IEEE Transactions on Nuclear Science, Vol. 43, No. 3, June 1996, pp. 1903-1909.

A Methodology and Tool for Measurement-Based Dependability Evaluation of Digital Systems in Critical Applications

Dong Tang, Myron Hecht, and Herbert Hecht
SoHaR Incorporated, Beverly Hills, CA 90211

Abstract[1]

This paper presents an overview of a measurement-based methodology for dependability[2] evaluation of critical digital systems and describes a software tool under development for it. The approach is based on measurements of operational systems and on dependability models to provide quantitative reliability and availability assessments with stated confidence levels. The methodology is described, and some of the lessons learned in its early use are discussed. The design of a software tool to implement the methodology is outlined and the current experience in applying the methodology is summarized.

I. INTRODUCTION

Both IEEE-Std-603-1991 [5] and IEEE-Std-7-4.3.2-1993 [6] include quantitative reliability criteria. The issue of the probability of failure on demand of a safety system is also a key factor in the regulatory process. However, methods and techniques for determining the dependability of digital systems which include software may not be suitable for nuclear safety systems if very high reliability requirements are imposed. Such requirements may arise as a result of the allocation of system level quantitative safety and reliability requirements to digital safety systems.

Software reliability estimation techniques [10] cannot be used in such cases because they require an execution time that will produce at least five failures for the specified reliability under conditions equivalent to operational usage in order to have a reasonable estimate of mean time between failures (MTBF) at the 95% confidence level (see Appendix A). A totally test-based approach will also not suffice. For example, in order to show a probability of failure of less than 10^{-6} with 95% confidence using system level testing, it would be necessary to run nearly 3 million independent test cases[3]. Use of only dependability models such as Markov chains [11] to analyze conditions that can lead to failure is equally unsuitable because there are no creditable data sources for estimating transition rates between states. The measurement-based approach builds on the complementary strengths of all these mentioned techniques and permits, at least under favorable conditions, creditable reliability assessments for safety systems. As such, this methodology represents a promising approach to addressing some of the design and licensing issues associated with digital safety systems.

Measurement-based analysis of computer system dependability has evolved into a mature process since the pioneer work in that area more than 15 years ago [7]. Techniques have been developed and results have been incorporated into the design and maintenance strategies of systems used in fault tolerant transaction processing [9], air traffic control [14], and other applications. However, these techniques have not yet been applied to safety grade systems in nuclear power plants. Issues of special concern in safety systems include an intermittent duty cycle for the application software and dealing with correlated and common cause failures.

This paper describes how these issues may be addressed in a combined measurement-based and modeling approach when it is applied to safety systems. Section 2 presents an overview of the methodology; section 3 provides background and discussion on using the methodology for safety systems; section 4 describes the design for a software tool currently under development; section 5 summarizes recent experience in applying the methodology; finally, section 6 concludes the paper.

II. METHODOLOGY DESCRIPTION

The measurement-based methodology was developed based on previous work in computer hardware/software failure measurement and probabilistic analysis [7] [12]. The process consists of the following steps: Data collection, Data classification, Statistical analysis, and Dependability modeling. These steps are described in the following paragraphs.

[1] This research was supported by the U.S. Nuclear Regulatory Commission (NRC) under Contract NRC-04-94-061. Mr. Robert Brill, Project Manager, provided valuable comments on this manuscript. However, the opinions and viewpoints expressed herein are the authors' personal ones and do not necessarily reflect the criteria, requirements, and guidelines of the NRC.

[2] The concept "dependability" was proposed at the *15th International Symposium on Fault-Tolerant Computing in 1985* [8]. Dependability is defined as the "quality of the delivered service such that reliance can be justifiably placed on this service." The dependability impairments are faults, errors, and failures. The means to achieve dependability is through fault avoidance and fault tolerance. Two major measures of dependability are reliability and availability.

[3] This result is based on the assumption of independence between trials (Binomial distribution). The relation is $n = \log(\alpha) / \log(1-p)$ where α is the level of significance ($1-\alpha$ is the confidence level), p is the probability of failure on demand, and n is the number of trials [14].

A. Data collection

In this step, data on failures, the system configuration, operating time, test configuration, test procedures (if applicable), and other related data are collected. These data may be available in a variety of forms (automatically generated logs, manually generated event reports, test reports, test plans, vendor data, etc.). They should contain the following information:

- Report number or event id
- Occurrence date and time
- Event duration
- System configuration, state, and operational mode at time of failure
- Failure mode (crash, hang, stop, incorrect response, late response, no response, etc.)
- Cause (hardware failure, software design or coding fault, requirements fault, operator error, configuration management error, etc.)
- Original failure event site (CPU, memory, operating system, interface, application software, etc.)
- Number of channels or elements affected
- System availability impact (no effect, successful recovery, unsuccessful recovery, extended outage)

Ideally, the failure data should be from the operational environment and for all modes in which the system will be run. However, this is not always possible. For example, safety events causing challenges to the safety systems are rare. Thus a major consideration is to make the best use of the available data from testing, operation, and from similar systems.

B. Data classification

In the data classification step, the failure data are categorized and evaluated. The classification categories and evaluation criteria are unique to each system. However, one general consideration is to distinguish between failures whose causes can be addressed through developmental or administrative controls in accordance with the quality criteria of IEEE Std 7-4.3.2-1993 (controllable failures), and those which are not controllable. The general criteria for identifying such failures include (1) a known root cause, (2) failures occurring in a process under the control of the safety system developer or the installation and maintenance process of the user, and (3) a means of determining that the failure has been mitigated[4].

On the other hand, failures in the non-controllable category can be characterized by statistical analysis and modeling. The basic assumption is that in a well tested system with a mature development process, failure and recovery are stochastic processes. That is, the predominant failure mechanisms are due to the interaction of randomly arriving inputs in a stable operational environment that interact with residual defects in the code or the hardware to cause failures. The extent to which this assumption is valid has to be verified by examining the causes identified in the failure data.

C. Statistical analysis

Once the appropriate failure data have been identified, statistical analysis methods identified in earlier work are used to provide point estimates and confidence intervals for failure rate (based on the chi-squared distribution), failure detection/ recovery probability (also called *coverage*) and probability of success upon demand (based on the binomial distribution), and to quantify probability of multiple correlated or common cause failures (correlation coefficients and conditional failure probabilities). The statistical analysis methods can also be used on the component or system level to

- construct distributions for events such as challenges to safety systems,
- understand relationships between parameters and identify correlations, and
- determine trends such as reliability growth or degradation.

D. Dependability modeling

In this step, system level dependability measures (reliability, availability, etc.) are evaluated from parameters obtained in the previous step, based on dependability models that reflect the system organization such as reliability block diagrams and Markov chains. This step can be used for

- assessing current dependability,
- identifying problem areas,
- predicting required test durations for achieving desired dependability levels, and
- performing sensitivity analyses to determine the impacts of changing parameters.

Further details on the methodology are discussed in [13]. The following section provides additional background and discussion on how the methodology can be applied to a safety system.

III. BACKGROUND AND DISCUSSION

Most previous work on measurement-based dependability assessment has been for highly available continuously operating

[4] If the failure has been mitigated and all associated faults have been identified and removed, it may be reasonable to assume that the specific relevant category of failures will not occur.

systems. Such systems can be characterized by capacity or load profiles, availability, and failure rates. Because of their continuous operation, highly available systems permit systematic studies of failure mechanisms and occurrence rates that can be used to build and validate measurement-based dependability models. In the following, the first subsection discusses differences between safety systems and other highly available systems which have been the subject of earlier research in this area; the second subsection introduces a model that can be used for some types of safety systems; the third and fourth subsections discuss approaches to estimating two important parameters: the success probability of the safety system upon demand and the probability of multiple simultaneous failures.

A. Differences between safety systems and other types of highly available systems

Safety systems differ from highly available, continuously operating systems in the following ways:

• *Both continuous and intermittent operating profiles.* Similar to highly available, continuously operating systems, safety systems also constantly monitor a set of parameters, and thus, the work on continuously operating systems is applicable. However, an additional factor in safety systems is that the operational profile of the application software for the alarming or safety mitigation function will be intermittent.

• *Importance of rare events such as common cause failures and correlated failures.* Unusual or rare events must be accounted for in the dependability predictions for critical systems. The incidence of common cause and correlated events can be measured and accounted for in the model. However, most such data will emerge from the testing environment, and determining whether the test or development environment is sufficiently similar to the operational environment is not always straightforward.

In order to demonstrate how the methodology accounts for these differences, the following subsection presents a model of a safety system. The subsequent subsections describe how the key parameters associated with the model can be estimated.

B. A safety system model

Fig. 1 shows a Markov model for a safety system such as a reactor protection system. The modeled configuration has two major components: a plant and a digital safety system which protects the plant by responding to and processing challenges from the plant instrumentation. The notation used in the model is as follows:

N Normal state in which both safety and plant are functioning within technical specifications

SP Safety processing state in which the safety system is processing a challenge

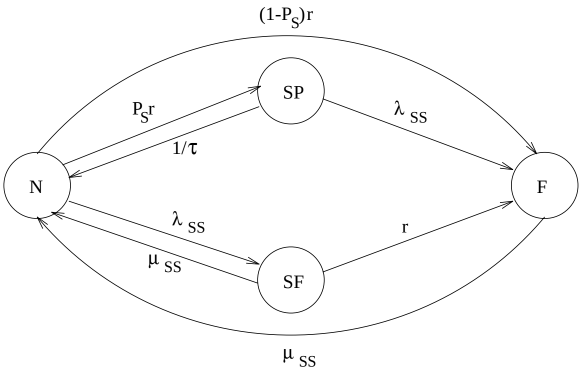

Fig. 1 A Safety System Model

SF Safety failure state in which the safety system is not available for responding to a challenge while the plant is functioning with technical specifications

F System failure state in which the safety system has failed to process a challenge

P_s Probability of success upon demand, i.e., that the safety system will be successful in responding a challenge

r Arrival rate of challenges from the plant requiring a response of the safety system

τ Challenge processing time

λ_{SS} Failure rate of the safety system system (evaluated from a submodel shown in Fig. 2)

μ_{SS} Recovery rate of the safety system after a failure

In the normal state, if a challenge arrives, the safety system will respond to it successfully with probability P_s and go to the safety processing state SP (modeled by the transition P_sr, from N to SP). During the safety processing, if the safety system fails due to its hardware or software problems, the system will fail (transition λ_{SS}, from SP to F). Otherwise, the safety system will go back to the normal state after the mean processing time τ (transition $1/\tau$, from SP to N). When a challenge arrives in the normal state, the safety system may respond to it unsuccessfully and go to the system failure state (transition $(1-P_s)$r, from N to F).

Sometimes the safety system fails in the normal state and goes into the safety failure state SF (transition λ_{SS}, from N to SF). The safety system will go back to the normal state with rate μ_{SS} (transition μ_{SS}, from SF to N). But during the recovery period in the state SF, if a challenge arrives, the system will fail (transition r, from SF to F) because the safety system is not available for use in this state. Table I shows how the safety-related concerns are accounted for in this model.

Table I
Key Issues Addressed by the Model in Fig. 1

Issue	State/Transition in Fig. 1	Remarks
Continuous and intermittent operation	Intermittent operation is modeled by N to/from SP transitions.	P_S is key parameter (see Sec. 3C).
	Continuous operation is modeled by N to/from SF transitions.	λ_{SS} is key parameter evaluated from a separate submodel.
Common cause or correlated multiple events causing safety system failure when needed	N to SF transition includes such failures.	The coverage factor in the hot-standby submodel (Sec. 3D) accounts for common cause and correlated multiple failures.
	SF to F transition models the joint event that the safety system is down and a challenge occurs.	

C. Estimating P_s

One approach to estimating the probability of success upon demand, P_s, is by means of measuring the proportion of successful test runs from test data and using the binomial distribution to determine the confidence interval. Because safety systems typically have fairly simple and well defined functions, and because these functions must generally be unambiguous and effective, their success can be described as a simple Bernoulli trial.

However, the validity of this approach is based on the assumption that the test environment is representative of the plant operational environment. This is not always possible to determine. In a safety system, it is unlikely that the system design or testing activity would not address conditions known to be likely to result in a catastrophic failure, i.e. those failures which may totally disable the safety system and are modeled as $1\text{-}P_s$ in Fig. 1.

Thus, a major question is what measurement should be made to account for catastrophic system failures, even if they have not yet been observed. It may be possible to use the results of earlier work which have shown that catastrophic failures in well-tested systems (not only digital systems) usually result from the coincidence of a number of independent conditions that are individually tolerable or, at least, non-catastrophic [2]. Some examples are documented in NUREG/CR-6293 [3]. The rate of occurrence of non-catastrophic predecessors can be often estimated based on reported failures. An estimate of the probability of a catastrophic failure can then be determined by the combined probability of two or more individual predecessor events. Whether the combined probability is a simple product of two individual probabilities or another function of both individual probabilities and correlation parameters can be determined by a correlation analysis.

D. Estimating λ_{SS}

The parameter λ_{SS} is the failure rate of the safety system hardware and software due to either multiple simultaneous random failures or due to a single common cause event. Estimation of this parameter is incorporated into a lower level model, or submodel. Fig. 2 shows an example of such a submodel for a system consisting of two channels: a primary (or "hot") channel, and a standby channel.

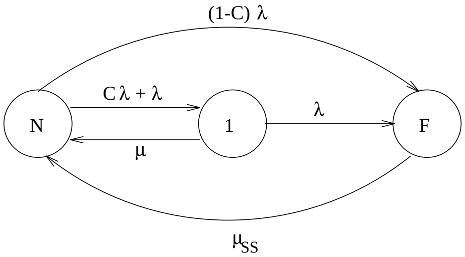

Fig. 2 A Markov Model for a System with a Standby Channel

The notation used in the model is as follows.

N Normal state in which both primary and standby channels are functioning

1 State in which one channel has failed and another channel is functioning

F Failure state in which both primary and standby channels have failed. This state is equivalent to SF in Fig. 1; the total transition rate from N to F represents λ_{ss} in Fig. 1.

λ Failure rate of a channel

μ Recovery rate of a channel

μ_{SS} Recovery rate of the safety system after a failure

C Probability that switchover to the standby will be successful when the primary channel fails. The complement of this quantity, or 1-C, is the probability of a common cause or correlated failure at the system level that defeats both channels.

In the normal state, either the primary or the standby channel can fail. If the primary channel fails, the system will switchover to the standby successfully with a probability C (coverage). This is modeled by a transition from state N to state 1 with rate $C\lambda$. If the standby channel fails, the system will also go to state 1, but with a transition rate λ because no switchover will be involved. Thus, the aggregate rate of the transition from state N to state 1 is $C\lambda+\lambda$. Note that this rate represents single channel failures and does not indicate any failure of multiple channels. If C=1, the aggregate rate would be 2λ, which implies failure independence (i.e., single channel failures arrive at rate 2λ and will not cause a second failure).

During the recovery of the failed channel (state 1), if another channel also fails (independently), the system will fail, which is modeled by the transition from state 1 to state F. There is a possibility, 1-C, that the switchover to the standby will not be successful, which leads the system directly to the failure state F. This is modeled by the transition from state N to state F with rate (1-C)λ. After a system failure, the system will recover from the failure at a rate μ_{SS}, modeled by the transition from state F to state N marked by μ_{SS}.

Other models would be appropriate for different redundancy schemes. For example, in a four-channel reactor trip system, a 2-out-of-4 reliability block diagram model could be used to handle multiple independent randomly occurring failures. Common cause and correlated failures could be modeled by Markov models similar to that in Fig. 2. The output of these models could then be incorporated into the higher level Markov model shown in Fig. 1.

IV. A SOFTWARE TOOL DESIGN

A modeling tool called MEADEP (MEAsure DEPendability), is being developed to implement the data classification, analysis, and dependability modeling described in Section 2. Fig. 3 outlines the tool design.

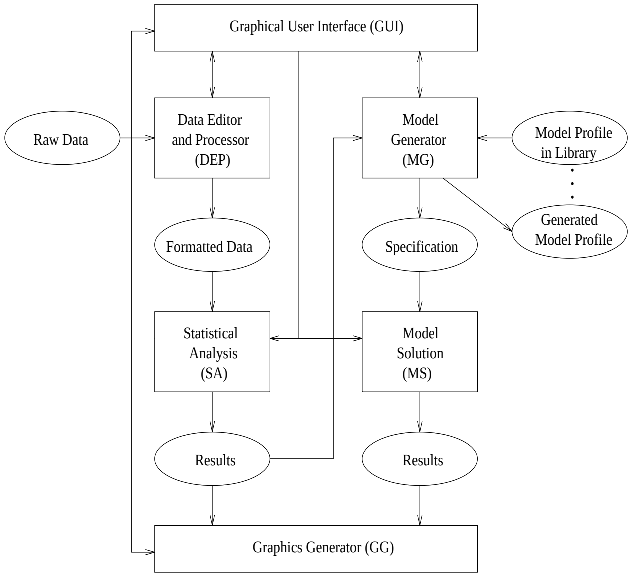

Fig. 3 Top Level Data Flow View of MEADEP

In the figure, rectangles represent software modules and ellipses represent input or output files. The Data Editor and Processor (DEP) module, interacts with the user to process raw data and generate formatted data. The raw data can be manually generated structured trouble reports or computer generated event logs. The Statistical Analysis (SA) module computes measures from the formatted data. Some of these measures will be parameters for use in the dependability models. The Model Generator (MG) module generates model specification files for the subsequent model solution. The specification can be generated from a model profile predefined in a library, or created by the user. The Model Solution (MS) module generates results based on the model specification and parameters obtained from the data by SA. The purpose of the Graphics Generator (GG) module is to generate visual graphs from numerical results, and it is integrated with the Graphical User Interface (GUI).

The raw data formats to be supported by DEP include structured trouble reports and event logs. The conversion is guided by a map that must be supplied for each raw data format by the user but that can be stored for repeated use. The user is also allowed to generate internal data manually by typing in each field requested by DEP. This option will be useful when handwritten operator logs are to be used. The internal data can be viewed and edited by the user. The internal data format generated by DEP will consist of the fields discussed above.

The MG module is similar to the DEP module in that it is also used for generating and editing files. But rather than a data file, the output of MG is a specification file for directing the model solution performed by the MS module. During a modeling session MG will ask the user a series of questions about details of the model and parameters. If a model is developed without assigned parameter values, a model profile will be created and added into the model profile library. Later, the user can generate a specification file when parameters are available. Parameters can be entered manually, or extracted from another file. If parameters are already obtained from data by the SA module, MG can open the output files produced by SA and extract parameters under guidance of the user. After MG obtains all required parameters for a defined model, a specification file will be generated.

A library of model profiles will be an important part of MEADEP. A model profile defines the structure of a dependability model for a particular system, but does not contain the parameter values. To generate the corresponding specification file, the required parameter values must be specified, either manually or from output files of the SA module. Finally, the model solution routines in the MS module solve the model. The MS module provides solutions to the dependability models described in the specification file. The library frees users in nuclear utilities and other critical areas from the need to develop structured models. The library will be expandable to accommodate likely dependability models of interest to customers.

All interactions between the user and the software modules discussed above are through the GUI module and are controlled by a central scheduler. The interface is menu-driven with extensive prompt and help information. Upon an input from the user, the central scheduler switches control to the appropriate software module. Numerical results from the SA and MS modules can be converted to visual graphs (e.g., reliability or availability curves) by the GG module.

V. CURRENT EXPERIENCE WITH THE METHODOLOGY

This section summarizes experience in using the methodology in our preliminary research [13]. Three systems were investigated: an integrated plant control system under development, a plant protection system during early operation, and an air traffic control system under development.

A. Integrated Control System (ICS)

The ICS data were gathered during the development and an early test phase at a plant simulator. Because this was a system under development, there was no sufficient operational time during which failure rate data could be gathered and assessed. However, reports on hardware and software problems found during testing were available for this analysis. Thus, the primary focus of the data analysis was in the data filtering and classification step, i.e., to understand how to filter data and understand the qualitative aspects of the failure process. Specifically, during the first six months 15 of the failures encountered were due to fairly obvious faults in algorithms and data (parameters). During the second six months there were no further failures due to these causes, but 14 of the reported failures were due to problems in timing and diagnostics. These are more subtle fault types. This qualitative classification process showed that although the number of failures in these types reported in the two 6-month periods were nearly identical, there was a measurable reliability improvement because the system remained operational for the time necessary to create the conditions which resulted in these more subtle failures. By way of contrast, an exclusively quantitative assessment would have concluded that there was no measurable reliability growth.

B. Plant Protection System

Data were collected during the first operational period of Westinghouse Eagle 21 systems at the Sequoyah Nuclear Plant [15]. In contrast to the ICS case, the system was operational and there were sufficient failure data and operating time to determine confidence limits on failure frequencies and failure rates for different categories which were used for a detailed analysis. The reliability growth of the two measured units was shown to be significant from the first year to the second year with a stated confidence level. However, the data which could be used to determine P_s were sparse because of the lack of safety events (challenges) in the operational environment. Thus, additional test data would be needed in order to provide a sufficiently large sample from which this parameter could be calculated. Additional data from the operational environment would also be needed to provide baselines for an objective translation of these test results to later phases by collecting and classifying the naturally occurring failures.

C. Air Traffic Control System

Data from an air traffic control (ATC) system undergoing stability testing were used successfully to estimate model parameters [14] and to evaluate system availability. The key to this success was the existence of an automated data collection capability and an adequate amount of accumulative operational time (thousands of processor hours). These characteristics permitted estimation of failure rates of basic software units and the coverage of fault tolerance provisions for use in the software availability models similar to that shown in Fig. 2. The models were developed from the task level for the ATC software. The data and models were used to assess system availability, to identify key problem areas, and to predict required test durations for demonstrating desired levels of availability.

VI. CONCLUSION

This paper presented an overview of a methodology for measurement-based evaluation of operational systems. The methodology consists of feasible and defensible methods in data collection and processing, statistical analysis, and dependability modeling. It assesses system dependability based on data with a reasonable size and allows evaluation of system dependability at a specified confidence level. A tool called MEADEP is being developed to allow non-experts to collect data and evaluate system dependability.

However, additional work on the methodology is necessary in order to develop approaches in dealing with non-deterministic failures introduced by controllable factors in the development process. Additional case studies are necessary in order to gain more experience and improve the library of models that will be included with the MEADEP tool.

VII. APPENDIX A — SAMPLE SIZE FOR MTBF

Assume that the estimated MTBF follows an exponential distribution with a mean of 1/λ. By [4], the required sample size can be determined by

$$n = \frac{z^2_{\alpha/2}\ \sigma^2}{\varepsilon^2} \qquad \textbf{(1)}$$

where ϵ^2 is the allowed error, σ^2 is the variance of the MTBF, and α is the desired significance level ($1-\alpha$ is the confidence level). The maximum symmetrical interval for $1/\lambda$ is $[0, 2\times 1/\lambda]$. That is, the maximum allowed error is $1/\lambda$. Thus, we substitute ϵ in the above equation with $1/\lambda$. Since the distribution of the MTBF is exponential, we have $\sigma = 1/\lambda$. The above equation is then reduced to

$$n = Z^2_{\alpha/2} \qquad \mathbf{(2)}$$

If the desired confidence level is 95%, $Z_{\alpha/2}$ is 1.96 and *n* is approximately 4. In order to have four MTBF instances, five failures are required (to construct four intervals).

VIII. REFERENCES

[1] H. Hecht and M. Hecht, *Reliability, Testability and Design of Fault Tolerant Systems*, RADC-TR-84-57, Rome Air Development Center, April 1984.

[2] H. Hecht and P. Crane, “Rare Conditions and Their Effect on Software Failures,” *Proceedings of Annual Reliability and Maintainability Symposium*, Anaheim, CA, pp. 334-337, Jan. 1994.

[3] H. Hecht, M. Hecht, G. Dinsmore, *et. al.*, *Verification and Validation Guidelines for High Integrity Systems*, U.S. Nuclear Regulatory Commission and Electric Power Research Institute Report NUREG/CR-6293, March, 1995.

[4] R.V. Hogg and E.A. Tanis, *Probability and Statistical Inference*, 2nd Ed., Macmillan Publishing Co., New York, 1983.

[5] IEEE, *Standard Criteria for Safety Systems for Nuclear Power Generating Stations*, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855, USA.

[6] IEEE, *IEEE Standard for Digital Computers in Safety Systems of Nuclear Power Generating Stations*, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855, USA.

[7] R.K. Iyer and D. Tang, “Experimental Analysis of Computer System Dependability,” in *Fault Tolerant Computer System Design*, D.K. Pradhan (ed.), Prentice-Hall, 1995.

[8] J.C. Laprie, “Dependable Computing and Fault Tolerance: Concepts and Terminology,” *Proceedings of the 15th International Symposium on Fault-Tolerant Computing*, pp. 2-11, June 1985.

[9] I. Lee and R.K. Iyer, "Software Dependability in the Tandem GUARDIAN System,” *IEEE Transactions on Software Engineering*, Vol. 21, No. 5, pp. 455-467, May 1995.

[10] J. Musa, A. Iannino, and K. Okumoto, *Software Reliability: Measurement, Prediction, Application*, McGraw-Hill Book Co. 1987.

[11] R. A. Sahner and K. S. Trivedi, "Reliability Modeling Using SHARPE", *IEEE Transactions on Reliability*, vol 36, pp. 186-193, February 1987.

[12] D. Tang and R. K. Iyer, “MEASURE+ — A Measurement-Based Dependability Analysis Package,” *Proceedings of the ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems*, Santa Clara, CA, pp. 110-121, May 1993.

[13] D. Tang, H. Hecht, Measurement-Based Dependability Analysis for Critical Digital Systems, SBIR NRC-04-94-061 Phase I Final Report, SoHaR Inc., May 1995.

[14] D. Tang and M. Hecht, “Evaluation of Software Dependability Based on Stability Test Data,” *Proceedings of the 25th International Symposium on Fault-Tolerant Computing*, Pasadena, CA, pp. 434-443, June 1995.

[15] TVA Letter to NRC Dated May 10, 1990, *Sequoiyah Nuclear Plant (SQN) — Eagle 21 Functional Upgrade Commitments*, NRC Public Document Room, Accession #910715001.