailing address; if contractor, provide name and mailing address.)*

Idaho National Engineering Laboratory
Idaho Falls, Idaho 83415

9. SPONSORING ORGANIZATION — NAME AND ADDRESS *(If NRC, type "Same as above"; if contractor, provide NRC Division, Office or Region, U.S. Nuclear Regulatory Commission, and mailing address.)*

Division of Safety Issue Resolution
Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission
Washington, DC 20555-0001

10. SUPPLEMENTARY NOTES

11. ABSTRACT *(200 words or less)*

A verification and validation (V&V) process has been performed for the System Analysis Programs for Hands-on Integrated Reliability Evaluation (SAPHIRE). SAPHIRE is a set of four computer programs that the Nuclear Regulatory Commission (NRC) developed to perform probabilistic risk assessments (PRAs). These programs allow an analyst to create, quantify, and evaluate the risk associated with a facility or process being analyzed. The programs included in this set are Integrated Reliability and Risk Analysis System (IRRAS), System Analysis and Risk Assessment (SARA), Models and Results Database (MAR-D), and Fault Tree/Event Tree/Piping and Instrumentation Diagram (FEP) graphical editor.

The V&V steps included a V&V plan to describe the process and criteria by which the V&V would be performed; a software requirements documentation review to determine the correctness, completeness, and traceability of the requirements; a user survey to determine the usefulness of the user documentation, identification and testing of vital and non-vital features, and documentation of the test results.

12. KEY WORDS/DESCRIPTORS *(List words or phrases that will assist researchers in locating the report.)*

Probabilistic Risk Assessment
IRRAS

13. AVAILABILITY STATEMENT

Unlimited

14. SECURITY CLASSIFICATION

(This Page)
Unclassified

(This Report)
Unclassified

15. NUMBER OF PAGES

16. PRICE

END

DATE FILMED
5/10/94