Fault Tolerant System Design in the Concept Exploration Stage of a Mission Critical Computing System

Abstract—As the DoD enters a new era of weapon systems procurement it faces the critical question of how to manage and procure dependable and cost effective, mission critical computing systems. Program offices must, within the extreme time pressures of modern weapon systems development, apply fault tolerant systems design principles early in the design cycle before major resources are committed to a particular systems architecture. This paper describes the dependability evaluations and trade studies that should be done in the concept exploration stage of a mission critical computing system. These studies are considered from the point of view of a government program office in charge of the RFP requirements and Statement Of Work (SOW) design review and milestone deliverables. This paper will describe a dependability paradigm that involves the interplay between the analysis of field failure data, analytic and functional modeling and fault injection experiments. The paper will then outline the dependability requirements and evaluation criteria for the System Requirements Review (SRR) and System Design Review (SDR) that flow from this paradigm.

TABLE OF CONTENTS
1. BACKGROUND
2. CRITICAL CONCEPT EXPLORATION STAGE EVALUATIONS
3. A DEPENDABILITY PARADIGM
4. FUNCTIONAL SIMULATIONS and ANALYTIC MODELING
5. SRR (SYSTEMS REQUIREMENTS DESIGN REVIEW)
6. SRR DELIVERABLES
7. SDR (SYSTEM DESIGN REVIEW)
8. SDR DELIVERABLES
8. CONCLUSION

1. BACKGROUND

The DoD is entering a radically new era of weapon systems procurement. The Cold War is over and new scenarios and radical force structure changes are altering the requirements of the next generation of weapon systems. Also, budgets are decreasing, development cycles are being stretched out, prototypes and demonstrations are being emphasized, open systems standards and Computer Off-The-Shelf (COTS) parts will dominate, and the next generation of systems will be intensively modeled and simulated before any hardware is...
built. Future complex weapon systems will increasingly rely on digital systems and the dependability of these digital systems will play a critical role in the effectiveness of those systems in the field. The frequency of hard fault rates is decreasing for both military and commercial digital parts but transient fault rates are increasing. The next generation of systems needs to automatically handle transient faults and notify the operator only when recovering from hard faults, yet record all fault activity to ensure quick diagnosis of failing parts.

A dependable computing system is one that reliably delivers its expected and specified service. The two methods of achieving dependability are fault avoidance and fault tolerance. [1] Fault avoidance is achieved by adequate testing at each stage of a system's development. [2] Fault tolerance is a designed-in attribute that enables an operational system to deliver its expected service even when faults are manifested in the system.

The Rand Corporation in 1988-89 studied the military's process of acquiring weapon systems and concluded: (1) there was no meaningful set of management measures for Reliability and Maintainability (R&M), (2) there was no adequate process for setting rationally based R&M goals for future systems, and (3) there was no strong assurances that needed levels of R&M would be delivered in the systems then under development. The Rand report concluded that R&M has many dimensions, (it is not an easily quantified attribute) and that the elusive quality of R&M will grow more illusive as system complexity grows. [3,4]

In response to the Rand study the Navy, in coordination with the Air Force, set up a number of programs and working groups to clarify the management milestones and metrics for the design of dependable mission computer systems. Two key efforts were the Advanced Avionics Subsystem Technology (AASST) Fault Tolerant Demonstration and the tri-service Dependability Working Group (DWG). The AASST Fault Tolerant Demonstration set out to clarify the Navy's fault tolerant avionics specifications, validation methods and acceptance tests. The program demonstrated clear and precise fault tolerant requirements for specifications and SOW validation metrics and acceptance tests from initial concept exploration to the brassboard Demonstration and Validation of the system (see Figure 1) as the government oversees the design and procurement of a modern, complex, computer based weapon system. The DWG was composed of leading researchers in the fault tolerant community and leading industry developers of fault tolerant computer systems who, under the leadership of the DoD, addressed the topic of the dependability validation of mission critical computer systems. The goal of the DWG was to coordinate industry and the research community to come to a consensus on: a) the specifications of the key dependability factors in computer based weapon systems, and b) the necessary and sufficient dependability validation criteria for these systems. This paper is based on the findings and documentation resulting from these efforts and their initial application in selected Navy and Air Force platforms.
This paper will give a brief overview of a fault tolerant system design methodology, the paradigm the methodology is based on, and specific deliverables that flow from this methodology that would be included in the first two major design reviews - the System Requirements Review (SRR) and System Design Review (SDR). (Figure 1) Costs.[5] By milestone I most of the decisions that will determine these costs have been made, see the top graph of Figure 1, while less than 5% of the Life Cycle Cost (LCC) has been spent, the lower graph of Figure 2. Thus the cycle of evaluations, candidate designs, analysis and trade studies done in the
conceptual evaluation phase of a mission critical computer systems is the critical phase in the system's life cycle cost benefits analysis.

For a program office to successfully oversee the development of a fault tolerant, mission critical, computer system they must specify adequate dependability and fault tolerant metrics (SOW exit criteria) and follow a rigorous and thorough fault tolerant design methodology that serves as the basis of the government and contractor interactions throughout the system’s development. The government does not design the mission critical computer system or give direct supervision of the ongoing work. The government's role is to specify and validate the work. The SOW contains the governments specifications and the validation testing that will be performed. The system Specification requirements and the SOW will call for specific information as deliverables in reports and demonstrations throughout the design and development process. The design review meetings will focus on this deliverable information to ensure that the contractors have considered all relevant trade offs for dependability and fault tolerance. The courts have generally held any ambiguities in the SOW against the government. Thus, for the government to ensure a dependable computing system it is essential to have accurate dependability specifications and validation methods and for the government and the contractor to have a clear, agreed upon, model of the system design methodology that will be followed.

A mission critical computer system development methodology is a collection of methods (the methods being some disciplined process for generating the models of the system) that apply a unified approach across the development life cycle of the system. This collection of methods serve a number of purposes: (1) they instill an ordered discipline into the development process that ensures that each element of the problem will be incrementally addressed, (2) they provide a common language for all members of the development team, and (3) they provide management with clear milestones to measure the progress of the system's development. [6] Failure to manage the complex development of a mission critical computer system with some disciplined approach is the cause of expensive cost overruns, patching and repairing a system late into its life cycle, and the eventual failure of the fielded system to fully perform up to its specified performance.

The design decisions of the concept exploration stage are going from a definition of the algorithms, capabilities and services of the system, to exploring initial requirements and architectures, to selecting candidates and testing them on a variety of analysis tools, and ending with the selection of a particular architecture and system design approach. The rationale for these choices should be provided in the deliverables of the SDR.

3. A DEPENDABILITY PARADIGM

"A model is an abstraction of something for the purpose of understanding it before building it" [7] It omits nonessential details and is thus easier to manipulate than a complete system. A model extracts certain aspects of the system for closer examination. The first models of the new system are either analytic, mathematical models, or functional, behavioral models. In analytic modeling the first models of the new system are based on the observations of past similar systems and components. From this measured data of past similar systems the developer formulates a theory that aims at explaining the behavior of the current, proposed system, and attempts to forecast
that future system's behavior. This theory is the basis of the analytic model. The model is a check on the theory in practice. Laboratory fault injection experiments on the early prototypes or behavioral models of the system will further validate and refine the analytic model. The analytic model will also reveal which system parameters are most sensitive to change and need closer laboratory examination, fault injection experiments or field measurements. Figure 3 shows this overall flow and feedback from field data, to analytic model, to fault injection experiments.

![Figure 3 - Dependability Paradigm](image)

4. FUNCTIONAL SIMULATIONS and ANALYTIC MODELING

Analytic modeling and fault injection experiments are two complementary streams of dependable system development. The fault injection experiments are performed on the functional simulation of the system that develops from the high level conceptual model of the system and is gradually transformed from behavioral descriptions, to structural descriptions, to the final brassboard implementation. The dependability examination of this transforming stream of simulations is fault injection. Fault injection is available at each stage of the system simulations and into hardware breadboard and brassboard stages.

Due to the vast number of possible failure modes of a system, fault injection can never completely validate a dependable design. Functional simulation and fault injection is thus complemented by analytic modeling. Analytic modeling provides insight into the fault injection experiments and helps determine which faults to inject, at what level of the system, and at what system states. The fault injection experiments also feedback and calibrate the parameters of the analytic modeling. Thus the results of both streams are constantly being reconciled and the estimates of dependability are becoming more accurate as the design approaches a brassboard implementation. Carefully planned fault injection experiments in the brassboard are the final proof of the system's dependability.
The basic modeling elements of analytic modeling can be divided into: 1) inputs to the probability of the next state of the system. (Markov models are good for sequence dependent failures and repairable systems.)

The system behavior modeling elements can be combinatorial or noncombinatorial. The combinatorial elements are reliability graphs, predicting the hard failure rate of the current system from the statistical past failure rates of similar components, and fault trees, trees of conditions that lead to certain system failures. Fault trees describe the hierarchical flow of faults that will trigger certain critical system faults. Fault trees come later in the design process as they are heavily dependent on a detailed knowledge of the system. The noncombinatorial modeling elements are Markov Chains and Petri Nets. A Markov Chain is a graph of failure states and probability of transitions that determine the probability of the next state of the system. Petri Nets, directed graphs of place and transition nodes, are useful when the limits of Markov Chains are reached. Petri Nets can be used to examine inherent concurrencies such as coincident faults (a second fault arriving while the system is recovering from the first fault). Monte Carlo simulations are useful as more system reality is put into the model and the analytic solutions become intractable. [8]

This cycle of field data to analytic modeling to functional simulation and fault injection experiments is the necessary dependability validation paradigm that forms a framework for the information to be provided to the government in the SRR, SDR and subsequent design reviews.
5. SRR (SYSTEMS REQUIREMENTS DESIGN REVIEW)

The object of the SRR is to specify the initial computing, and dependability models of the system. The application of the system drives of the need and the desired result that will satisfy the need, start the process. This specification should include a clear description of the services, algorithms, system responsibilities, and constraints of the system.

The contractor derives a performance model from the specifications and begins the analysis
of the specification in light of the
dependability requirements of the system.
The system performance model and
dependability models are developed and
analyzed in parallel with constant feedback
between them. (Figure 5, taken from [9]).
These models will be refined in subsequent
phases and are used to make the architectural
tradeoffs. Figure 5 shows a global view of
this modeling and feedback between the initial
performance and dependability models of the
system. (The application is from mission
avionics where the key scenario/specification
system drivers are: mission phases, target
tracking, engagement of targets and planning
for the next targets.)

The design of a dependable mission critical
computer system begins with a mission
scenario and the associated sensor processing
requirements (the top bubble in Figure 5).
Added to these requirements would be the
security and fault tolerant system
requirements. After a coarse workload
description, a coarse performance model is
derived and an initial architecture description
is derived and an interaction between the
coarse performance and dependability models
would begin until these initial
performance/architecture and dependability
requirements are met. This combined
modeling of performance and dependability
continues throughout the design cycle. Even
at the earliest stages of a systems
development, reliability modeling can be used
to analyze parametric sensitivities in order to
determine which factors have the strongest
impact on the systems reliability. [10]

The range of possible dependability
parameters is wide and a reliable validation of
them is an open question. To ensure that the
delivered system is dependable and fault
tolerant the government must be informed
customers, active in the evaluation of the
current models of the design, and not merely
passive customers using broad and vague
language to describe the system’s
requirements. Typical dependability
requirements for the system might include:
mission life, reliability, readiness or
availability goals, identification of critical
information to be protected, recovery times,
requirements for testing, maintenance and
checkout, minimum performance levels, safety
goals, and subsystem fault containment
regions. The modes of service should also be
specified: full, reduced, degraded, emergency,
and safe shut down. And a generic set of
faults should be identified according to timing
(how often), duration (how long), and extent
of damage, [11, 12] and some indication of
how the final design will be validated against
these faults should be given. Choosing a
reasonable set of trade-off criteria and fault
tolerant metrics is critical to ensuring that the
evolving system is capable of being verified
analytically and experimentally at each stage
of the design process.

The SRR lays the groundwork for establishing
a chain of traceability for the contractor’s
design decisions in subsequent reviews. The
RFP/SOW packages should specify a careful
series of checks on the design development
that track the contractor’s progress and
validate that the resulting system will perform
its specified services within its functional
bounds (specified fault environment).

The goal of this stage is to begin the design
implications of the high level dependability
requirements that distinguish between broad
classes of design. As the dependability
models mature they will produce a finer
discrimination between designs. The
qualitative analysis will become more through
and the quantitative analysis more exact.

In addition to the exploratory work of this
phase a preliminary implementation plan
should be specified. The reliability aspects of
this plan should include: Computer Aided Engineering (CAE) and Computer Aided Design (CAD) tools and reliability prediction tools that will be used.

6. DELIVERABLES of SRR

(These deliverables are a partial list, with a focus on the dependability information.)

The government, in this phase, will assess the contractor’s knowledge of the problems being addressed, their knowledge of the algorithms needed and their ability to use this knowledge to make the trade-offs and justify the architecture that will be chosen in the second phase of the conceptual exploration.

A list of deliverables with a special focus on dependability are:

1. Computational Requirements Report
   This report includes the following:
   - Computing model
   - Baseline Application Algorithms
   - Properties of the Algorithms
   - Throughput
   - Memory Requirements
   - Information and Control Flow
   - Results of Software Simulations of the Baseline Algorithms
   - A Sensitivity Analysis With some prediction of Expansion Needs

2. Dependability Report

   The focus of this report is the dependability features of the proposed system. It should contain the dependability requirements: mission life, reliability, critical state, recovery time, testability, safety goals, availability, and modes of service.

   The dependability requirements should flow down from the mission of the system to justify the particular dependable design strategies that are being considered. There should also be a list of the hardware and software fault types that the system will be designed to protect against. This report should also contain the rationale for the trade-offs and criteria for selecting the fault handling techniques. This methodology will be used in the phase II trade-off studies and selection of the architecture and design approach.

3. Implementation Strategy Report

   This report outlines the management plan for the selection and implementation of the selected architecture.

4. Review and Evaluation Criteria at the System Requirements Review

   The questioning at the design review should probe the work done so far with questions like:
   - Are the specified computer requirements consistent with the mission requirements?
   - Does the contractor have the adequate experience base, tools, and methodology to select the best architecture and implement it?
   - Is the fault set complete and accurate and the methodology for testing against them adequate?
   - Is the computational model complete and accurate?
   - Are the candidate architectures of a high quality and do they have serious potential to meet the objectives?
   - Is the management plan complete, including the plans to acquire the tools and develop the needed technology?
7. SDR (SYSTEM DESIGN REVIEW)

The objective of the SDR is to conduct trade-off studies on the alternative architectures and select the architecture that best meets the requirements. The dependability modeling at the SDR stage is exploring the high level dependability requirements. Thus for each potential architecture a preliminary functional design and preliminary dependability modeling should be used to evaluate the architecture against the system dependability requirements.

The selected architecture should have a fuller evaluation and justification, including a block diagram partition and allocation of the computing functions, identification of the fault types, containment approaches, recovery mechanisms, redundant sparing needed, assumptions made in the analytic modeling, and executive strategies for recovery of critical state information.

Validation is the demonstration that a given system meets its specification. This validation in the concept exploration consists of the quantitative and qualitative evaluation of the selected candidate designs.

The quantitative evaluation of a design is done by using reliability prediction tools (HARP, SURE, etc.) and fault injection. Fault injection and reliability prediction research is of vital concern to the government. Currently there is no agreed upon way to validate a complex computing system. The FAA uses a combination of Failure Modes and Effects Analysis (FMEA), fault tree analysis and pin level fault injection on the actual hardware. [13] Research is ongoing with a wide variety of fault injection methods and levels of injection: hardware pin faults and gate level injection [14,15], hardware pin faults with gate at the Register Transfer (RT) level [16,17,18], hardware pin faults at the bus RT level [19], simulation at the opcode RT level [20], simulation and fault injection at the gate and chip level [21], fault emulation at the memory word and RT level [22], fault emulation at the RT level [23], memory corruption at the RT level [24], and ion radiation at the device level [25].

At the concept exploration stage the architectures are defined at a functional level. Fault injection in the concept exploration would better be characterized as failure injection in the functional descriptions of the architectures being considered. Simulations at this stage are experiments with high level architectural descriptions of the proposed system. This description is a compromise between the full, complex, description of the system and the fault handling features of the system that is the focus of the simulation.

The goal of the qualitative evaluation of the fault tolerant design in the concept exploration and subsequent stages of the design is to evaluate the completeness of the design and its potential to meet the stated goals. This qualitative evaluation comprises a series of probing questions at design reviews or working group meetings before the formal design review. [26] This type of examination of the system requires expert knowledge. The reviewers should question such things as: the fault assumptions, man machine interaction faults, design faults, the occurrence of a fault while the system is still in the process of recovering from the first fault, and faults in the fault tolerance functions themselves.

The Air Force (using the services of The Aerospace Corporation, a Federally Funded Research Center (FFRC)) has incorporated a successful "expert team" approach to the qualitative evaluation of proposed fault tolerant designs for a number of critical satellite projects and in the SDIO BM/C3 project. [27] This expert team was comprised
of leaders in the field of fault tolerance that the government asked to meet with the contractor and evaluate the design in informal working group meetings. These experts in the field, comprised of university researchers, technical experts from FFRCs and industry, provide the necessary team synergy to give the government an excellent qualitative evaluation of the proposed design. Their comments and criticisms are aimed at improving the design and ensuring that the future government owned mission critical computer system is state of the art and not simply the product of the contractor's in-house efforts. This expert knowledge base of the team can be applied at every stage of the development but is especially important in the conceptual exploration down-select of the final design approach and architecture.

8. SDR DELIVERABLES

Some dependability deliverables for SDR might include:

1. Progress Report on Tools and Technology

This report describes the development and status of the CAE/CAD, and reliability prediction tools (e.g./ HARP, SURE, Heiress SHARPE, Save).[28, 29]

2. Architecture Tradeoff Report

This report presents the rationale behind the choice of the architecture chosen. It lists the selection criteria, describes the alternative architectures, the evaluation of those architectures using the analysis and reliability tools, and a definition and evaluation of the architecture chosen. The fault tolerant features of this full definition of the selected architecture should include:

(1) a block diagram of the architecture with the allocation of applications processing to partitions and the allocation of redundant sparing;
(2) error detection mechanisms and error containment strategies used in each partition;
(3) a description of the error and fault recovery mechanisms and estimates of their effectiveness;
(4) the software executive strategy of preserving critical information during fault recovery;
(5) design features to support the initial testing and periodic diagnosis and repair.

9. CONCLUSION

This paper presented a description of a dependability paradigm and its ramifications in the concept exploration stage of a mission critical computing system. The purpose of the paper was not to give an exhaustive list of the government evaluations of a mission critical computer system in the concept exploration stage but an outline of those evaluations when a paradigm similar to the one presented in this paper is the agreed upon design and validation process of the system. The paradigm described is one of constant checks between field data, analytic modeling and experimental fault injection on the functional system (simulation to brassboard). For successful oversight of the development of a fault tolerant, mission critical computer system, the government must specify adequate dependability and fault tolerant metrics and with the contractor follow a rigorous and thorough fault tolerant design methodology that serves as the basis of the government and contractor interactions throughout the system's development. The government does not design the mission critical computer system or give direct supervision of the ongoing work. The government's role is to specify and validate the work. But this specification and validation
should be done as informed customers, active in the evaluation of the current models of the system's design, and as cooperative members of the integrated team that is developing the complex weapon systems of the future.

REFERENCES

[1] A. Avizienis and J-C. Laprie, "Dependable Computing: From Concepts to Design Diversity," in Proceedings of the IEEE, Vol. 74, No. 5, May 1986, pp. 629-638.

[2] D. Siewiorek and L. K. -W. Lai, "Testing of Digital Systems," in Proceedings of the IEEE, Vol. 69, No. 10, October 1981, pp. 1321-1333.

[3] J. R. Gebman, "Needed reforms to the Defense System Management of Avionics Acquisition and Support," The RAND Corporation, 1700 Main Street, P.O. Box 2138, Santa Monica, CA 90406-2138.

[4] J.R. Gebman and H.L. Shulman, "A Strategy for Reforming Avionics Acquisition and Support," July 1988, A Project Air Force report prepared for the United States Air Force R-2908/2-AF, The RAND Corporation, 1700 Main Street, P.O. Box 2138, Santa Monica, CA 90406-2138.

[5] Defense System Management College, Systems Engineering Management Guide, January 1990, U.S. Printing Office.

[6] G Booch, Object Oriented Design With Applications, The Benjamin/Cummings Publishing Company, Inc., Redwood City, California, 1991.

[7] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, W. Lorensen, Object-Oriented Modeling and Design, Prentice Hall

[8] A. Johnson, M. Malek, "Survey of Software Tools for Evaluating Reliability, Availability, and Serviceability", ACM Computing Surveys, Vol. 20, No.4, Dec. 1988

[9] C.O. Scheper, S.C. Johnson, R.L. Baker, "Modeling Methods for the Design and Evaluation of Fault Tolerant Systems," Digital Avionics Systems Conference, October 1991

[10] J.B. Dugan, S.J. Bavuso, M.A. Boyd, "Dynamic Fault Tree Models for Fault Tolerant Computer Systems," NASA Tech Report

[11] SDIO BM/C3 Processor and Algorithm Working Group, Algirdas Avizienis, ed., "Volume One, Design of Fault-Tolerant Systems," Application of Fault Tolerance Technology, October 20, 1987.

[12] SDIO BM/C3 Processor and Algorithm Working Group, D.A. Rennels, ed., "Volume Two, Management Issues: Contractor Milestones and Evaluation," Application of Fault Tolerance Technology, October 20, 1987.

[13] H. P. Curd, "Integrated Assurance Assessment," in Digital Systems Validation Handbook - Volume II, Technical Report DOT/FAA/CT-88/10, Computer Resource Management Incorporated, 950 Herndon Parkway, Suite 360 Herndon, VA 22070, p. 3-1 - 3-55.

[14] A. Avizienis and D.A. Rennels, "Fault-Tolerance Experiments With the JPL Star Computer," in Digest of Papers, COMPCON '72, September 1972, pp. 321-324.

[15] J. Lala, "Fault Detection Isolation and Reconfiguration in FTMP: Methods and Experimental Results," in 5th AIAA/IEEE Digital Avionics Systems Conference (DASC), 1983, pp. 21.3.1-21.3.9.

[16] B. Decouty, G. Michel, and C. Wagner, "An Evaluation Tool of Fault Detection Mechanisms Efficiency," in 10th International Symposium on Fault-Tolerant Computing, 1980, pp. 225-227.

[17] Y. Crouzet and B. Decouty, "Measurement of Fault Detection Mechanisms Efficiency: Results," in 12th International Symposium on Fault-Tolerant Computing, 1982, pp.573-576.

[18] J. Arlat, Y. Crouzet, and J.-C. Laprie, "Fault Injection for Dependability Validation of Fault-Tolerant Computing , 19th International Symposium on Fault-Tolerant Computing, June 1989, pp. 348-355.

[19] M. Schuette, J. Shen, D. Siewiorek, and Y. Zhu, "Experimental Evaluation of Two Concurrent Error Detection Schemes," in 16th International Symposium on Fault-Tolerant Computing, 1986, pp. 138-143.

[20] B. Courtois, "Some Results About the Efficiency of Simple Mechanisms for the Detection of Microcomputer Malfunctions," in 9th International Symposium on Fault-Tolerant Computing, 1979, pp. 71-74.

[21] J.G. McGough and F. Swern, "Measurement of Fault Latency in a Digital Avionic Mini Processor," NASA CR-3462, Bendix Corp., October 1981, Part II, CR-3651, January 1983.
[22] X. Yang, G. York, W. Birmingham, and D. Siewiorek, "Fault Recovery of Triplicated Software on the Intel iAPX 432," in Distributed Computing Systems, May 1985, pp. 438-443.

[23] Z. Segall, D. Vrsalovic, et al., "FIAT - Fault Injection Based Automated Testing Environment," in 18th International Symposium on Fault-Tolerant Computing, 1988, pp. 102-107.

[24] R. Chillarege and N. S. Bowen, "Understanding Large System Failures - A Fault Injection Experiment," in 19th International Symposium on Fault-Tolerant Computing, June 1989, pp. 355-363.

[25] U. Gunneflo, J. Karlsson, and J. Torin, "Evaluation of Error Detection Schemes Using Fault Injection by Heavy-Ion Radiation," in 19th International Symposium on Fault-Tolerant Computing, June 1989, pp. 340-347.

[26] A. Avizienis, "The Dependability Problem: Introduction and Verification of Fault Tolerance for a Very Complex System," UCLA Dependable Computing and Fault-Tolerant Systems Laboratory, Computer Science Department, University of California, Los Angeles, California 90024.

[27] G. Gilley, "Dependable Systems: A Technical Management Program," The Aerospace Corporation, October 1990.

[28] P. S. Babcock IV, F. Leong, and E. Gai, "On the Next Generation of Reliability Analysis Tools," October 1987, NASA Contractor Report 178380, The Charles Stark Draper Laboratory, Inc. Cambridge, MA.

[29] R. Geist and K. Trivedi, "Reliability Estimation of Fault-Tolerant Systems: Tools and Techniques," in Computer, July 1990.

Tim Monaghan is a fault tolerant technical expert with the Naval Air Warfare Center in Warminster, Pa. Tim is the chairman of the Navy's Next Generation Computing Resources (NGCR) Fault Tolerant Task Group (Assessing Network, OS, and Backplane Standards for Navy use), he was the Lead Research Engineer on the Advanced Avionics Subsystem Technology Fault Tolerant Demonstration (Research Aimed at Generating Metrics and Techniques to Evaluate the Fault Tolerance of Mission Critical Computing Systems), and provides program office technical support to a number of Air and Sea platforms. He has a BSEE from Temple University and a MSCE from National Technological University.