2009

Mathematical modeling for risk-based system testing / by Karl D. Pfeiffer, Valery A. Kanevsky, and Thomas J. Housel.

Pfeiffer, Karl D.

Monterey, California. Naval Postgraduate School

NPS-GSBPP-09-028

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

Mathematical Modeling for Risk-based System Testing

08 September 2009

by

Dr. Karl D. Pfeiffer, Assistant Professor
Dr. Valery A. Kanevsky, Research Professor, and
Dr. Thomas J. Housel, Professor
Graduate School of Operational & Information Sciences
Naval Postgraduate School

Prepared for: Naval Postgraduate School, Monterey, California 93943

THIS PAGE INTENTIONALLY LEFT BLANK

Naval Postgraduate School
Monterey, California

Daniel T. Oliver
President

Leonard A. Ferrari
Executive Vice President and Provost

The Acquisition Program, Graduate School of Business & Public Policy, Naval Postgraduate School supported the funding of the research presented herein.

The report was prepared by:

_____
Karl Pfeiffer, Assistant Professor
Graduate School of Operational & Information Sciences

_____
Valery Kanevsky, Research Professor
Graduate School of Operational & Information Sciences

_____
Thomas Housel, Professor
Graduate School of Operational & Information Sciences

Reviewed by:

_____
William R. Gates, Ph.D.
Dean, Graduate School of Business & Public Policy

Released by:

_____
Karl van Bibber, Ph.D.
Vice President and Dean of Research

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form approved
OMB No 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	08 September 2009	1 October 2008 through 30 September 2009

4. TITLE AND SUBTITLE
Mathematical Modeling for Risk-based System Testing

5. FUNDING

6. AUTHOR (S)
Karl D. Pfeiffer, Valery A. Kanevsky and Thomas J. Housel

7. PERFORMING ORGANIZATION NAME (S) AND ADDRESS (ES)
NAVAL POSTGRADUATE SCHOOL
GRADUATE SCHOOL OF BUSINESS AND PUBLIC POLICY
555 DYER ROAD
MONTEREY, CA 93943-5103

8. PERFORMING ORGANIZATION REPORT NUMBER
NPS-GSBPP-09-028

9. SPONSORING/MONITORING AGENCY NAME (S) AND ADDRESS (ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words.)

Testing of complex systems is a fundamentally difficult task whether locating faults (diagnostic testing) or implementing upgrades (regression testing). Branch paths through the system increase as a function of the number of components and interconnections, leading to exponential growth in the number of test cases for exhaustive examination. In practice, the typical cost for testing in schedule or in budget means that only a small fraction of these paths are investigated. Given some fixed cost, then, which tests should we execute to guarantee the greatest information returned for the effort? In this work, we develop an approach to system testing using an abstract model flexible enough to be applied to both diagnostic and regression testing, grounded in a mathematical model suitable for rigorous analysis and Monte Carlo simulation. The goal of this modeling work is to construct a decision-support tool for the Navy Program Executive Office Integrated Warfare Systems (PEO IWS) offering quantitative information about cost versus diagnostic certainty in system testing.

14. SUBJECT TERMS	15. NUMBER OF PAGES
diagnostic testing, regression testing, automated testing, Monte Carlo simulation, sequential Bayesian inference	47
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT:	18. SECURITY CLASSIFICATION OF THIS PAGE:	19. SECURITY CLASSIFICATION OF ABSTRACT:	20. LIMITATION OF ABSTRACT:
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UU

NSN 7540-01-280-5800

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Abstract

Testing of complex systems is a fundamentally difficult task whether locating faults (diagnostic testing) or implementing upgrades (regression testing). Branch paths through the system increase as a function of the number of components and interconnections, leading to exponential growth in the number of test cases for exhaustive examination. In practice, the typical cost for testing in schedule or in budget means that only a small fraction of these paths are investigated. Given some fixed cost, then, which tests should we execute to guarantee the greatest information returned for the effort? In this work, we develop an approach to system testing using an abstract model flexible enough to be applied to both diagnostic and regression testing, grounded in a mathematical model suitable for rigorous analysis and Monte Carlo simulation. The goal of this modeling work is to construct a decision-support tool for the Navy Program Executive Office Integrated Warfare Systems (PEO IWS) offering quantitative information about cost versus diagnostic certainty in system testing.

Keywords: diagnostic testing, regression testing, automated testing, Monte Carlo simulation, sequential Bayesian inference

THIS PAGE INTENTIONALLY LEFT BLANK

About the Authors

Author: **Karl D. Pfeiffer** is an Assistant Professor of Information Sciences at the Naval Postgraduate School; and an active duty Air Force officer. His current research interests include decision-making under uncertainty, particularly with regard to command and control (C2) systems; stochastic modeling of environmental impacts to weapons and communication systems; and probability modeling and numerical simulation in support of search, identification and pattern recognition applications (e.g., complex system testing, allocation of effort for reconnaissance, etc.).

Author: **Valery A. Kanevsky** is a Research Professor of Information Sciences at the Naval Postgraduate School. His research interests include probabilistic pattern recognition; inference from randomly distributed inaccurate measurements, with application to mobile communication; patterns and image recognition in biometrics; computational biology algorithms for microarray data analysis; Kolmogorov complexity, with application to value allocation for processes without saleable output; and Monte Carlo methods for branching processes and simulation of random variables with arbitrary distribution functions. Valery's most current work is focused on statistical inference about the state of a system based on distributed binary testing. Another area of interest is in the so-called needle-in-a-haystack problem: searching for multiple dependencies in activities within public communication networks as predictors of external events of significance (e.g., terrorist activities, stock market anomalies).

Author: **Thomas J. Housel** is a Professor of Information Sciences at the Naval Postgraduate School. Prof Housel specializes in valuing intellectual capital, knowledge management, telecommunications, information technology, value-based business process re-engineering, and knowledge value measurement in profit and non-profit organizations. His current research focuses on the use of knowledge-value added (KVA) and real options models in identifying, valuing, maintaining, and exercising options in military decision-making. His work on measuring the value of

intellectual capital has been featured in a Fortune cover story (October 3, 1994) and Investor's Business Daily, numerous books, professional periodicals, and academic journals (most recently in the *Journal of Intellectual Capital*, 2005). His latest books include: *Measuring and Managing Knowledge* and *Global Telecommunications Revolution: The Business Perspective* with McGraw-Hill (both in 2001).

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

1. Overview 1

2. Background 3

3. Model Formulation 5

3.1 System and Module Objects 5

3.2 Test Objects 6

3.3 Summary 9

4. Mathematical Fundamentals 11

4.1 Objective Measures of the System State 11

4.2 Simple Step-wise Testing 13

4.3 Variable Cost per Test 16

4.4 Summary 16

5. Simulation Results and Analysis 19

5.1 Model Details 19

5.2 Results from Initial Experiments 21

6. Summary and Future Work 25

List of References 27

Initial Distribution List 33

THIS PAGE INTENTIONALLY LEFT BLANK

1. Overview

Testing of complex systems is a fundamentally difficult task whether locating faults (diagnostic testing) or implementing upgrades (regression testing). The number of branch paths through the system typically grows as a function of the number of components and interconnections, leading to near-exponential growth in test cases for an exhaustive examination. This study examines optimal system testing by using classic fault diagnosis scenarios as the basis from which to develop a mathematical model that is flexible enough to extend to regression testing cases.

In this research, we establish the groundwork for a decision-support tool for the Navy Program Executive Office Integrated Warfare Systems (PEO IWS). This tool will provide quantitative information about trade-offs among cost, risk, and the degree of system testing conducted. Initially, we seek to answer the question: given a failure in an operational system, what is the best test-risk-cost strategy to locate the failed unit of replacement? Further development of this model will investigate the question: given an engineering upgrade to a module, how much regression testing must we accomplish on the system for a given level of risk?

The scientific contribution of this work lies in a novel, information-driven approach to testing. Having characterized our system in terms of probabilities of failure of individual components, we can assess at any time the information entropy associated with that knowledge and assess the change in entropy possible by applying particular tests from our diagnostic inventory. We can then more readily assess quantitatively the information returned for the cost incurred by a test or battery of tests.

We expect the practical results of this work will be useful throughout the system lifecycle, from acquisition to fielding and maintenance. The decision-support prototype tools delivered should, for example, yield significant insight into designing test suites for new weapons systems and for improving the use of existing suites in current systems, such as the Aegis combat system. This work should also be useful for optimizing decisions within the corrective maintenance (courses of action) module within the

condition-based maintenance (CBM) and distance support (DS) systems for the Surface Warfare Enterprise.

2. Background

Mathematical models of component and system reliability have roots in the work of von Neumann (1952) and Moore and Shannon (1956), as well as in the seminal text by Barlow and Proschan (1965). The focus of these early works is generally on assessing the overall system reliability, particularly with regard to the economics of preventative vice reactive maintenance (see, for example, Bovaird, 1961). In the present work, the focus is on efficiently identifying a defective-by-design or failed component in a complex system.

This fault diagnosis, sometimes referred to as the test-sequencing problem, has also been well studied (see, for example, Sobel & Groll, 1966; Garey, 1972; Fishman, 1990; Barford, Kanevsky, & Kamas, 2004). In general, these investigators start with a system in a known, failed state with the goal of finding the most cost-effective sequence of diagnostics to locate the failed component (or components) under a given set of assumptions.

In contrast to fault diagnosis, the general case of regression testing appears to have received less attention in the open literature, with more specific cases examined in the realm of software engineering (see, for example, Weyuker, 1998; White, 1992; Tsai, 2001; Mao & Lu, 2005; Leung, 1991; Rothermel, 2001). These studies typically start with a fully functioning system that is undergoing component modification or upgrade, with the task of establishing that component modifications have not introduced new defects into the system.

In the present study, we treat testing as a unified activity, with risk and cost as the common tension regulating the degree of testing required. From a fault-diagnosis perspective, we consider both the cost of module replacement and the cost of testing. We want to replace the fewest number of components as quickly as possible while ensuring the system is restored to perfect functionality. From a regression-testing perspective, we test our system following a component or system upgrade to ensure what our system remains in perfect function under load. The element of risk

from this perspective is that costs incurred for perfect knowledge may rapidly approach infinity. From an operational perspective, then, we must accept with some level of confidence (e.g., 99%, 95%) that our diagnosis or prognosis is correct.

3. Model Formulation

The growing use of commercial off-the-shelf technologies in current weapons systems (Caruso, 1995; Dalcher, 2000), coupled with the complexity of end-to-end systems (Athans, 1987), suggests that we may never have enough information to fully specify our system as a white box, with all software, hardware and communication interfaces perfectly characterized. We thus construct our model with broad parameters that can be constrained as narrowly as available information permits.

We characterize the model system as a collection of modules comprising the system and a collection of tests used to interrogate the system. When the system is down, we assume that one or more modules have failed. We examine the system through this test suite to locate the correct module or modules to replace. We assume that tests return ambiguous information about the state of modules within the system and that some sequences of tests must typically be applied to arrive at a correct diagnosis. Stochastic simulation of the model system provides a framework in which different strategies may be applied and measured for further insight. Using this Monte Carlo approach, we may also test the bounds of our initial assumptions with additional simulation.

3.1 System and Module Objects

We form the model system **S** as a collection of modules or units of replacement. Each module M_i represents the smallest diagnostic unit, which does not necessarily correspond to a single physical component in the modeled system. We consider, for example, a computer server comprised of motherboard, hard drive and power supply, each of which may cause the computer server to fail. This would be modeled as a single module labeled *Server* if the standard corrective maintenance action is to replace the entire unit. A fundamental assumption in this abstraction is that the physical system is decomposable into these units of replacement. We note that even in this example, a separate diagnostic model could

be applied to the server, treating each of its subcomponents (motherboard, hard drive, power supply) as replaceable units.

We assume **S** is always in one of two states: fully functioning (UP) or inoperable (DOWN). Each module is similarly assessed as GOOD or BAD. We take **S** as UP if and only if every M_i is GOOD. In practice, this means that if we find **S** inoperable, then we may assume that one or more modules have failed. In this event, we seek to replace the fewest number of modules with the least testing and in the shortest time.

Each M_i is modeled as the unit circle A_i. Defects, when present, are assumed uniformly distributed on this circle. We assume that while multiple modules may be defective, only one defect exists per module. A defect in M_i is modeled as a random point on A_i or, equivalently, a random point on the interval [0, 1].

Fundamental to this aspect of the model is a source of failure-rate data for the system components. These failure rates become the a priori data in the larger probability model and so do not necessarily need to be precise to add value to simulation results. The relative rates among the modeled components (e.g., the *Server* module fails about five times as often as the *Router* module) should be close to the observed data in the physical system to provide the most realistic convergence in testing to a correct diagnosis.

3.2 Test Objects

Tests are modeled as system objects that, when executed, provide an ambiguous assessment of one or more modules within **S**. This ambiguity stems from two essential elements that map the tractable model to physical reality.

The first aspect is that any given test likely exercises only a portion of the functionality within a module. Although the module is the unit of replacement, we parameterize the sub-module details by treating them as a continuous space covered, in part, by a given test.

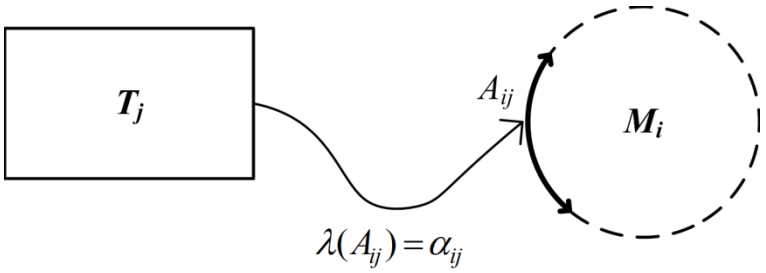

Figure 1. The simple coverage of test T_j on module M_i indicated by the solid arc A_{ij}. The measure of this coverage $\lambda(A_{ij}) = \alpha_{ij}$ represents the fraction of M_i exercised by T_j.

We model the coverage of test T_j on module M_i as the arc A_{ij} (see Figure 1). When T_j is executed, or applied to the model system, the arc A_{ij} on M_i is inspected for a defect. Given the assumption that defects appear uniformly on this unit circle, the probability that a defect in M_i will be detected by T_j is the measure of this arc or $\lambda(A_{ij}) = \alpha_{ij}$. The scalar probability of detection by a test is precisely the user-specified functionality exercised by the test. This element of our language of description permits some ambiguity in characterizing the physical system (e.g., built-in self-test 3 exercises about 45% of the functionality of the graphics processing unit) without loss of rigor in modeling these tests and modules. In practice, given a sufficient number of real-world cases from the physical system, this estimate for A_{ij} could be refined through analysis of simulation results.

The second ambiguous aspect of results from testing is that any given test likely covers multiple modules, such that any test result must be interpreted as applying to *all* modules covered by that test (see Figure 2). For example, a positive result (FAIL) from a diagnostic test that covers the modules *Carburetor, Distributor Cap,* and *Spark Plug Wiring* indicates that at least one of these modules contains a defect (has failed), though additional testing would be required to identify which of these modules is the culprit. Because we expect that a given test exercises multiple modules in the system, we speak more generally of the coverage of T_j on S (see Figure 2).

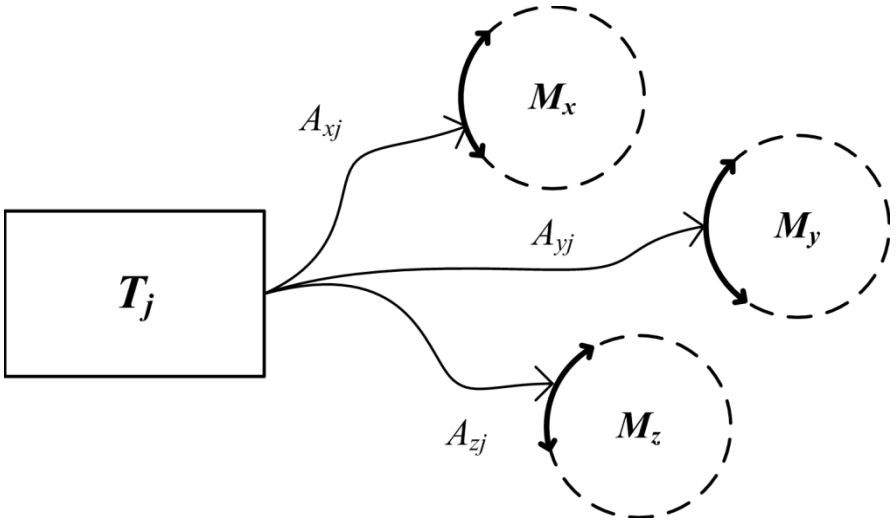

Figure 2. Notional depiction of the coverage of T_j on **S**, with multiple modules exercised upon execution of this test. A FAIL result from T_j indicates that at least one of the subset $\{M_x, M_y, M_z\}$ has failed.

Within the model, a test when executed assumes one of two values: PASS or FAIL. A PASS result for a given test T_j indicates that no region covered by this test contains a defect. A FAIL result indicates that at least one of the modules covered by T_j contains a defect, or is BAD in the model definition. While a FAIL result should reduce the set of modules that may need to be replaced, a perfect result—replacing only those modules that have failed—will typically require some sequence of tests. Indeed, for a particular configuration of tests and modules this perfect result may not be achievable. Analysis of simulation results should help to identify those cases where further testing will yield no new information.

The use of vector arcs to model the coverage relationship between tests and modules enables precision when specifying the coverage by multiple tests on a single replaceable unit (see Figure 3). Although several tests in the system suite may exercise a given module, it is likely in the physical system that these tests overlap significantly. This language of description, then, permits a user specification of the physical system in broad terms (e.g., the *Remote Control* test and *Obstacle Detection* test both exercise about 70% of the *Garage Door Motor*

module, with about 20% overlap between the two tests). Even if these data are estimated from the physical system, existing case data and simulation results could be used to provide better specification of these joint coverages.

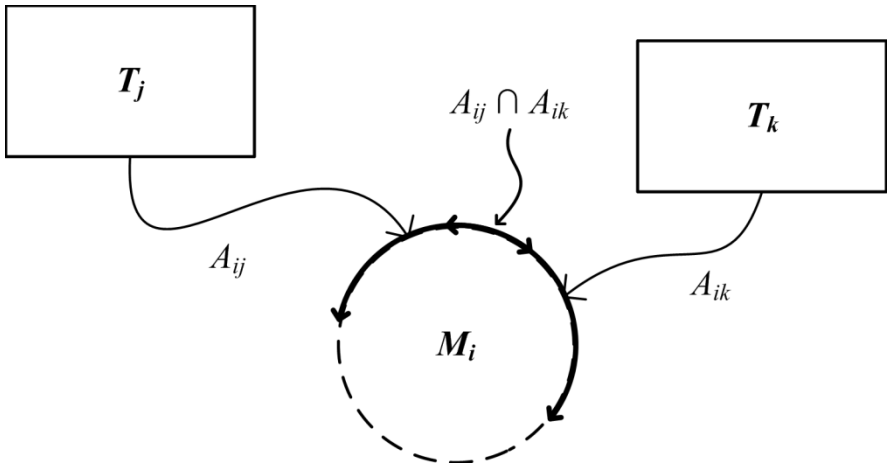

Figure 3. Overlapping coverage between tests T_j and T_k are characterized with the arcs A_{ij} and A_{ik}. The joint coverage is computable as the intersection of these arcs.

3.3 Summary

This conceptual model captures the essential elements of a system with respect to diagnostic testing and module repair or replacement. The physical system is specified in terms of modules, tests, and coverages, with model elements constructed in such a way that imperfect information can still be used as an initial state. Although the model requires that the physical system be decomposable into discrete units of replacement, this does not limit the usefulness of this approach. The fundamental diagnostic techniques could easily be applied at the sub-module level by treating a given module as a system, with sub-components then modeled as modules. In the present study, we limit our investigation to a single-layer model, though future work could nest diagnostic levels across a complex system. We next formalize these model elements in mathematical language to construct a suitable computer simulation to investigate these testing strategies.

THIS PAGE INTENTIONALLY LEFT BLANK

4. Mathematical Fundamentals

Our goal in system testing is to maximize certainty at minimum cost. In developing a probability framework to model this process, we first form simple objective measures to characterize knowledge of the system state. We next examine a simple, step-wise strategy to predict a test sequence that will maximize or minimize these measures. We then extend these simple strategies by considering a variable cost per test to examine diagnosis under limited resources.

4.1 Objective Measures of the System State

Let B_i be the event that module M_i is BAD, with probability $P(B_i) = b_i$. Given a system **S** comprised of *m* modules, we can characterize our knowledge of the system state as a vector of these probabilities:

$$\mathbf{K}^t = \left\{ b_1^t \ldots b_m^t \right\} \tag{4.1}$$

The index *t* is time-like, indicating the number of tests that have been applied to the system. At *t = 0,* no tests have been applied and all b_i are set to their initial failure rates. Fundamental to our conceptual model is a source of failure-rate data, or an *a priori* probability that a particular replaceable unit is defective.

We desire a diagnosis in which the components of **K** are only zero or one, meaning that we know with absolute certainty that a particular module is GOOD or BAD. In practice, this ideal diagnosis may be too costly or simply impossible to determine (see, for example, Cover & Thomas, 1991, Ch. 7). Instead, we take a step-wise approach in which we apply tests from our suite of diagnostics to incrementally improve our knowledge of **S**.

One intuitive measure of $\mathbf{K^t}$ is the information entropy (Shannon, 1948). For a single module, we compute the entropy h_i as:

$$h_i = -b_i \log_2 b_i - (1-b_i)\log_2(1-b_i) \tag{4.2}$$

We see that as b_i tends to zero or one, h_i is minimized (see Figure 4). By applying tests from our diagnostic suite, we should become more certain about the state of a module (GOOD or BAD). We measure this improvement in certainty as a reduction in the individual module entropy. Across the system, we take the aggregate measure as:

$$H^t = \frac{1}{m}\sum_{i=1}^{m} h_i^t = \frac{1}{m}\sum_{i=1}^{m} -b_i^t \log_2 b_i^t - (1-b_i^t)\log_2(1-b_i^t) \tag{4.3}$$

Using this measure, we seek an ordering of *k* tests such that:

$$H^0 \geq H^1 \geq \ldots \geq H^k$$

That is, each test applied should act to modify some subset of module b_i to reduce the entropy of $\mathbf{K^t}$. An optimal step-wise strategy, then, would seek to maximize $\Delta H = H^t - H^{t+1}$ for each test applied to the system under diagnosis.

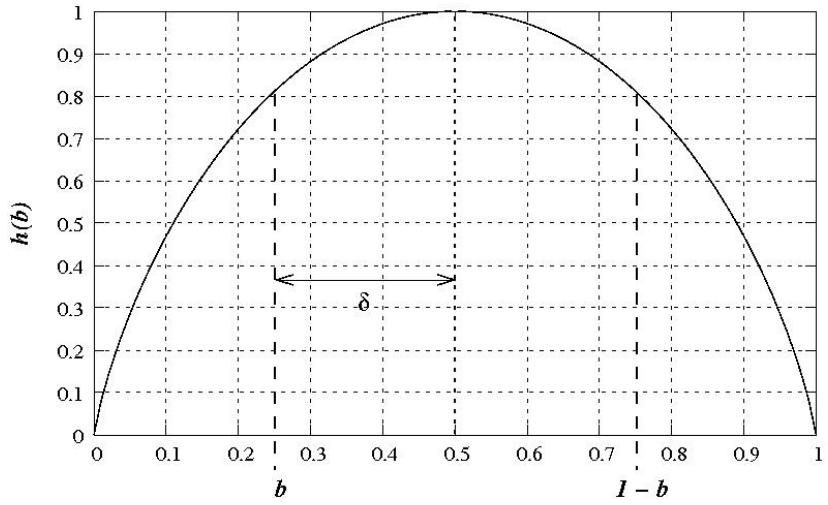

Figure 4. Module entropy $h(b_i)$, with notional module probability b_i indicated. Note that by symmetry, $h(b_i) = h(1 - b_i)$, with distance 2δ between these states.

Entropy is computationally attractive, though h_i may be less intuitive when deciding which modules to replace. The probability b_i offers some insight into the likelihood that the module M_i should be replaced, such that a reasonable decision criterion D_i would be:

$$D_i = \begin{cases} \text{KEEP} & \text{if } b_i = \min(b_i, 1-b_i) \\ \text{REPLACE} & \text{if } b_i = \max(b_i, 1-b_i) \end{cases} \tag{4.4}$$

In effect, if the probability that M_i is bad is above ½, then we should replace it, and if the probability is below ½, then we should keep it. If, for example, a particular module has a b_i = 0.70, then we replace it knowing that this informed guess should be correct 70% of the time; 30% of the time we will unnecessarily replace a GOOD module. Our number of correct guesses across the system will increase as each b_i is adjusted by testing away from ½ towards either zero or one (see Figure 4). Thus, minimizing system entropy *H* in a step-wise process is equivalent to maximizing the number of correct replacement decisions, or correct diagnoses.

4.2 Simple Step-wise Testing

We seek to minimize the entropy of the probability vector **K** (Equation 4.1) by applying tests in step-wise fashion to update the component probabilities. For each candidate T_j in our diagnostic suite, we can compute a candidate ΔH, and then choose that test which causes the maximum reduction in entropy (largest ΔH).

In forming these predicted ΔH, we must account for both possible test outcomes. Let the event P_j represent the execution of test T_j with a PASS result. Similarly, let F_j represent the event of a FAIL result. To estimate the reduction in entropy possible by execution of test T_j at some point *t* in testing, we use the weighted sum:

$$\Delta H(T_j) = H^t - P(P_j)\sum_i h(B_i \mid P_j) - P(F_j)\sum_i h(B_i \mid F_j) \tag{4.5}$$

The entropy of the Bayesian result from either outcome is computed with:

$$h(B_i \mid P_j) = -P(B_i \mid P_j)\log_2 P(B_i \mid P_j) - P(G_i \mid P_j)\log_2 P(G_i \mid P_j) \tag{4.6}$$

$$h(B_i \mid F_j) = -P(B_i \mid F_j)\log_2 P(B_i \mid F_j) - P(G_i \mid F_j)\log_2 P(G_i \mid F_j)$$

We first consider those probabilities that describe whether a test will likely PASS or FAIL. If T_j only covers one module, then the simple probability that this test will PASS becomes:

$$\begin{aligned} P(P_j) &= P(M_i \text{ is GOOD}) \bigcup P(M_i \text{ is BAD but undetected}) \\ &= (1-b_i)+(1-\alpha_{ij})b_i \\ &= 1-b_i\alpha_{ij} \end{aligned}$$

We note that if T_j has no coverage on M_i then $\alpha_{ij} = 0$ and this test will always PASS. Similarly:

$$\begin{aligned} P(F_j) &= P(M_i \text{ is BAD and detected}) \\ &= \alpha_{ij}b_i \end{aligned}$$

We note that if T_j has perfect coverage on M_i ($\alpha_{ij} = 1$), then the probability that this test will pass reduces to the probability that the covered module is BAD.

In practice, we expect a given T_j will cover multiple modules, requiring that for a PASS event all modules are either GOOD or BAD but undetected.

$$P(P_j) = \prod_{i=1}^{m}(1-\alpha_{ij}b_i) \tag{4.7}$$

We can then compute a FAIL event for T_j as the complement of a PASS, thus:

$$P(F_j) = 1-\prod_{i=1}^{m}(1-\alpha_{ij}b_i) \tag{4.8}$$

Even though these products (Equations 4.7, 4.8) are computed over all modules in the system, we note that for those modules with no coverage by T_j, α_{ij} reduces to zero and the product is unaffected.

The Bayesian results required for Equation 4.6 can be computed with:

$$P(B_i \mid F_j) = \frac{P(F_j \mid B_i)P(B_i)}{P(F_j \mid B_i)P(B_i) + P(F_j \mid G_i)P(G_i)} \quad (4.9)$$

$$P(B_i \mid P_j) = \frac{P(P_j \mid B_i)P(B_i)}{P(P_j \mid B_i)P(B_i) + P(P_j \mid G_i)P(G_i)}$$

Individual terms are computed as:

$$P(F_j \mid B_i) = \text{Given } M_i \text{ is BAD, probability } T_j \text{ will FAIL} = \alpha_{ij} b_i$$
$$P(F_j \mid G_i) = \text{Given } M_i \text{ is GOOD, probability } T_j \text{ will FAIL} = 1 - \prod_{k \neq i} (1 - \alpha_{kj} b_k)$$

$$P(P_j \mid B_i) = \text{Given } M_i \text{ is BAD, probability } T_j \text{ will PASS} = 1 - \alpha_{ij}$$
$$P(P_j \mid G_i) = \text{Given } M_i \text{ is GOOD, probability } T_j \text{ will PASS} = \prod_{k \neq i} (1 - \alpha_{kj} b_k)$$

$$P(B_i) = \text{Prior probability that } M_i \text{ is BAD} = b_i$$
$$P(G_i) = \text{Prior probability that } M_i \text{ is GOOD} = 1 - b_i$$

We note that in the case of $P(F_j|G_i)$ and $P(P_j|G_i)$ we must examine all other modules covered by test T_j to compute these probabilities.

This analysis provides a tractable, one-step method for sequencing tests by maximum reduction in entropy, though the computation grows approximately as the product of the number of modules, *m,* and number of tests, *p*. Additional insight may be possible by considering in our prediction the next best two tests, or perhaps *n* tests, in reducing entropy, though the computational cost grows as mp^n.

We have assumed implicitly in this analysis that all tests carry the same cost in some unified measure of time and money. We next discuss briefly the analysis with variable cost per test.

4.3 Variable Cost per Test

We can estimate the information gain possible with any test in our model system (Equation 4.5). With the inclusion of information about the cost per test $C(T_j)$, we can modify our objective function to compute, in effect, a cost per bit or:

$$\Phi(T_j) = \frac{\Delta H(T_j)}{C(T_j)} \tag{4.10}$$

Our step-wise strategy, then, is to choose not the largest ΔH but the largest Φ. Although the analysis in Section 4.2 does not change, this additional model element permits a broader range of investigation in computational scenarios.

For example, with this extension of the present analysis (Equation 4.10), we could examine a scenario where the diagnostic resources were limited by some finite purse (in terms of C) that, when exhausted, required the operators to make a replacement decision. In this case, a simple step-wise scenario would likely be less effective. Indeed, this particular example is more similar to the classic knapsack problem (see, for example, Corman, Leiserson, & Rivest, 1990).

Given data on both test and module cost, we could also examine, at every iteration, whether the next best test (or next best *n* tests) cost more than simply replacing the current "best" candidates in the system of modules. Stochastic simulation of this scenario, given approximate real-world data, should yield significant insight into the physical systems under maintenance.

4.4 Summary

We have presented a mathematical framework to support the conceptual model of testing described in Section 3. Upon finding our system is down, our notional diagnostic algorithm is:

1. Form the initial vector $\mathbf{K}^0$ from the given module failure rates.
2. From our diagnostic test suite, choose that T_j which maximizes ΔH.
3. After performing the selected test, update $\mathbf{K}^t$ to $\mathbf{K}^{t+1}$.
4. If we have not reached our stopping criteria, then return to (2).

In practice, stopping criteria might include

a. System entropy is very close to zero,
b. Time or resources have expired,
c. Cost of the next test exceeds cost of replacing candidate modules, or
d. Actual change in entropy on this cycle is very close to zero.

Using entropy reduction as an objective measure, a simple analysis demonstrates the general utility of this approach while additional physical data (e.g., cost per test) could easily be incorporated into the computation. We next discuss the implementation of these ideas in a computer simulation and then review results from idealized scenarios.

THIS PAGE INTENTIONALLY LEFT BLANK

5. Simulation Results and Analysis

To demonstrate the feasibility of the ideas developed in this study, a simulation was developed suitable for desktop computing. Because no physical system data were immediately available, distributions of modules and test coverages were constructed, randomly subject to certain design constraints. While these scenarios provide some insight into this approach to systems testing, sufficient flexibility exists in the computer code to extend the model easily to real-world systems.

5.1 Model Details

A Java development environment was selected based on the strong numerical facilities available under most implementations and the widely portable nature of most Java code. Simulations were run primarily on a Windows Vista (x64) workstation while portability tests were run on both Ubuntu Linux 8.04 and Mac OS X 10.5 (Leopard) machines.

The code implements object models of tests and modules, collected under a system object. In most scenarios, 30 modules and 60 tests were constructed within the system, with test coverages spread randomly by test over some number of modules, nominally no fewer than 2 and no more than five. That is, for each test T_j, a random integer *q* was chosen from {2, 3, 4, 5}, and then *q* modules were randomly selected from the system set and connected to T_j with random coverages. Initial failure rates were assigned to modules from a uniform distribution on the interval (0,1). While the code is quickly reconfigurable for more robust or physically realistic scenarios, these parameters were fixed for an initial comparison among simple test strategies.

The best-next test strategy, based on reduction of entropy, was described in Section 4. To make at least initial comparisons with the simulation code, a worst-next test strategy was implemented within the software to explore a pathological case where every test selected maximized entropy, or equivalently, increased

uncertainty. As a baseline scenario, a random test strategy was implemented as well, with tests selected at random from the system suite.

Prior to the start of a set of trials, a failure deck was created based on the relative failure rates of modules within the system. Similar to a deck of playing cards, modules appear in the failure deck based on their standing relative to the minimum failure rate in the system; thus, if the minimum failure rate across the system is 0.2, then a module with a failure rate of 0.6 will appear three times within this failure deck. The same deck is employed across all trials to simulate the relative appearance of failures in a physical system.

Prior to the start of a simulation, a test deck with one entry for each test is created (copied) from the system configuration. Strategies that compute the next-best (or next-worst) test operate on this deck. As a test is executed, it is removed from the deck, insuring that no test in our system will be executed more than once per trial. This also reduces the search space for the next test. A new test deck must be generated with each trial.

A single trial is processed in the following manner:

1. All module b_i are initialized from failure-rate data.
2. A module is selected from the failure deck, and a defect is planted in this module.
3. A test is chosen based on a simple strategy (best, random, worst).
4. The test is applied to the system object.
5. All affected b_i are updated based on the outcome of (4).
6. If we still have a test in the test deck, then we return to (3).

Using a 2 GHz Intel processor, a simulation of 1,000 trials required on average about 2.5 minutes for a randomized configuration with 60 tests and 30 modules. For a larger system configuration with 100 tests and 50 modules, run-time averaged about 5 minutes for 1,000 trials. In general, a ratio of 2:1 between tests and modules appeared to guarantee a correct diagnosis was obtainable with the random configuration of coverages between tests and modules constrained to no fewer than 2 and no more than 5 modules per test.

5.2 Results from Initial Experiments

Over some number of trials (nominally 100 to 1,000), the module traces for each strategy were aggregated. In these initial experiments, no stopping criteria were applied, and with the idealized scenario, the best-next strategy showed little improvement after 40 tests (out of 60) were executed (see Figure 5).

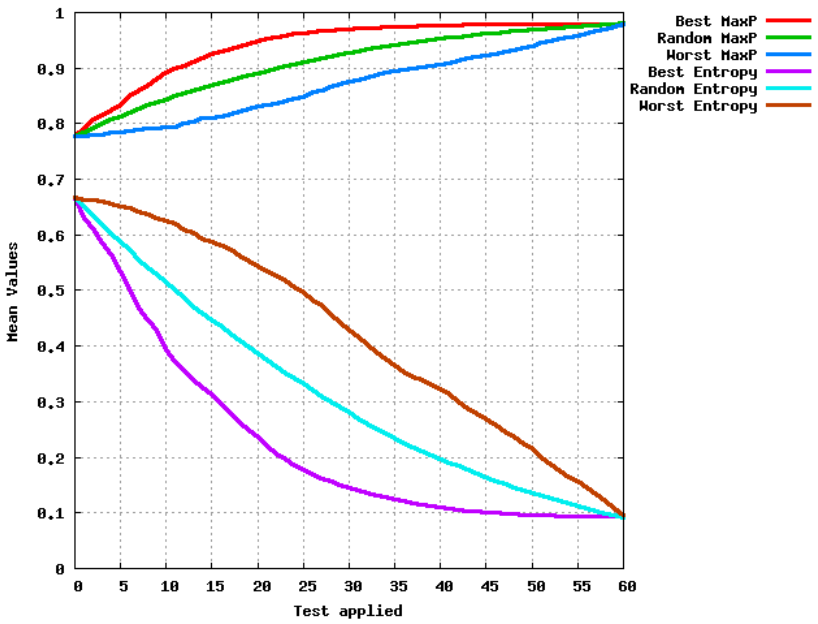

Figure 5. Mean diagnostic traces from 100 trials using best, random, and worst next-test strategies. Both system entropy (bottom traces) and maximum probability (top traces) are depicted, with all 60 tests applied, though in practice we would likely stop sooner.

Entropy variance (see Figure 6) for the best-next strategy shows a peak at about test 19, with the caveat that test 19 would be a different system object for each of the trials. This peak is consistent with the reduction in steepness of descent in the best-next mean entropy trace (see Figure 5) and the increase of the maximum probability function to greater than 90%.

The distribution of model probabilities from one of the best-next trials shows the evolution of a correct diagnosis (see Figures 7 and 8) as system testing unfolds.

Although solid lines are used to highlight this dynamic in Figures 7 and 8, the module probabilities are, in fact, discrete. Early in testing, about test 5, the module probabilities seem unremarkable compared to the true state (see Figure 7); the best-next strategy strongly identifies Module 2 as the defective candidate, though several other modules still keep the aggregate entropy relatively high (*H = 0.40*, see Figure 7). By test 25, however, Module 2 shows a relatively large b_i = 0.84 with an overall aggregate entropy (*H=0.25*, see Figure 8). Additional testing refines the individual module probabilities so that by test 30, a correct diagnosis appears evident (see Figure 8).

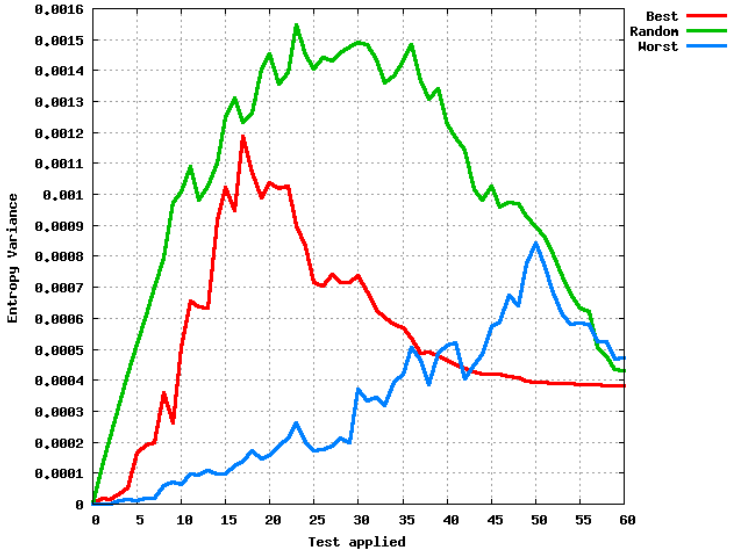

Figure 6. Diagnostic trace of variance in entropy from 100 trials using best, random, and worst next-test strategies, with all 60 tests applied.

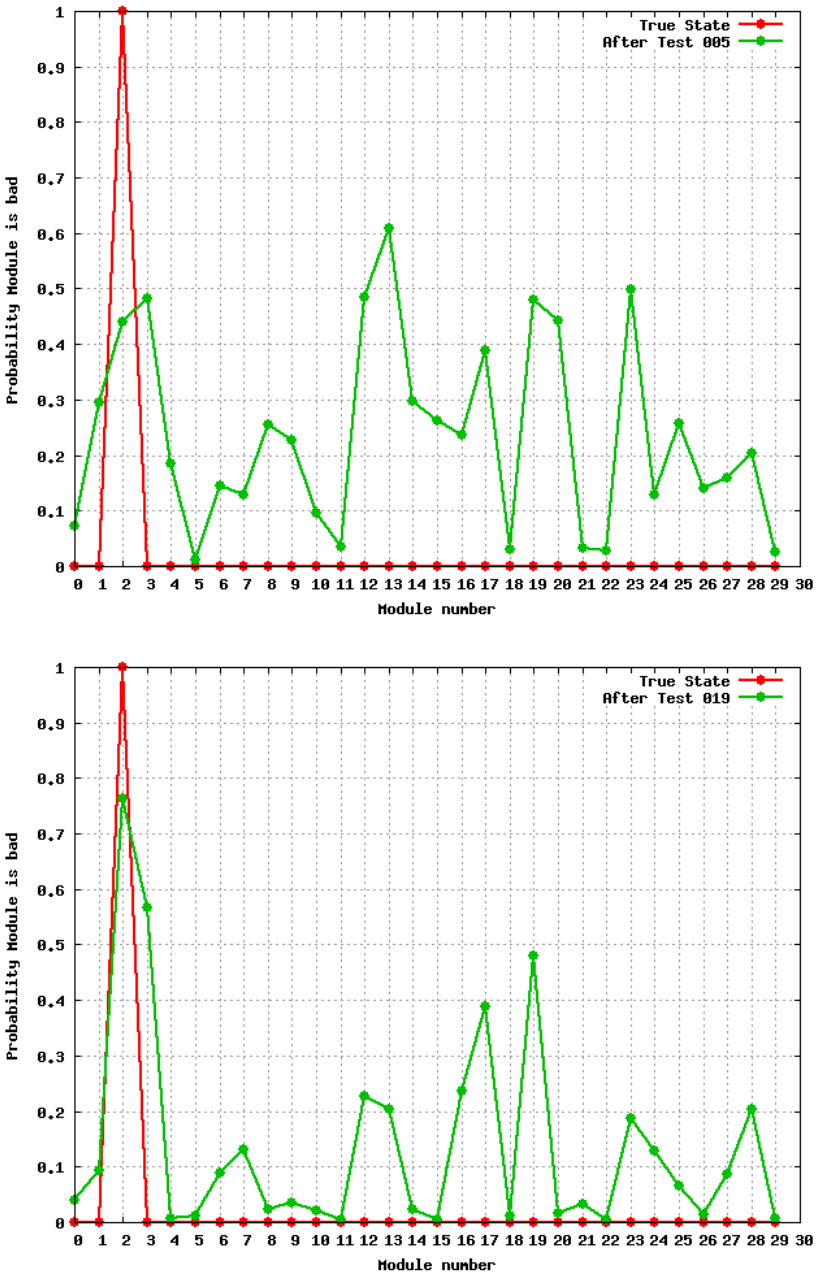

Figure 7. From a best-next test trial, module probabilities (b_i, in green) are shown versus the true state (in red) after test 5 (top) and test 19 (bottom). After test 5, H=0.64, and after test 19, H = 0.40.

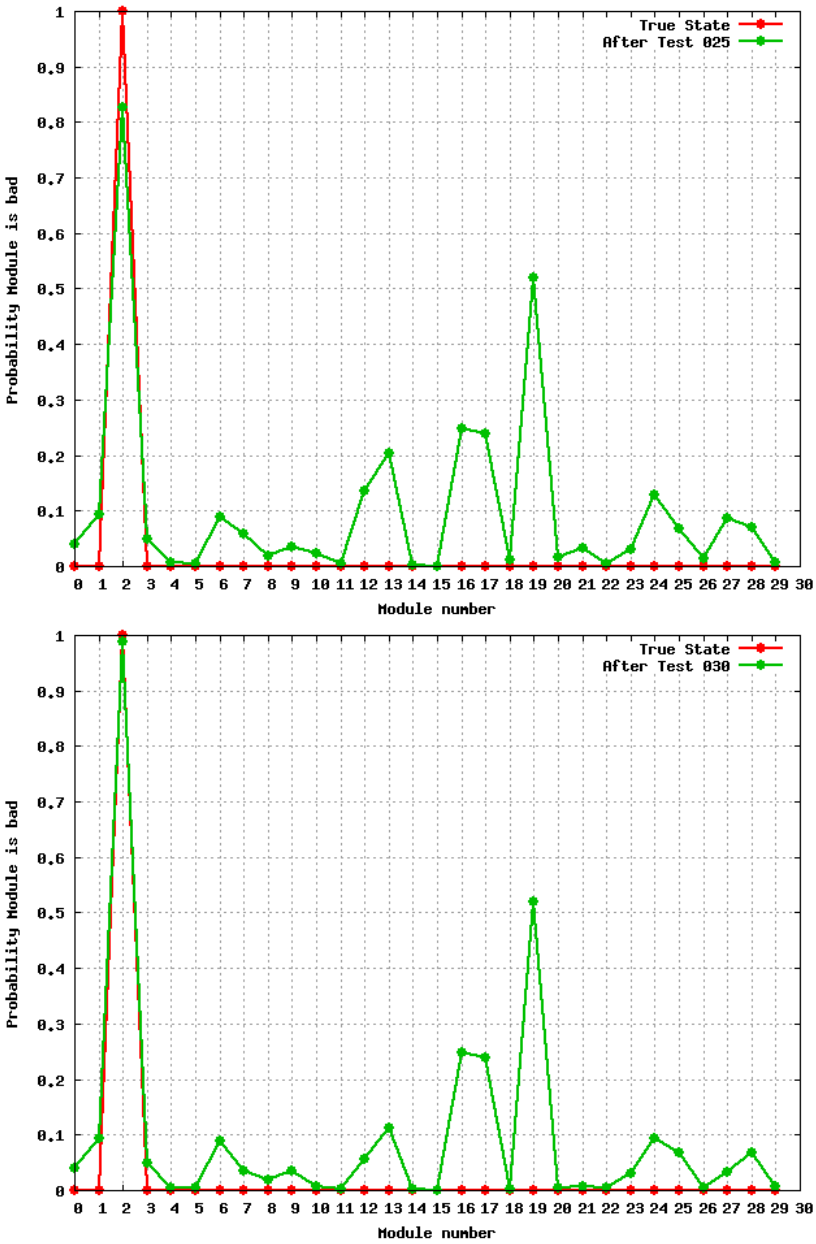

Figure 8. From the same best-next test trial as Figure 7, module probabilities (b_i, in green) are shown versus the true state (in red) after test 25 (top) and test 30 (bottom). After test 25, H=0.25, and after test 30, H = 0.15.

6. Summary and Future Work

In this study, we have developed a simple but effective framework to examine the testing of complex systems. The idealized numerical experiments conducted in this study support the use of entropy reduction as an effective means to guide diagnostic testing, though these initial simulations can provide only simple insights. Real-world failure rates and coverages are needed to further investigate the usefulness of this approach for diagnosing physical systems.

Additional avenues of research will open up with more realistic scenarios with which to exercise and develop this model. For example, simulation studies could inform the design of test suites for new weapons systems. By using available cost data for both tests and replaceable units, further research could help to develop or refine a diagnostic strategy to balance the cost of expensive, granular testing against the cost of routine maintenance. When modeling a fielded system, real-time, failure-rate data could be used to update the simulation and further improve fidelity.

By using the flexible but precise language of our conceptual model, we can investigate the underlying probabilistic relationships of existing, complex systems. Although the original motivation for this work was the diagnostic testing of mechanical and electronic systems, with little modification classic regression testing scenarios could be modeled in simulation code to estimate the degree and cost of testing following system upgrades.

THIS PAGE INTENTIONALLY LEFT BLANK

List of References

Athans, M. (1987). Command and control (C2) theory: A challenge to control science. *IEEE Transactions on Automatic Control*, *32*(4), 286–293.

Barford, L., Kanevsky, V., & Kamas, L. (2004). Bayesian fault diagnosis in large-scale measurement systems. In *IMTC 2004: Instrumentation and Measurement Technology Conference* (pp. 1234–1239). Como, Italy: IEEE.

Barlow, R.E., & Proschan, F. (1965). *Mathematical theory of reliability*. Philadelphia, PA: John Wiley and Sons.

Bovaird, R.L. (1961). Characteristics of optimal maintenance policies. *Management Science*, *7*(3), 238–253.

Caruso, J.A. (1995). The challenge of the increased use of COTS: A developer's perspective. In *Proceedings of the Third Workshop on Parallel and Distributed Real-Time Systems* (pp. 155–159).

Corman, T.H., Leiserson, C.E., & Rivest, R.L. (1990). *Introduction to algorithms*. Cambridge, MA: MIT Press.

Cover, T.M., & Thomas. J.A. (1991). *Elements of information theory*. New York: John Wiley and Sons.

Dalcher, D. (2000). Smooth seas -rough sailing: The case of the lame ship. In *Proceedings* of the Seventh IEEE International Conference and Workshop (pp. 393–394). Engineering of Computer Based Systems.

Fishman, G.S. (1990). How errors in component reliability affect system reliability. *Operations Research*, *38*(4), 728–732.

Garey, M.R. (1972). Optimal binary identification procedures. *SIAM Journal on Applied Mathematics*, *23*(2), 173–186.

Leung, H.K.N. (1991). A cost model to compare regression test strategies. In *Proceedings* of the Conference on Software Maintenance (pp. 201–208).

Mao, C., & Lu, Y. (2005). Regression testing for component-based software systems by enhancing change information. In *Proceedings* of the 12th Asia-Pacific Software Engineering Conference (APSEC'05) (p. 611).

Moore, E.F., & Shannon, C.E. (1956a). Reliable circuits using less reliable relays, Part I. *Journal of the Franklin Institute*, *262*, 191–208.

Moore, E.F., & Shannon, C.E. (1956b). Reliable circuits using less reliable relays, Part II. *Journal of the Franklin Institute*, *262*, 281–298.

Rothermel, G. (2001). Prioritizing test cases for regression testing. *IEEE Transactions on Software Engineering*, *27*(10), 929–948.

Shannon, C.E. (1948, July, October). A mathematical theory of communication. *Bell System Technical Journal*, *27*, 379–423, 623–656.

Sobel, M., & Groll, P.A. (1966). Binomial group-testing with an unknown proportion of defectives. *Technometrics*, *8*(4), 631–656.

Tsai, W.T. (2001). End-to-end integration testing design. In *Proceedings* of the 25th Annual International Computer Software and Applications Conference. (COMPSAC 2001) (pp. 166–171).

von Neumann, J. (1952). Probabilistic logics and the synthesis of reliable organisms from unreliable components. In C.E. Shannon & J. McCarthy (Eds.), *Annals of Mathematics Studies*, (pp. 45-98). *Automata Studies* (No. 34). Princeton, NJ: Princeton University Press.

Weyuker, E.J. (1998). Testing component-based software: A cautionary tale. *IEEE Software*, *15*(5), 54–59.

White, L.J. (1992). A firewall concept for both control-flow and data-flow in regression integration testing. In *Proceedings* of the IEEE Conference on Software Maintenance (pp. 262–271).

2003 - 2009 Sponsored Research Topics

Acquisition Management

- Acquiring Combat Capability via Public-Private Partnerships (PPPs)
- BCA: Contractor vs. Organic Growth
- Defense Industry Consolidation
- EU-US Defense Industrial Relationships
- Knowledge Value Added (KVA) + Real Options (RO) Applied to Shipyard Planning Processes
- Managing Services Supply Chain
- MOSA Contracting Implications
- Portfolio Optimization via KVA + RO
- Private Military Sector
- Software Requirements for OA
- Spiral Development
- Strategy for Defense Acquisition Research
- The Software, Hardware Asset Reuse Enterprise (SHARE) repository

Contract Management

- Commodity Sourcing Strategies
- Contracting Government Procurement Functions
- Contractors in 21st Century Combat Zone
- Joint Contingency Contracting
- Model for Optimizing Contingency Contracting Planning and Execution
- Navy Contract Writing Guide
- Past Performance in Source Selection
- Strategic Contingency Contracting
- Transforming DoD Contract Closeout
- USAF Energy Savings Performance Contracts
- USAF IT Commodity Council
- USMC Contingency Contracting

Financial Management

- Acquisitions via leasing: MPS case
- Budget Scoring
- Budgeting for Capabilities Based Planning
- Capital Budgeting for DoD
- Energy Saving Contracts/DoD Mobile Assets
- Financing DoD Budget via PPPs
- Lessons from Private Sector Capital Budgeting for DoD Acquisition Budgeting Reform
- PPPs and Government Financing
- ROI of Information Warfare Systems
- Special Termination Liability in MDAPs
- Strategic Sourcing
- Transaction Cost Economics (TCE) to Improve Cost Estimates

Human Resources

- Indefinite Reenlistment
- Individual Augmentation
- Learning Management Systems
- Moral Conduct Waivers and First-tem Attrition
- Retention
- The Navy's Selective Reenlistment Bonus (SRB) Management System
- Tuition Assistance

Logistics Management

- Analysis of LAV Depot Maintenance
- Army LOG MOD
- ASDS Product Support Analysis
- Cold-chain Logistics
- Contractors Supporting Military Operations
- Diffusion/Variability on Vendor Performance Evaluation
- Evolutionary Acquisition
- Lean Six Sigma to Reduce Costs and Improve Readiness

- Naval Aviation Maintenance and Process Improvement (2)
- Optimizing CIWS Lifecycle Support (LCS)
- Outsourcing the Pearl Harbor MK-48 Intermediate Maintenance Activity
- Pallet Management System
- PBL (4)
- Privatization-NOSL/NAWCI
- RFID (6)
- Risk Analysis for Performance-based Logistics
- R-TOC Aegis Microwave Power Tubes
- Sense-and-Respond Logistics Network
- Strategic Sourcing

Program Management

- Building Collaborative Capacity
- Business Process Reengineering (BPR) for LCS Mission Module Acquisition
- Collaborative IT Tools Leveraging Competence
- Contractor vs. Organic Support
- Knowledge, Responsibilities and Decision Rights in MDAPs
- KVA Applied to Aegis and SSDS
- Managing the Service Supply Chain
- Measuring Uncertainty in Eared Value
- Organizational Modeling and Simulation
- Public-Private Partnership
- Terminating Your Own Program
- Utilizing Collaborative and Three-dimensional Imaging Technology

A complete listing and electronic copies of published research are available on our website: www.acquisitionresearch.org

THIS PAGE INTENTIONALLY LEFT BLANK

Initial Distribution List

1. Defense Technical Information Center — 2
   8725 John J. Kingman Rd., STE 0944; Ft. Belvoir, VA 22060-6218
2. Dudley Knox Library, Code 013 — 2
   Naval Postgraduate School, Monterey, CA 93943-5100
3. Research Office, Code 09 — 1
   Naval Postgraduate School, Monterey, CA 93943-5138
4. Robert N. Beck — 1
   Dean, GSBPP
   E-mail: rnbeck@nps.edu
5. Bill Gates — 1
   Associate Dean for Research, GB
   E-mail: bgates@nps.edu
6. Karl D. Pfeiffer — 1
   Assistant Professor, IS
   E-mail: pfeiffer@nps.edu
7. Valery A. Kanevsky — 1
   Research Professor, IS
   E-mail: vakanevs@nps.edu
8. Thomas J. Housel — 1
   Professor, IS
   E-mail: tjhousel@nps.edu

Copies of the Acquisition Sponsored Research Reports may be printed from our website www.acquisitionresearch.org