The Hybrid Automated Reliability Predictor

Joanne Bechta Dugan,* Kishor S. Trivedi†
Duke University, Durham, North Carolina

and

Mark K. Smotherman,‡ Robert M. Geist§
Clemson University, Clemson, South Carolina

In this paper, we present an overview of the hybrid automated reliability predictor (HARP), under development at Duke and Clemson Universities. The HARP approach to reliability prediction is characterized by a decomposition of the overall model into distinct fault-occurrence/repair and fault/error-handling submodels. The fault-occurrence/repair model can be cast as either a fault tree or as a Markov chain and is solved analytically. Both exponential and Weibull time to failure distributions are allowed. There are a variety of choices available for the specification of the fault/error-handling behavior that may be solved analytically or simulated. Both graphical and textual interfaces are provided to HARP.

Nomenclature

Acronyms

ARIES = automated reliability estimation system, see Refs. 20, 21
CARE = computer-aided reliability estimation, see Refs. 4, 19, 32
ESPN = extended stochastic Petri net, see Refs. 15, 36
FEHM = fault/error-handling model
FORM = fault occurrence and repair model
FTMP = fault tolerant multiprocessor, see Ref. 23
HARP = hybrid automated reliability predictor, see Refs. 16-18, 28
SIFT = software implemented fault tolerance, see Ref. 24

FEHM-Related Symbols

c = probability of permanent coverage before interfering fault occurs
$\hat{c}$ = probability of reaching C exit of FEHM model (solved in isolation)
C = FEHM model exit representing permanent coverage
$F_{Y_C}(\tau)$ = conditional distribution of time to C exit ($= P_{IC}(\tau)/\hat{c}$)
$F_{Y_R}(\tau)$ = conditional distribution of time to R exit ($= P_{IR}(\tau)/\hat{r}$)
$F_{Y_S}(\tau)$ = conditional distribution of time to S exit ($= P_{IS}(\tau)/\hat{s}$)
I = FEHM model entry point
n = probability of interfering fault occurring before another exit is reached
N = FEHM model exit representing near-coincident fault
$P_{IC}(\tau)$ = distribution of time to C exit of FEHM model (solved in isolation)
$P_{IR}(\tau)$ = distribution of time to R exit of FEHM model (solved in isolation)
$P_{IS}(\tau)$ = distribution of time to S exit of FEHM model (solved in isolation)
r = probability of transient restoration before interfering fault occurs
$\hat{r}$ = probability of reaching R exit of FEHM model (solved in isolation)
R = FEHM model exit representing transient recovery
s = probability of single-point failure before interfering fault occurs
$\hat{s}$ = probability of reaching S exit of FEHM model (solved in isolation)
S = FEHM model exit representing single-point failure
Y_C = time to reach the C exit given that the C exit is reached (random variable)
Y_R = time to reach the R exit given that the R exit is reached (random variable)
Y_S = time to reach the S exit given that the S exit is reached (random variable)

Received May 17, 1985; revision received Oct. 16, 1985.

*Assistant Professor, Department of Computer Science.
†Professor, Department of Computer Science, Center for Computer Systems Analysis.
‡Associate Professor, Department of Computer Science.
§Assistant Professor, Department of Computer Science.

I. Introduction

THE demand for improved methods of predicting the reliability of fault-tolerant systems has certainly kept pace with the demand for the systems themselves. A reliability prediction tool must not only be sophisticated enough to analyze life-critical systems, but practical enough for the novice analyst to use. In this paper, we describe the key features of the hybrid automated reliability predictor (HARP), a package being developed with both goals in mind.

In this section we provide an introduction to reliability analysis and to the problems inherent in the analysis of systems that possess high reliability requirements. We also present one important characteristic of the HARP model, a *behavioral decomposition* of the overall model into two distinct submodels. The next two sections discuss the two submodels, Sec. IV the merging of the submodels, Sec. V the numerical solution methods and parametric sensitivity prediction, and Sec. VI several examples.

A. Methods for Reliability Analysis

Reliability is the most common measure of the effectiveness of a fault-tolerant system. There are three basic approaches to its evaluation: life testing, simulation, and analytic modeling. Life testing consists of observing n copies of the system and measuring the times to failure of the first m copies, or measuring the times to failure of those copies that have failed

by some specified time. Using these measurements, one can estimate the system reliability with certain levels of confidence.[1] When the expected lifetime of a component is long, or the component is expensive, life testing becomes unreasonable. In this case reliability estimation must depend on abstracting models of the system and analyzing these models.[2]

A simulation can include any level of detail and is thus flexible, but many replications of the simulation are needed to insure accuracy. In life-critical applications where reliability on the order of 0.999 999 999 is required, 1×10^{12} simulation replications might well be necessary! We are generally limited then to an analytic approach.

The first serious attempts at solving analytic models for reliability prediction were based on a combinatorial method that assumed the system was composed of a series of independent subsystems. Each subsystem was assumed to be a particular instance of hybrid redundancy and was solved separately using a known expression;[3] the reliability expressions for the subsystems were then multiplied to predict system reliability. CARE,[4] REL,[5] and RMS[6] are examples.

Not all systems can be broken down into a series of smaller subsystems, each of which operates independently. In such cases, combinatorial methods have been superseded by the more general Markov methods that collectively form the basis of most current models.[7] Such methods use continuous-parameter Markov chains in which a state typically represents the number of operational components in the system. The reliability of the system is then (conceptually) the sum of the probabilities for operational states.

B. Behavioral Decomposition in Reliability Modeling

Systems with high reliability requirements are designed with a great degree of fault tolerance. These systems make extensive use of redundancy in both hardware and software, have complex recovery management techniques, and are highly reconfigurable. A large state space is necessary to model each possible configuration and condition, especially if the analyst wishes to include details of the fault-recovery mechanisms.

The most common solution to the large state-space problem consists of physically or structurally dividing the system into smaller subsystems (e.g., processors, memory units, buses), solving the subsystems separately, and then combining the subsystem solutions to obtain the overall system solution. Decomposition and aggregation are the complementary activities of separating and combining parts of the system to facilitate analysis.[8] If we can assume that the subsystems' fault-tolerant behaviors are mutually independent, then a decomposition into subsystems, separate analysis of subsystems, and aggregation to obtain the final solution can be used. The usefulness of this approach diminishes as systems become more highly integrated.

An alternative to the above structural decomposition may be called *behavioral decomposition*.[9] We observe that the fault-occurrence/repair behavior of a system is composed of relatively infrequent events, while fault/error-handling behavior is composed of relatively frequent events. Faults may occur over periods of weeks or even years, but detection and recovery may take only seconds. Therefore, it is desirable to decompose along temporal rather than structural lines. In this way, we can separately analyze the fault/error-handling behavior and then reflect its effectiveness in an aggregate model by one or more parameters. It is also then possible to use different model types and solution techniques for the submodels.

Thus, the overall reliability model is decomposed into fault-occurrence/repair and fault/error-handling submodels. The fault-occurrence/repair model contains information about the structure of the hardware redundancy, the fault arrival processes, and manual (off-line) repair. The fault/error-handling model (often called the *coverage* model) allows for permanent, intermittent, and transient faults,[10] and models the (on-line) recovery procedure necessary for each type. These submodels are specified separately and are combined automatically for the solution of the overall model.

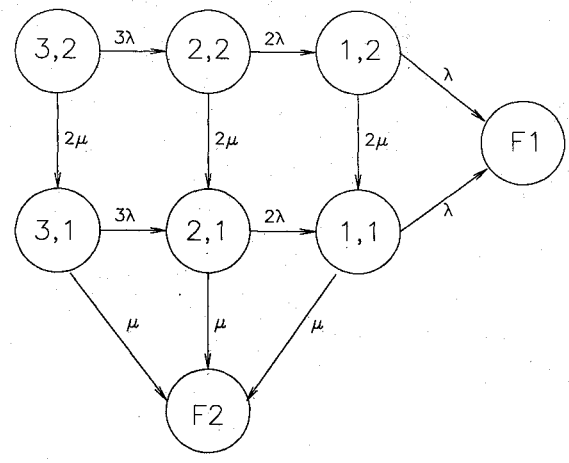

Fig. 1 "Perfect coverage" Markov chain representation of a three-processor, two-bus system (λ failure rate of the processors, μ failure rate of the buses).

II. Fault-Occurrence/Repair Model

The fault-occurrence/repair model (FORM) contains information about the structure of the system (how many components of what type interconnected in what way) and about the fault arrival and repair processes (how often does each component type fail and how long does it take to fix it). This information can be entered as either a Markov chain or a fault tree, depending on whether a stochastic or a combinatorial representation of the system is more appropriate. A fault tree representation of the system is often more concise than the corresponding Markov chain, but a Markov chain can model more complex system behavior, for example, sequence dependent failures or repairable systems (where there is not an independent repair crew for each component).

A. Markov Chain Representation of FORM

A Markov chain is entered as a state-transition diagram, in which each state represents a particular operational configuration of the system. Transitions between states represent units failing and being repaired. Additionally, failure states represent various configurations that fall below the minimum configuration necessary for an operational system. The Markov chain FORM model represents system configurations and is, in a sense, a "perfect coverage" model of the system. The part of the model representing "imperfect coverage" is specified separately. As an example, Fig. 1 shows the Markov chain representation of a nonrepairable three-processor, two-bus system in which one component of each type is necessary for the system to be operational. In Fig. 1, λ is the failure rate of processors and μ the failure rate of the buses. The states are labeled with an ordered pair, where the first element of the pair represents the number of operational processors and the second the number of operational buses.

An arc is symbolically labeled with an expression containing symbols,[¶] and constant multipliers; these symbols may be connected by the operations of addition, subtraction, and multiplication. One level of parenthesization is allowed. In our three-processor, two-bus example, the labels are of the form: constant × failure rate. An arc between states i, j and $i-1, j$ is labeled with the value $i \times \lambda$ (where λ is the failure rate of component type 1). Likewise, an arc between states i, j and

[¶]Failure rate transitions are denoted by a single failure "rate" variable (i.e., λ or μ) even though HARP does not require the failure rates to be constant. The specification of the failure distribution as either exponential or Weibull is done at the time of the run.

$i, j-1$ is labeled with the value $j \times \mu$ (where μ is the failure rate of component type 2). Although most transitions will be of this type, transitions between arbitrary pairs of states with more general labels are certainly permitted.

If the modeler desires a comparison of the probabilities of exhaustion of n different component types, then the failure state may be divided into n different failure states. These "exhaustion of redundancy" failure states are labeled F*, where * represents the index of the component type. In Fig. 1, state F1 represents failure of the processor cluster and state F2 the failure of the bus cluster. The "exhaustion of redundancy" failure states are also used in the calculation of bounds (see Sec. V).

Although HARP was designed for the separate specification of the "perfect coverage" Markov chain and the corresponding FEHM models, the direct specification of an arbitrary Markov chain is allowed. It will be solved without alteration.

B. Fault Tree Representation of FORM

A fault tree is a model that graphically and logically represents the various combinations of events occurring in a system that may lead to system failure.[11] The fault tree is structured so that the combination of events that lead to the undesired top event (system failure) is shown below the top event and is logically related to the top event by logic gates. The fundamental logic gates of fault trees allowed by HARP are the AND, OR, and K/N gates.

The input events to each logic gate may also be outputs of other logic gates at a lower level; each event is decomposed into lower events until the basic causes of the faults are reached. These "basic events" appear as circles on the bottom of the fault tree and represent the limit of resolution of the fault tree.

If system failure is caused by the occurrence of one of many basic events, these events are input to an OR gate whose output then represents system failure. If the system fails only when all events occur, these events are input to an AND gate instead. A K/N gate is used when the occurrence of K or more of the N possible events cause failure.

After the fault tree has been graphically (or textually) input to HARP, it is internally converted to a "perfect coverage" Markov chain (as was described in the previous section) for solution. All sequences of basic events that leave the system operational are enumerated; each combination becomes a state in the Markov chain. Figure 2 shows the fault tree representation of the system whose "perfect coverage" Markov chain is shown in Fig. 1. The basic events are component failures; each basic event is labeled with the component type it represents. The top event, system failure, is caused by the failure of both buses (component type 2) OR by the failure of all three processes (component type 1). The failure rate symbols[¶] for each component are specified when the fault tree is activated.

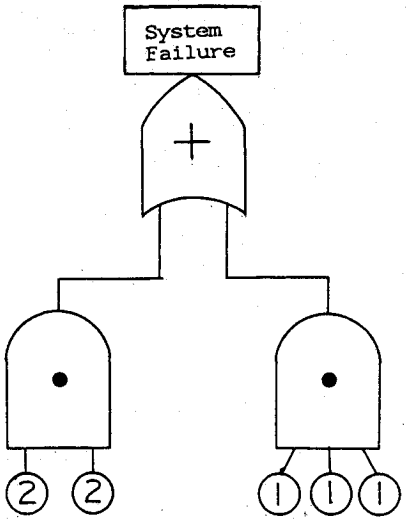

Fig. 2 Fault tree representation of the three-processor, two-bus system (a basic event is labeled with the type of component it represents).

In order to reduce the size of the resulting Markov chain, HARP allows the combination of identical components into single basic events. A multiple basic event is labeled with an expression of the form $m \times n$, representing m replications of component type n. (See Fig. 9 below for an example.) This combination of functionally identical components considerably reduces the size of the resulting Markov chain. Suppose a fault tree has j basic events, each with a replication factor of k_i. If every component were required to fail before the system failed, the resulting Markov chain using the multiple basic events would have $\Pi(k_i+1)-1+j$ states. If the basic events were all separate and there were then Σk_i basic events, the resulting Markov chain would have $2^{\Sigma k_i}$ states. Consider such a system in which there are five basic events, each with a replication factor of three. The Markov chain resulting from the tree with multiple basic events would have 1028 states, while the fault tree without multiple basic events would have $2^{15} = 32{,}768$ states. We are currently working on ways to further reduce the size of the Markov chain that is automatically generated from a fault tree.

III. Fault/Error-Handling Model

A fault/error-handling model is designed to capture the sequence of events that occur within the system once a fault occurs. A fault may be permanent (always present and capable of producing errors, e.g., a broken connection), transient (present for only a short time, e.g., a glitch in the power line), or intermittent (always present, but not always active, e.g., a loose connection). A typical recovery process could include such events as self-testing,[10] fault or error detection,[12] rollback,[13] fault isolation and reconfiguration.[13] Thus, the fault/error-handling model includes aspects of both the physical fault behavior and the system recovery mechanisms.

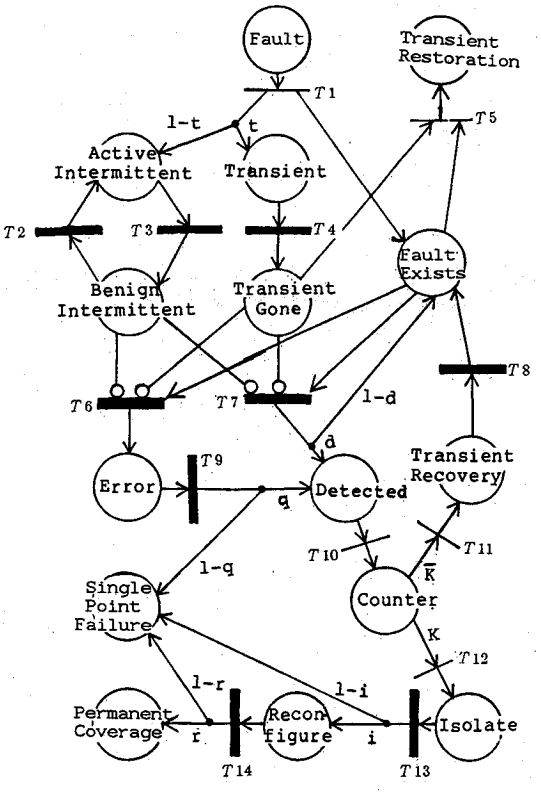

Fig. 3 ESPN single-fault coverage model.

The general structure of the fault/error-handling model (FEHM) is a single-entry, (up to) four-exit model entered when a fault occurs. The four exits represent the four possible outcomes of the attempted system recovery. The transient restoration exit R represents the correct recognition of, and recovery from, a transient fault. The permanent coverage exit C represents the successful reconfiguration of the system to eliminate a permanent, intermittent, or transient fault (mistaken as permanent). The third and fourth exits from the fault/error-handling model represent system failure. The single point failure exit S represents a single fault causing the system to fail without the interference of a second fault. The near-coincident[14] fault exit N is taken when a second interfering fault occurs before another exit is reached. The coverage factors r, c, s, and n are derived from the FEHM model and the near-coincident fault rate. (Section IV discusses the near-coincident fault rate that links the FEHM solution with the FORM model.)

A variety of choices is possible for the specification of the FEHM model, ranging from constant exit probabilities to a detailed ESPN (extended stochastic Petri net)[15] model. The chosen FEHM model is solved in (near) isolation and a coverage factor for each of the four exits are derived. These coverage factors are reflected into the FORM model, which is then solved for the reliability of the system. Some of the choices for the fault/error-handling model are shown in Fig. 3-5 and are described in the following subsections.

A. HARP Default ESPN Model

The HARP default ESPN model,[15-18] shown in Fig. 3, models three aspects of a fault recovery process: physical fault behavior, transient recovery, and permanent recovery. The fault behavior model captures the physical status of the fault, such as whether the fault is active or benign (if permanent or intermittent) or whether the fault still exists (if transient). Once the fault is detected, it is temporarily assumed to be transient and an appropriate recovery procedure may commence. The transient recovery procedure may be attempted more than once. If the detection/recovery cycle is repeated too many times, a permanent recovery procedure (isolation/reconfiguration) is invoked. If the permanent recovery is successful, the system is again operating correctly, although in an operationally degraded state. A more detailed explanation of ESPNs and this default model appears in the Appendix.

The user inputs to this model are the distribution of time for each activity and the probabilities of correct error detection, fault detection, fault isolation, and reconfiguration. (The distributions need not be exponential.) The user must specify the number of attempts at transient recovery, the percentage of faults that are transient, and, since this model is simulated for solution, the confidence level and percent error desired.

For this model, the transient restoration exit probability $\hat{r}$ is the probability of a token reaching the place labeled transient restoration; $\hat{c}$, coverage exit probability, is the probability of a token reaching the place labeled permanent coverage; and $\hat{s}$ is the probability of a token reaching the place labeled single-point failure. The coverage factors r, c, n, and s are derived from these exit probabilities and the relative passage time to the three exits. This derivation is discussed in Sec. IV.C.

B. CARE Coverage Model

Another option for the FEHM is a Markov version of the CARE III single-fault model[19] shown in Fig. 4. The CARE III coverage model, like the HARP, can be used to model permanent, transient, and intermittent faults. In the active state, a fault is both detectable and capable of producing an error. Once an error is produced, if it is not detected, it propagates to the output and causes system failure. If the fault (error) is detected, the faulty element is removed from service with probability P_A or P_B. With the complementary probabilities, the element is returned to service following the detection of the fault. This action is based on the belief that the detected fault was transient. Note that both states A_D and B_D are "instantaneous" (i.e., zero holding time) states. The model is solved analytically for the Laplace transform of the distribution of the time to exit or for the exit probabilities and first few moments of the time to exit (depending on whether the near-coincident fault rate is constant).

C. ARIES Transient Recovery Model

The ARIES transient fault recovery model[20,21] represents a multiphase recovery process that executes n successive recovery phases. (See Fig. 5.) Transition to the next phase takes place if the present phase is not effective; the duration of each phase is constant. The recovery process terminates and normal processing begins if successful recovery is achieved in the present phase. If transient recovery is unsuccessful after all n phases, then a permanent recovery process is initiated. The

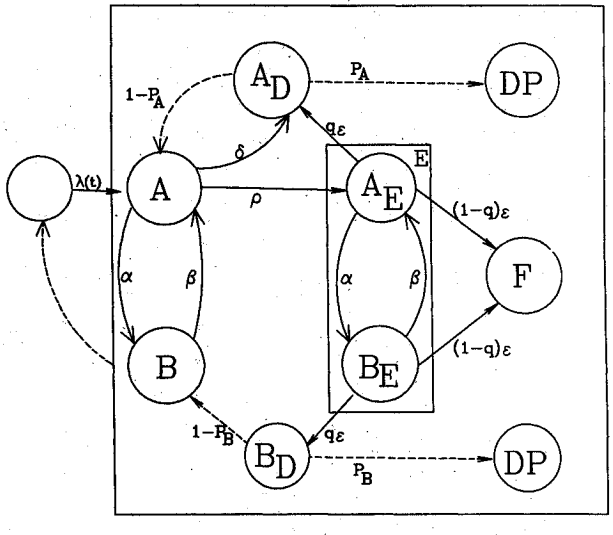

Fig. 4 Markov version of the CARE III coverage model: A = active, B = benign, D = detected, E = error, F = failure, DP = detected as permanent (nontransient).

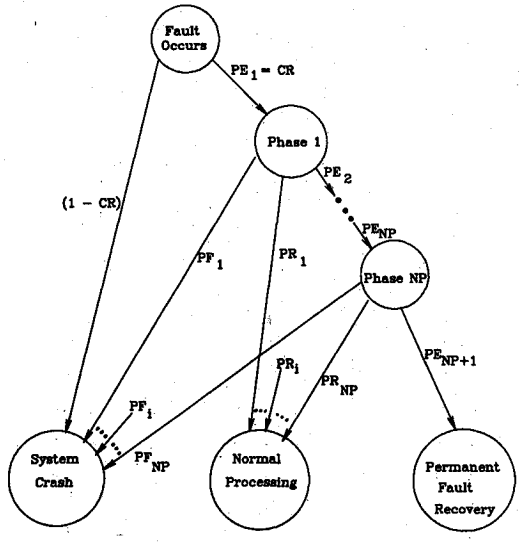

Fig. 5 ARIES transient fault recovery model.

original ARIES model included the concept of interfering faults in the sense that a failure in the recovery hardware during an attempted recovery causes system failure. The HARP version of the ARIES model extends the concept of interference to include near-coincident faults as discussed in Sec. IV.B.

D. Probabilities and Distributions

Under this option, the user merely specifies an exit probability for the transient restoration, permanent coverage, and single-point failure exits. (That is, the user provides $\hat{r}$, $\hat{c}$, and $\hat{s}$, which sum to one.) Then, for each nonzero probability, a distribution of time to exit is specified, choosing from constant, uniform, exponential, hypoexponential, hyperexponential, gamma, and Weibull. The distribution may also be given as empirical data. These probabilities and distributions are to be given, disregarding the occurrence of a second, near-coincident fault. The coverage factors are then automatically derived from these data. This is discussed in detail in Sec. IV.C.

E. Probabilities and Moments

This model allows the user to specify, for the transient recovery, permanent coverage, and single-point failure exits, a probability of reaching the particular exit ($\hat{r}$, $\hat{c}$, and $\hat{s}$) and the first three moments of the time to exit. The coverage factors (r, c, s, and n) are automatically derived from the given data.

F. No Coverage Model

As a final option, the user may wish to assign the coverage values directly or ignore the notion of coverage completely. In the latter case, we assume that all faults are permanent and are perfectly covered.

IV. Combining FORM and FEHM Models

To predict the reliability of the overall model, the various FEHM models are solved in isolation for the exit probabilities $\hat{r}$ (transient restoration), $\hat{c}$ (permanent coverage), and $\hat{s}$ (single-point failure). The coverage factors r, c, and s are derived from these exit probabilities which are adjusted by the probability of reaching the exit before the occurrence of a second fault. Coverage factor n (near-coincident fault) is calculated as the probability of experiencing a second fault while in the coverage model. (We use $\hat{r}$, $\hat{c}$, and $\hat{s}$ to denote the exit probabilities of the FEHM models when solved in isolation and r, c, s, and n to denote the reduced exit probabilities adjusted for near-coincident faults. Thus, $\hat{r}+\hat{c}+\hat{s}=1$ and $r+c+s+n=1$). These coverage factors are then used to modify the Markov chain that represents the FORM model. Note that these probabilities are now state dependent, since the "second fault" process changes according to the current state in the FORM model. In a state that represents 100 active components, the probability of a second fault is substantially higher than in a state with only 3 active components.

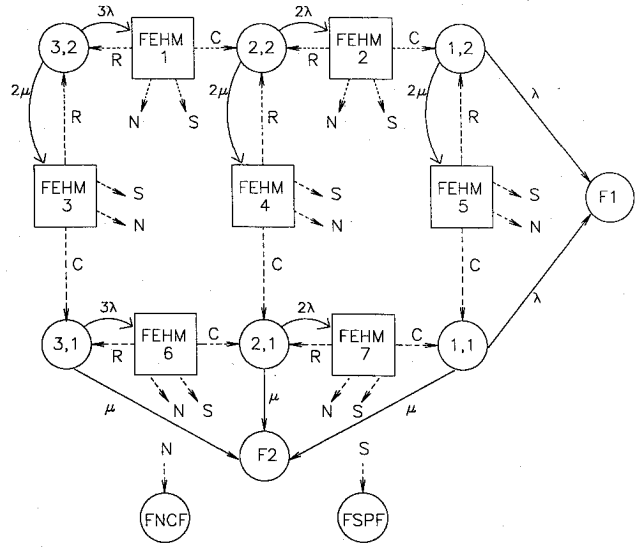

Fig. 6 Hierarchical state diagram representation of a three-processor, two-bus system (R = transient restoration, C = permanent coverage, S = single-point failure, N = near-coincident fault).

A. Automatic Incorporation of Imperfect Coverage

The possibility of imperfect fault coverage is automatically incorporated into the "perfect coverage" Markov chain in the following way. Associated with each component type in the system is a fault/error-handling model that describes the recovery behavior of that particular component. In the three-processor, two-bus system (Figs. 1 and 2), one FEHM model would be described to represent the fault/error-handling behavior of processor faults and a (possibly) different FEHM would be associated with bus faults. (See Fig. 6.) These FEHM models are solved in isolation for the exit probabilities for the three exits ($\hat{r}$, $\hat{c}$, and $\hat{s}$) and for the time needed to reach the exit. (The "time needed to reach the exit" is generally captured in a Laplace transform of the distribution or the first few moments of time to reach the exit.) The probabilities are then adjusted according to the probability of an interfering fault to produce (state-dependent) coverage probabilities that are then used to modify the transition rates in the Markov chain. Additionally, two failure states are added to the model, one to represent single-point failures and one near-coincident faults.

More specifically, assume that a fault of component type 1 in state (i, j) leads to state $(i-1, j)$ in the "perfect coverage" Markov chain. In the "imperfect coverage" Markov chain, this transition to state $(i-1, j)$ is completed with probability $c_{(i,j),(i-1,j)}$ and a transition to the single-point failure state occurs with probability $s_{(i,j),(i-1,j)}$. A transition back to state (i, j) occurs with probability $r_{(i,j),(i-1,j)}$ and a transition to the near-coincident failure state occurs with probability $n_{(i,j),(i-1,j)} = (1 - c_{(i,j),(i-1,j)} - r_{(i,j),(i-1,j)} - s_{(i,j),(i-1,j)})$.

This probability of imperfect coverage is thus incorporated into the Markov chain by reducing the rate of flow from state (i, j) to state $(i-1, j)$, by multiplying the rate (call it ν) by $c_{(i,j),(i-1,j)}$, and by adding arcs from state (i, j) to the failure states. These additional arcs represent flows of $\nu \times s_{(i,j),(i-1,j)}$ (to the single-point failure state) and $\nu \times n_{(i,j),(i-1,j)}$ (to the near-coincident fault failure state). An implied self-loop to state (i, j) occurs with a rate $\nu \times r_{(i,j),(i-1,j)}$. This is done for all failure arcs between the operational states.

The coverage failure states are differentiated from the exhaustion of the components failure states to enable a comparison of the respective failure probabilities. Figure 7 shows the imperfect coverage representation of the three-processor, two-bus system of Fig. 1, where FSPF represents the "single-point failure" state and FNCF the failure of the system caused by a "near-coincident fault." In this figure, the coverage factors are singly subscripted for ease of notation. The Markov chain of Fig. 7 is an approximation to the stochastic process represented by Fig. 6. For a discussion of the conservativeness of this approximation, see McGough et al.[22]

The modeler may wish to define a different FEHM model for each individual failure arc, rather than for each component type. This is accomplished by providing the hierarchical state diagram (Fig. 6) or the "imperfect coverage" Markov chain (Fig. 7) directly and, when prompted, providing the FEHM model and near-coincident fault rate for each different coverage symbol.

B. Near-Coincident Fault Rate

Since we are including consideration of a near-coincident fault, we need to know the rate at which such catastrophic faults occur for each FEHM model. Second-fault rates that bound the failure probability due to near-coincident faults can be determined automatically from the FORM model.

Conservatively, we may wish to assume that a second, near-coincident fault *anywhere in the system* (while attempting to handle a single fault) causes immediate system failure. The (all-inclusive) near-coincident fault rate is the failure rate from the target state. In our running example, for the FEHM between states (i, j) and $(i-1, j)$, the all-inclusive near-coincident fault rate is the sum of the outgoing arcs from state $(i-1, j)$, that is, $(i-1) \times$ (failure rate of component type 1) $+ j \times$ (failure rate of component type 2). Transitions representing repair do not contribute to the near-coincident fault rate.

Optimistically, we may wish to assume that only near-coincident faults of *the same component type* cause system failure (while attempting to handle a single fault). In the example, the (same-type) near-coincident-fault rate for the FEHM between states (i, j) and $(i-1, j)$ is $(i-1) \times$ (failure rate of component type 1).

More generally, the user may wish to define explicitly for each component those (other) components that can interfere with fault recovery. In this case, the (user-defined) near-coincident fault rate for each FEHM depends on the user input. For example, suppose we have a system consisting of three processors P1, P2, and P3 (all distinct with unique failure rates) and a bus B. Suppose further that the processors are connected (from a monitoring point of view) in a ring-type network, so that processor P1 detects errors and performs recovery for processor P2, processor P2 likewise monitors P3, and P3 monitors P1. Thus, a failure in processor P1 can interfere with recovery in processor P2. Similarly, a failure on processor P2 can interfere with recovery on P3, etc. Since the processors are connected by the data bus, a bus failure can interfere with recovery on any of the processors (the bus does not rely on any other component for its recovery). This behavior cannot be captured by either the all-inclusive or the same-type fault rates. For this example, the user would define P3 and the bus as the interfering components for P1, P1 and the bus as interfering components for P2, P2 and the bus as interfering components for P3, and no interfering components for the bus. This option is also useful for modeling recovery hardware failures. The recovery hardware can be defined as a distinct component type whose failure interferes with the recovery of the appropriate component types, but cannot alone cause system failure (i.e., the failure of the recovery hardware does not appear in the fault tree or the Markov chain representation of the FORM model).

Thus, the all-inclusive, same-type, and user-defined near-coincident fault rates for each instance of a fault/error-handling model are generated automatically. At program execution time, the user is asked which of the three rates should be used (or none at all). The all-inclusive and same-type near-coincident fault rates for the three-processor, two-bus system are shown in Table 1.

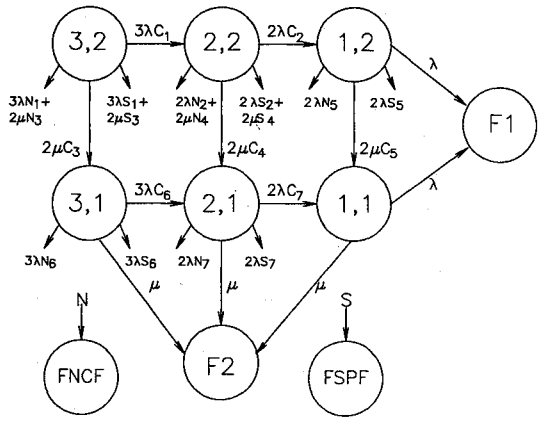

Fig. 7 Imperfect coverage Markov chain representation of a three-processor, two-bus system.

C. Computation of Coverage Factors

In this section, we demonstrate how the effect of a near-coincident fault[14,23,24] is incorporated into the traditional concept of coverage (the probability that the system can recover from a fault).[3] The probability of successful fault coverage involves two phenomena: the system must be able to recover from the fault and this recovery must occur before another fault can interfere. We discuss the method for combining these two aspects into the coverage factors r, c, s, and n in the following.

In a general coverage model, fault/error-handling begins (the submodel is entered) when a fault occurs in the system. Let us label the entry point of the coverage model I. There are three mutually exclusive exits from the submodel, labeled R, C, and S, representing transient restoration, permanent coverage, and single-point failure, respectively. Fault-handling completes (the submodel is exited) when the fault is handled (exits R and C) or when the system fails (exit S).

Let $P_{IC}(\tau)$ denote the probability of the system recovering to a degraded state (i.e., reaching the C exit from the FEHM model) in an amount of time $\leq \tau$ from the time of occurrence of the fault.** Likewise, $P_{IR}(\tau)$ represents the probability of successful transient restoration in time $\leq \tau$, and $P_{IS}(\tau)$ represents the probability of a single-point failure in time $\leq \tau$. These distributions represent the solution of the FEHM model *in isolation*, that is, without consideration of its relation to the FORM model. Let $\hat{r}$, $\hat{c}$, and $\hat{s}$ denote the probability of eventually reaching the appropriate exit; thus,

$$\hat{r} = \lim_{\tau \to \infty} P_{IR}(\tau), \quad \hat{c} = \lim_{\tau \to \infty} P_{IC}(\tau), \quad \hat{s} = \lim_{\tau \to \infty} P_{IS}(\tau)$$

Then, using the traditional notion of coverage (that is, ignoring the possibility of near-coincident faults), we could set the desired coverage factors to these limiting probabilities,[1,9,25,26]

$$r = \hat{r}, \quad c = \hat{c}, \quad s = \hat{s}$$

More realistically, it is not sufficient to know that the system will eventually recover. Recovery is successful only if it

**The distribution P_{IC} may be defective, in the sense that $\lim_{\tau \to \infty} P_{IC}(\tau) < 1$. P_{IR} and P_{IS} are also defective distributions, and $\lim_{\tau \to \infty} [P_{IC}(\tau) + P_{IR}(\tau) + P_{IS}(\tau)] = 1$.

Table 1 Near-coincident fault (NCF) rates for automatically embedded FEHMs for three-processor, two-bus system

State transition arc	Failure of which component type	FEHM no.	All-inclusive NCF rate	Same-type NCF rate
(3,2) to (2,2)	1 (processor)	1	$2\lambda + 2\mu$	2λ
(3,2) to (3,1)	2 (bus)	3	$3\lambda + \mu$	μ
(2,2) to (1,2)	1 (processor)	2	$\lambda + 2\mu$	λ
(2,2) to (2,1)	2 (bus)	4	$2\lambda + \mu$	μ
(3,1) to (2,1)	1 (processor)	6	$2\lambda + \mu$	2λ
(2,1) to (1,1)	1 (processor)	7	$\lambda + \mu$	λ
(1,2) to (1,1)	2 (bus)	5	$\lambda + \mu$	μ

is completed *before* the occurrence of a second, interfering fault. Thus, setting the coverage factors to the limiting value of the exit distributions is optimistic, especially if the recovery time is long in comparison to the fault arrivals. These limiting values must then be "adjusted" to account for the time needed to recover from a fault.

Let X be a random variable that represents the time between occurrences of interfering faults, where $F_X(\tau) = 1 - e^{-\gamma\tau}$. Let Y_R, Y_C, and Y_S be random variables representing the time to reach the corresponding exit of the FEHM model (conditioned on actually reaching the exit), where $F_{Y_R}(\tau)$, $F_{Y_C}(\tau)$, and $F_{Y_S}(\tau)$ are the conditional distributions of the time to exit,

$$F_{Y_R}(\tau) = \frac{P_{IR}(\tau)}{\hat{r}}, \quad F_{Y_C}(\tau) = \frac{P_{IC}(\tau)}{\hat{c}}, \quad F_{Y_S}(\tau) = \frac{P_{IS}(\tau)}{\hat{s}}$$

Then, the coverage factor c is the probability that permanent fault recovery is completed before a second fault occurs,[27]

$$c_\gamma = \text{Prob}[\text{time to reach C exit} < X]$$

$$c_\gamma = \int_0^\infty e^{-\gamma\tau}\, dP_{IC}(\tau)$$

$$= \int_0^\infty e^{-\gamma\tau}\, \hat{c}\, dF_{Y_C}(\tau) \qquad = \hat{c} \lim_{u\to\gamma} F_{Y_C}{}^*(u)$$

where $F_{Y_C}{}^*(u)$ is the Laplace-Stieltjes transform of the random variable Y_C. For many FEHM models like the CARE III single-fault model (Fig. 4), the derivation of the Laplace transform of the time-to-exit distribution is not difficult, especially since we are not interested in inverting to the time domain. However, for FEHM models that are simulated (i.e., ESPNs) for solution, the Laplace transform of the time to exit is not an easily estimated parameter and must be approximated from the simulation data. Fortunately, this approximation produces good results.[17,18]

If we perform a Taylor series expansion of the Laplace-Stieltjes transform we have

$$c_\gamma = \hat{c}\left[1 - \gamma E[Y_C] + \frac{\gamma^2}{2!}E[Y_C^2] - \frac{\gamma^3}{3!}E[Y_C^3] + \cdots\right]$$

With entirely analogous expressions for r_γ and s_γ.

Thus, the information needed from the fault/error-handling model is, for each exit, the exit probability and the first few moments of the time-to-exit distribution. The calculation of the state-dependent near-coincident fault factor from these three factors is then

$$n_\gamma = 1 - (c_\gamma + r_\gamma + s_\gamma)$$

Since $(\gamma^n/n!)E[Y_C^n]$ normally approach zero very rapidly, only the first few moments of the recovery time distribution are usually necessary.[14,17] Note that the use of an odd number of moments for r_γ and c_γ and an even number of moments for s_γ leads to conservative reliability estimates.

In the case in which the failure process has a Weibull distribution[1] $[\gamma(t) = \lambda_0\alpha t^{\alpha-1};\ t > 0]$, the Taylor series expansion generalizes[28] to[‡‡]

$$c_{\gamma(t)} = \hat{c}\left[1 - \gamma(t)E[Y_C] + \left(\gamma^2(t) - \frac{\alpha-1}{t}\gamma(t)\right)\frac{E[Y_C^2]}{2!} - \left(\gamma^3(t) - \frac{3(\alpha-1)}{t}\gamma^2(t) - \frac{(\alpha-1)(\alpha-2)}{t^2}\gamma(t)\right)\frac{E[Y_C^3]}{3!} + \cdots\right] \qquad t > 0$$

‡‡The time-dependent coefficient of $E[Y_C^n]/n!$ is given by $g^{(n)}(t)/g(t)$, where $g(t) = e^{-\lambda_0 t^\alpha}$.

And if the near-coincident fault rate is a sum of Weibull terms $[\gamma(t) = \Sigma\lambda_i\alpha_i t^{\alpha_i - 1}]$, the coefficient of the second moment in the Taylor series expansion becomes

$$\left[\gamma^2(t) - \left(\Sigma\lambda_i\alpha_i(\alpha_i - 1)t^{\alpha_i - 2}\right)\right]$$

and the coefficient of the third moment is given by

$$\left\{\gamma^3(t) - 3\gamma^2(t)\left[\Sigma\lambda_i\alpha_i(\alpha_i - 1)t^{\alpha_i - 2}\right] - \left[\Sigma\lambda_i\alpha_i(\alpha_i - 1)(\alpha_i - 2)t^{\alpha_i - 3}\right]\right\}$$

V. Numerical Solution Technique and Error Bounds

Once the FEHM models have been solved and the state-dependent coverage factors automatically inserted into the model, the Markov chain representation of the FORM model remains to be solved. The Markov chain (such as the one shown in Fig. 7) produces a general system of ordinary differential equations,

$$P'(t) = P(t)Q(t), \quad P(0) = P_I \tag{1}$$

where $P(t)$ is the probability (row) vector for the states in the Markov chain and $Q(t)$ the associated matrix of (possibly) time-dependent transition rates. Thus, the entry $q_{i,j}(t)$ denotes the transition rate from state i to state j, and $q_{ii}(t) = -\Sigma_{j\neq i} q_{ij}(t)$. This analytic model is then solved for the state probabilities $P_i(t)$ using a variation of an adaptive Runge-Kutta procedure, GERK.[29] The reliability (and unreliability) of the system are then given by the appropriate sum of the state probabilities,

$$R(t) = \sum_{i\epsilon\text{UP states}} P_i(t)$$

$$U(t) = \sum_{i\epsilon\text{DOWN states}} P_i(t)$$

Both the system reliability and unreliability are calculated separately and reported to improve accuracy. Additionally, the probabilities for each type of failure (exhaustion of redundancy, single-point failure, near-coincident faults) are reported separately. In the case where the failure states are not absorbing, the instantaneous availability of the system is reported.

We note again that we allow the transition rate matrix to have globally time-dependent entries; thus, Weibull time-to-failure distributions are possible. For closed fault-tolerant systems [$Q(t)$ is upper triangular; the aggregate model is acyclic], Weibull times to failure are properly modeled using this approach; however, "cold" spares (i.e., spares that do not fail until switched into operation) should not be combined with time-dependent failure processes because of an implied "good as old" effect.[9] (HARP will allow this combination, but will warn the user of the possible inaccuracy of the result.) If the user is interested in modeling systems with cold (or warm) spares or repairable systems [those with cycles in the $Q(t)$ matrix], the assumption of constant transition rates should be made.

Since many of the input parameters to the FORM model are not known exactly (i.e., coverage values from simulation are given as confidence intervals and the user may know only a range of values for the failure rates), HARP expects the input parameters to be expressed in terms of ranges of values, rather than point estimates. HARP produces upper and lower bounds on the system reliability based on these ranges of values, as well as the predicted reliability based on the nominal values.[17,28,30]

We approach the analysis of parametric errors by decomposing the original model into two simpler models that can be

combined to obtain bounding reliability estimates.[31] We first consider bounds on system failure probability caused by lack of sufficient redundancy (call this event A) and then consider bounds on the system failure probability because of imperfect coverage (call this event B). We find the overall system failure probability by use of the probability combining rules:

$$P[A \cup B] \leq \min\{1, P[A] + P[B]\}$$

$$P[A \cup B] \geq \max\{P[A], P[B]\}$$

The first rule will give us the conservative bound and the second rule will give us a complementary optimistic bound. Since the computations involved here are rather simple when compared with the solution of Eq. (1), we perform parametric sensitivity analysis using these bounds. HARP performs this bound analysis for the parametric variations provided by the user.

VI. Examples

A. The Three-Processor, Two-Bus System

Consider a three-processor, two-bus system that is operational as long as one bus and one processor are operational. The "perfect coverage" Markov chain is shown in Fig. 1 and the fault tree representation in Fig. 2. We assume that a bus interface unit (BIU) is responsible for detecting and handling faults on the bus. (For the sake of simplicity, the BIU is assumed to be fault free.) All single faults on the bus are assumed transient and are detected and corrected by using an error-correcting code. If an error appears many times within a short period of time, the offending bus is removed from the system; no single-point failures are possible for the bus cluster. From measurements and data analysis of the fault/error-handling behavior of the BIU, we assume that we have determined the first three moments of the time needed to correct a single error or to remove a failed bus from the system; thus, the "probabilities and moments" FEHM model is used for bus failures. The fault/error-handling behavior of the processor is much more complicated, and consists of multiple attempts at transient recovery, (possibly) followed by fault isolation and reconfiguration. The ESPN recovery model (Fig. 3) is used to model the processor fault/error-handling behavior.

After the user enters the description of the FORM model (Fig. 1 or 2) and the FEHM model parameters for the bus and processor (Tables 2 and 3), HARP automatically inserts a FEHM model for each arc representing a unit failing, as in Fig. 6. In this hierarchical diagram, FEHMs 1, 2, 6, and 7 are processor recovery models (Table 2) and FEHMs 3-5 are bus recovery models (Table 3). The FEHM models are solved for the state-dependent coverage factors shown in the "imperfect coverage" model of the system, see Fig. 7.

The Markov chain shown in Fig. 7 was solved twice, once for constant failure rates and once for Weibull time-to-failure distributions. In both cases, the rate parameters were equal: λ was 10^{-3} and μ 10^{-2}. In the second case, the failure rate was a linearly increasing function of time (α, the shape parameter, was 2). A plot of the predicted unreliabilities for a 10 h mission appears in Fig. 8. In this and the other examples in this paper, all-inclusive near-coincident fault rates were used. The solution of this model took (on a VAX 11/750 running Unix) 10.6 CPU s to solve the exponential case, including approximately 4 CPU s to simulate the FEHM model for processor failures. (The ESPN model needs to be simulated only once for a given set of input parameters, regardless of how many times the FEHM model is used.) Since the entire transition rate matrix needs to be re-evaluated at each time step (including the recalculation of coverage factors), the solution of a model with Weibull failure distributions can be slow. This example took 325 CPU s to solve.

Table 2 Bus fault/error-handling model for three-processor, two-bus system

Probabilities and moments
Transient restoration exit
Exit probability: 0.9000
First moment of time to exit: 0.660×10^{-2} s
Second moment of time to exit: 0.580×10^{-4} s^2
Third moment of time to exit: 0.780×10^{-6} s^3
Reconfiguration coverage exit
Exit probability: 0.1000
First moment of time to exit: 0.4500 s
Second moment of time to exit: 0.2500 s^2
Third moment of time to exit: 0.1750 s^3
Single-point failure exit
Exit probability: 0

B. A Flight Control System

As a second example, consider a flight control system consisting of five stages. Stages 1 and 2, the inertial reference and pitch rate stages, are triplicated sensors; the stage fails if

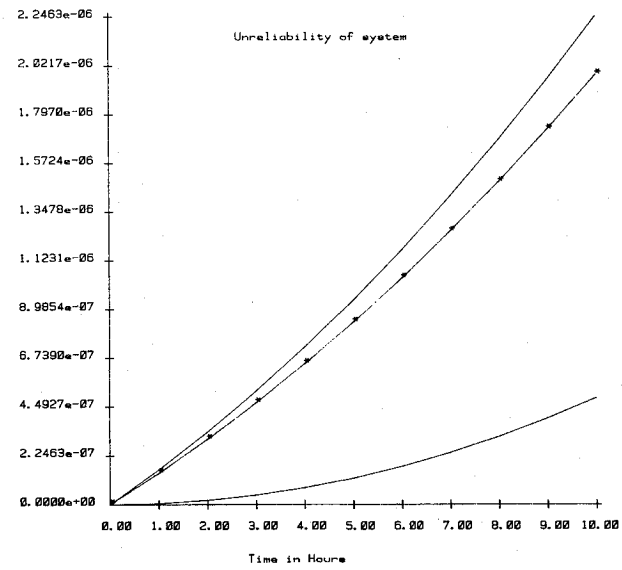

Fig. 8 Predicted unreliability for a 10 h mission of a three-processor, two-bus system (upper line represents a linearly increasing failure rate, lower line a constant failure rate).

Table 3 Processor fault/error-handling model for three-processor, two-bus system (HARP single-fault model)

Coverage input parameters	
Time	Distribution and parameters
ACTIVE transition	Uniform (0, 1)
BENIGN transition	Uniform (0, 0.5)
Transient lifetime	Exponential (100)
DETECT transition	Uniform (0, 0.4)
ERROR transition	Weibull (10.0, 2.5)
ERROR-DETECT transition	Weibull (50.0, 0.25)
ISOLATION transition	Normal (4.0, 1.0)
RECOVERY transition	Erlang (100.0, 2.0)
RECONFIGURATION transition	Normal (1.0, 0.5)
Other parameters	
Probability of fault detection by self-test: 0.95	
Probability of error detection: 0.95	
Probability of isolating detected fault: 0.95	
No. of recovery attempts: 5	
Probability of successful reconfiguration: 0.95	
Fraction of faults which are transient: 0.90	
Desired confidence level: 90%	
Allowable error: 10%	

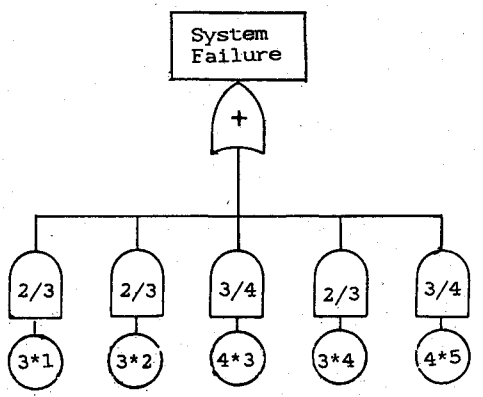

Fig. 9 System fault tree for flight control system of Sec. VI.B (a basic event labeled with $i * j$ represents i components of type j).

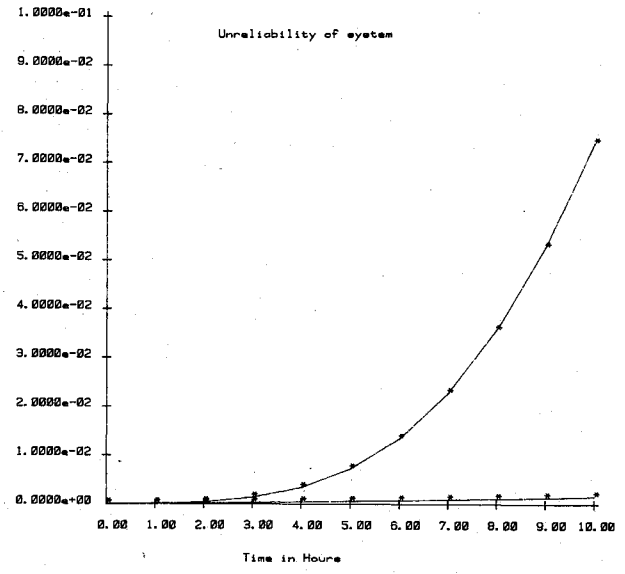

Fig. 10 Unreliability bands for 10 h mission of flight control system.

two of three modules fail. The computer (third) stage and bus (fifth) stage fail if three of the four modules fail. The fourth stage, the secondary actuator stage fails if two of the three modules fail. The system fault tree for this system is shown in Fig. 9. There is no fault/error-handling for stages 1, 2, or 4 and the FEHM models for stages 3 and 5 are both CARE III models (Fig. 4); the input parameters for these models are shown in Tables 4 and 5. The model was solved with the failure rates given in Table 6; the unreliability plot is shown in Fig. 10. The bounds analysis described in Sec. V was used to obtain the upper and lower bounds on system unreliability as the parameters vary over the specified ranges. (This example was adapted from the CARE III user's guide.[32]) The Markov chain that HARP automatically generates from this simple fault tree has 76 states. This model took approximately 15 min to describe interactively to HARP using the textual interface. It took 64 CPU s to solve this system for system unreliability for a 10 h mission time on a VAX 11/750 running under Unix.

C. A Repairable System

Figure 11 shows a diagram of a portion of a flight control system containing triplicated sensor/multiplexor modules whose outputs are majority voted. Two spare sensors and one spare multiplexor are provided, and failed components may be repaired as long as the system has not failed. The system is operational as long as two of the three sensor/multiplexor

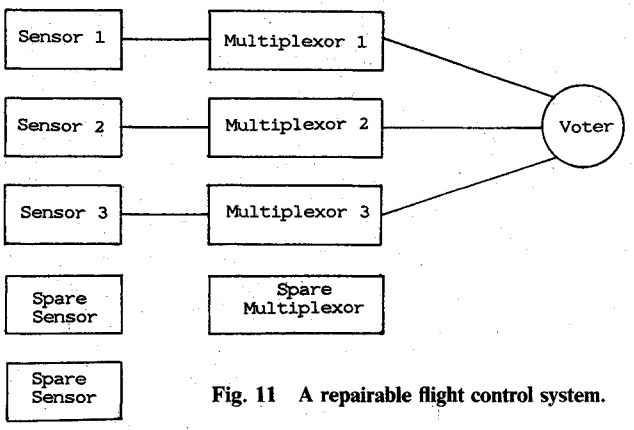

Fig. 11 A repairable flight control system.

Table 4 Fault/error-handling model for computer stage of flight control system (CARE single-fault model)

Probability of permanent: 0.1000
Probability of intermittent: 0.1000
Probability of transient: 0.8000
Permanent model parameters
δ: 360.0
ϵ: 3600.
ρ: 180.0
PA: 1.0000
Q: 0.9990
Intermittent model parameters
α: 2100.
β: 2100.
δ: 360.0
ϵ: 3600.
ρ: 180.0
PA: 1.0000
PB: 0.
Q: 0.9990
Transient model parameters
α: 0.3600×10^5
δ: 360.0
ϵ: 3600.
ρ: 180.0
PA: 1.0000
PB: 0.
Q: 0.9990

Table 5 Fault/error-handling model for bus stage of flight control system (CARE single-fault model)

Probability of permanent: 0.2000
Probability of intermittent: 0.2000
Probability of transient: 0.6000
Permanent model parameters
δ: 0.1000×10^5
ϵ: 0.000×10^{-2}
ρ: 0.
PA: 1.0000
Q: 1.0000
Intermittent model parameters
α: 2100.
β: 2100.
δ: 0.1000×10^5
ϵ: 0.1000×10^{-2}
ρ: 0.
PA: 1.0000
PB: 1.0000
Q: 1.0000
Transient model parameters
α: 0.360×10^5
δ: 0.1000×10^5
ϵ: 0.1000×10^{-2}
ρ: 0.
PA: 1.0000
PB: 1.0000
Q: 1.0000

subsystems and the voter are operational. (A failure of the voter is often called a "common mode" failure, since it brings the system down immediately.)

When an active component fails, a spare component is switched into active operation and the failed unit is returned to the spare pool when repaired. The spare units are assumed to be "cold" spares, that is, they do not fail while inactive. If no spares remain, when an active sensor fails its corresponding multiplexor is disconnected and thus cannot fail; however, if the multiplexor fails, the sensor is still operating and may still fail. There is only one technician available to repair the failed components. If more than one component is down at any given time, the component that is most crucial will be repaired first.

Figure 12 is the perfect coverage Markov chain of this system, and Table 7 describes the state space. The voter mechanism is capable of handling transient and permanent faults in the multiplexor. The transient recovery consists of two phases, a short delay (to allow the transient to disappear) followed by a retry. An ARIES transient fault recovery model is chosen to represent multiplexor fault/error-handling behavior; the parameters are shown in Table 8. All faults on sensors are assumed permanent; fault/error-handling consists of switching out the faulted part and switching in a spare (if available); the switching time is hyperexponentially distributed. The parameters for the sensor fault/error-handling model are listed in Table 9. There is no fault/error-handling possible for the voter mechanism, since the system fails catastrophically upon voter failure.

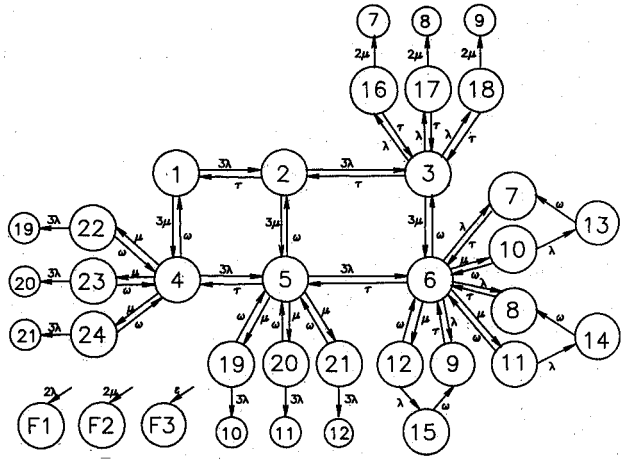

Fig. 12 Markov chain representation of repairable flight control system: a transition to state F1 (exhaustion of sensors) is possible in states 7-18; a transition to state F2 (exhaustion of multiplexors) is possible in states 7-15 and 19-24; a transition to state F3 (voter failure) is possible from every state (λ = failure rate, τ = repair rate for sensors, μ = failure rate, ω = repair rate for multiplexors, ϵ = failure rate of the voter).

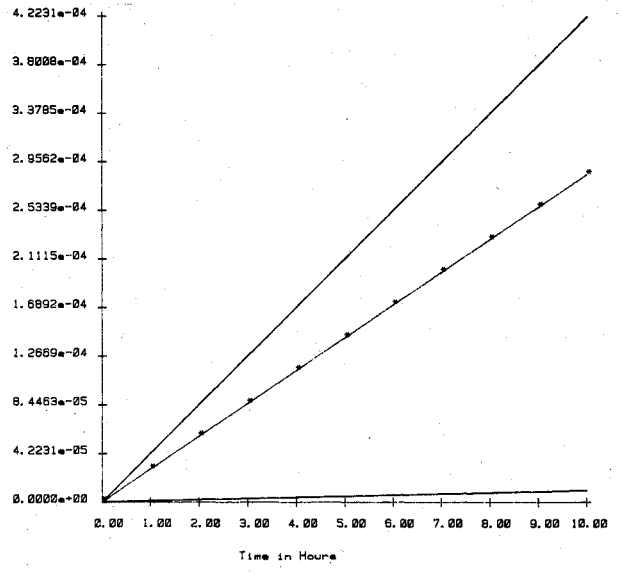

Fig. 13 Unreliability bands for repairable flight control system.

Table 6 Failure rates (per hour) for flight control system

Parameter	Description	Value ±10%
λ_i	Inertial reference module	1.5×10^{-5}
λ_p	Pitch rate sensor module	1.9×10^{-5}
λ_c	Computer module	4.8×10^{-4}
λ_s	Secondary actuator module	3.7×10^{-5}
λ_b	Computer bus	2.7×10^{-6}

Table 7 Explanation of states in Fig. 12

State no.	Notes	Spare sensors	Spare multiplexors	Failed sensors	Failed multiplexors	In repair
1	Start	2	1	0	0	0
2		1	1	1	0	Sensor
3		0	1	2	0	Sensor
4		2	0	0	1	Mux
5		1	0	1	1	Mux
6		0	0	2	1	Mux
7	S1 down	0	0	3	1	Sensor1
8	S2 down	0	0	3	1	Sensor2
9	S3 down	0	0	3	1	Sensor3
10	M1 down	0	0	2	2	Mux1
11	M2 down	0	0	2	2	Mux2
12	M3 down	0	0	2	2	Mux3
13	S1&M1 down	0	0	3	2	Mux1
14	S2&M2 down	0	0	3	2	Mux2
15	S3&M3 down	0	0	3	2	Mux3
16	S1 down	0	1	3	0	Sensor1
17	S2 down	0	1	3	0	Sensor2
18	S3 down	0	1	3	0	Sensor3
19	M1 down	1	0	1	1	Mux1
20	M2 down	1	0	1	1	Mux2
21	M3 down	1	0	1	1	Mux3
22	M1 down	2	0	0	1	Mux1
23	M2 down	2	0	0	1	Mux2
24	M3 down	2	0	0	1	Mux3

Table 8 Fault/error-handling parameters for multiplexor faults in repairable flight control system (ARIES transient recovery model)

Probability that fault is transient: 0.9000
Mean duration of transient fault: 0.1000×10^{-2}
Probability that fault is catastrophic: 0.1000×10^{-4}
Number of transient recovery phases: 2
  Phase: 1 Duration: 0.2000×10^{-2}, effectiveness: 0.
  Phase: 2 Duration: 0.1000×10^{-2}, effectiveness: 0.5000
Coverage of permanent fault: 0.8500

Table 9 Fault/error-handling model for sensors in repairable flight control system (distributions and probabilities)

Transient restoration exit
  Exit probability: 0
Reconfiguration coverage exit
  Exit probability: 1.000
  Distribution type: HYPER
  Number of stages: 2
  Probability: 0.4000; rate: 100.0
  Probability: 0.6000; rate: 200.0
Single-point failure exit
  Exit probability: 0

Table 10 Comparison of failure probabilities (at 10 h) for repairable flight control system

Type of failure	Probability
Exhaustion of sensors (F1)	0.3761807×10^{-9}
Exhaustion of multiplexors (F2)	$0.1092289 \times 10^{-10}$
Failure of voter (F3)	0.9998575×10^{-5}
Single-point (coverage) failure (FSPF)	0.2748934×10^{-3}
Near-coincident fault (FNCF)	0.1925304×10^{-9}

This system was solved with the following values: the failure rate of the sensor was assumed to be $10^{-3} \pm 5\%$, that of the multiplexor was $10^{-4} \pm 5\%$, and for the voter 10^{-6} (all in the time unit of hours^{-1}). The average time to repair a sensor was 1 h ($\pm 25\%$) and to repair a multiplexor was 2 h ($\pm 25\%$). The reliability plots for this model are shown in Fig. 13, while Table 10 shows a comparison of the relative failure probabilities for the exhaustion of the components, single-point failure (coverage failure), and near-coincident faults for a 10 h mission. This model took approximately 20 min to describe interactively to HARP using the textual interface. It took 24.5 CPU s to solve this model on a VAX 11/750 running under Unix.

VII. Conclusions

We have presented an overview of HARP, the hybrid automated reliability predictor. The HARP approach to reliability modeling is characterized by a behavioral decomposition of the overall model into separate FORM (fault occurrence and repair) and FEHM (fault/error-handling model) submodels. The FORM model can be specified as a Markov chain or as a fault tree and the FEHM model in many different ways. These two models are specified and solved separately and the solutions are then combined automatically into an overall model that is solved numerically. The modeler is not restricted to constant failure rates, as HARP allows time-to-failure distributions to be either exponential or Weibull.

Behavioral decomposition provides an attractive approach to solving the large state-space problem, but we must remember that the solution obtained will, in general, be an approximation to the desired solution. It is worth noting that the behavioral decomposition approximation, as implemented in HARP, is indeed a conservative one (when restricted to constant failure rates),[22] an important consideration when predicting reliability.

The input parameters to HARP are all specified symbolically and are bound to their numerical values only at run time. The description of the models can be done graphically (on a Vectrix or an IBM PC AT, for example) or textually. Automatic parametric sensitivity analysis is performed and the solution of the model produces an error band about the predicted reliability. Although the examples presented in this paper are not large, HARP has been used to solve models with over 25,000 states. In addition to solving closed (nonrepairable) systems, HARP can be used to solve repairable systems (with constant failure rates) as well.

HARP consists of approximately 27,000 lines of standard FORTRAN 77 code including comments (we estimate that comment lines outnumber executable code by at least four to one) and is accompanied by three manuals: a mathematical description, a programmer's maintenance manual, and an introduction and guide for users. HARP is currently undergoing beta testing and is scheduled for release by NASA in 1986.

Appendix: The ESPN Fault/Error-Handling Model

A *Petri net* is an abstract, formal graph model useful for representing systems that exhibit concurrent, asynchronous, or nondeterministic behavior. A Petri net (PN) is a bipartite graph[33]: a set of *places* P (drawn as circles), a set of *transitions* T (drawn as bars), and a set of directed *arcs* A, which connect transitions to places or places to transitions. Places may contain *tokens* (drawn as dots). The state of a PN, called the *PN marking*, is defined by the number of tokens contained in each place.

A place is an input to a transition if an arc exists from the place to the transition; a place is an output from a transition if an arc exists from the transition to the place. A transition is *enabled* when each of its input places contains at least one token. Enabled transitions can fire by removing one token from each input place and placing one token in each output place. Thus, the firing of a transition causes a change of state (produces a different marking) for the PN. An *inhibitor arc* from a place to a transition has a small circle rather than an arrowhead at the transition.[33] The firing rule is changed as follows. A transition is enabled when tokens are present in all of its (normal) input places and no tokens are present in the inhibiting input places. When the transition fires, the tokens are removed from the normal input places and deposited in the output places as usual, but the number of tokens in the inhibiting input place remains zero.

A stochastic Petri net (SPN)[34] is obtained by associating with each transition a so-called firing time. Once a transition is enabled, an exponentially distributed amount of time elapses. If the transition is still enabled, it will then fire. A generalized stochastic Petri net (GSPN)[35] allows immediate (zero firing time) as well as timed transitions (exponentially distributed firing times); immediate transitions are drawn as thin bars, timed transitions as thick bars.

An extended stochastic Petri net[36] allows firing times to belong to an arbitrary distribution. In addition to the general firing time distributions, two other extensions to Petri nets are considered here. A *probabilistic arc* from a transition to a set of output places deposits a token in one (and only one) of the places in the set. The choice of which place receives the token is determined by the probability labels on each branch of the arc. In Fig. 3, when transition T1 is enabled, it fires by removing the token from the input place and depositing it in either place "active intermittent" (with probability $1 - t$) or in place "transient" (with probability t).

A *counter arc* from a place to a transition is labeled with an integer value k. This changes the firing rule such that a transition is enabled when tokens are present in all of its (normal) input places and at least k tokens are present in the counter input place. When the transition fires, one token is removed from each normal input place, while all k tokens are removed from the counter input place. Associated with a particular counter arc can be a *counter-alternate arc*, which enables an alternate transition when the count is between 1 and $(k - 1)$, inclusive. The alternate transition can fire once each time a token is deposited in the counting input place until there are k tokens present. The count remains unchanged by the firing of the alternate transition, as it removes no token from the counter input place. A counter-alternate arc is labeled with a $\bar{k}$. Neither the counter arc nor the counter-alternate arc are true extensions to Petri nets, as both can be realized by a cascade of normal places and transitions.[36] Rather they are useful shorthand notations for such a cascade.

The ESPN model was developed as an aid in modeling coverage in fault-tolerant computer systems. Many issues must be considered in the design of a general fault/error-handling model. Among these are the different classes of faults, the available recovery mechanisms, and the various possibilities for reconfiguration. The inherent concurrency between the fault activity and the system fault treatment mechanism can be captured most effectively in terms of an ESPN. As an example, consider the HARP fault-handling model shown in Fig. 3. When a fault occurs in the system, a token is deposited in the place labeled "fault," enabling transition T1, which fires immediately. A token is then deposited in place "active intermittent" or "transient" with probability $1 - t$ and t, respec-

tively, depending upon whether the fault is intermittent or transient. (t is a user-input value defining the percentage of transient faults.) Simultaneously, a token is deposited in place "fault exists," which serves as a logical marker representing the presence of a fault. If the fault is intermittent, the token deposited in place "active intermittent" will circulate between places "active intermittent" and "benign intermittent," signifying the oscillation of the fault between the active and benign states. If the fault is transient, eventually the token deposited in place "transient" will be passed to place "transient gone," signifying the disappearance of the fault. Note that if a token exists in both places "transient gone" and "fault exists," transition T5 can fire. This represents a transient fault that disappears before its presence is felt.

While the fault is present and is still active (i.e., a token in place "fault exists" and *no* token in either places "benign intermittent" or "transient gone"), two things may happen: an error may be produced or the fault may be detected directly. These two events are represented by transitions T6 and T7, respectively. If the self-test procedure is run while the fault is active, it will be detected with probability d (d is a user-input value defining the detectability of stuck-at faults). Once an error is produced, it is detected with probability q or it propagates through the system, causing a system failure.

Once the fault is detected, a token is deposited in place "counter" that records the number of times transient recovery is attempted. As long as there are fewer than K tokens in place "counter," transient recovery can begin. When recovery is completed, the fault may still exist and the detection/recovery cycle may repeat. If recovery has completed and the transient fault is gone, T5 is enabled and the system is once again functioning correctly. If the recovery has completed and the intermittent fault has gone benign, transitions T6 and T7 wait for the fault to become active again before they are enabled.

If the fault is detected too often (more than K times), the fault is then assumed to be permanent in nature and no automatic recovery process is begun. This is modeled by the accumulation of K tokens in place "counter." Once K tokens are present, transition T11 is disabled (transient recovery procedures are inhibited) and transition T12 is enabled (permanent recovery procedures begin). Once the fault is determined to be permanent, a diagnosis procedure is invoked to isolate the faulty unit; this is represented by a token in place "isolate." The diagnosis procedure is successful with probability i. If the faulted unit is isolated, the system attempts automatic reconfiguration, which is represented by place "reconfigure." Reconfiguration is successful with probability r and the token is passed to place "permanent coverage," which represents that the system may again be operating correctly, although the performance may be somewhat degraded.

The user of this model must define the distributions for each timed transition, the probability of fault detection d, error detection q, isolation i, and reconfiguration r. The user must also provide the number of attempts at transient recovery K and the percentage of transient faults t.

In the ESPN fault-handling model, exit R (transient restoration) represents depositing a token in the "transient restoration" place, exit C depositing a token in the "permanent coverage" place, exit S depositing a token in the "single-point failure" place, and entry I the initial marking of the net. (The initial marking of the net consists of a token in the place labeled "fault.")

Acknowledgments

We thank the NASA Langley Research Center for their continued support of this project under Grant NAG1-70, and particularly Salvatore Bavuso, project engineer. From the Duke University Center for Computer Systems Analysis, we thank Beth Rothmann for her work on the graphic and textual interfaces; Mark Boyd for his work on fault trees; and Philip Thambidurai and Gianfranco Ciardo for their work on testing HARP. We would also like to thank the anonymous referees for their careful reading of the text and their very helpful suggestions. We would particularly like to credit the referees for the phrase "good as old" used in Sec. V. This phrase clearly illustrates the problem inherent in combining repair and time-dependent failure rates in the same model.

References

[1]Trivedi, K.S., *Probability and Statistics with Reliability, Queuing and Computer Science Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1982.

[2]Trivedi, K.S., Gault, J.W., and Clary, J.B., "A Validation Prototype of System Reliability in Life-Critical Applications," *Proceedings Pathways to System Integrity Symposium*, National Bureau of Standards, Gaithersburg, MD, 1980.

[3]Bouricius, W.G., Carter, W.C., and Schneider, P.R., "Reliability Modeling Techniques for Self-Repairing Computer Systems," *Proceedings 24th Annual ACM National Conference*, 1969, pp. 295-309.

[4]Mathur, F.P., "Automation of Reliability Evaluation Procedures through CARE—The Computer-Aided Reliability Estimation Program," *Proceedings AFIPS Fall Joint Computer Conference*, Vol. 41, 1972, pp. 65-82.

[5]Carter, W.C., Jessep, D.C., Bouricius, W.G., Wadia, A.B., McCarthy, C.E., and Milligan, F.G., "Design Techniques for Modular Architecture for Reliable Computer Systems," IBM T.J. Watson Research Center, Rept. RA-12, March 1970.

[6]Rennels, D.A. and Avizienis, A., "RMS: A Reliability Modeling System for Self-Repairing Computers," *Proceedings IEEE 3rd Fault-Tolerant Computing Symposium*, IEEE, New York, June 1973, pp. 131-135.

[7]Geist, R. and Trivedi, K., "Ultra-High Reliability Prediction for Fault-Tolerant Computer Systems," *IEEE Transactions on Computers*, IEEE, New York, Dec. 1983, pp. 1118-1127.

[8]Tripathi, S.K., "On Approximate Solution Techniques for Queueing Network Models of Computer Systems," Tech. Report CSRG-106, University of Toronto, Canada, Tech. Rept. CSRG-106, 1979.

[9]Trivedi, K. and Geist, R., "Decomposition in Reliability Analysis of Fault-Tolerant Systems," *IEEE Transactions on Reliability*, IEEE, New York, Dec. 1983, pp. 463-468.

[10]Siewiorek, D.P. and Swarz, R.S., *The Theory and Practice of Reliable System Design*, Digital Press, 1982.

[11]Barlow, R.E. and Lambert, H.E., "Introduction to Fault Tree Analysis," *Proceedings of Society for Industrial and Applied Mathematics*, edited by J.B. Fussell and N.D. Singpurwalla, 1975, pp. 7-35.

[12]Avizienis, A., "Fault-Tolerance: The Survival Attribute of Digital Systems," *Proceedings of the IEEE*, IEEE, New York, Oct. 1978, pp. 1109-1125.

[13]Conn, R.B., Merryman, P.M., and Whitelaw, K.L., "CAST—A Complementary Analytic-Simulative Technique for Modeling Fault-Tolerant Computing Systems," AIAA Paper, Nov. 1977.

[14]McGough, J., "Effects of Near-Coincident Faults in Multiprocessor Systems," *Proceedings 5th IEEE/AIAA Digital Avionics Systems Conference*, IEEE, New York, Nov. 1983.

[15]Dugan, J.B., Trivedi, K.S., Geist, R., and Nicola, V.F., "Extended Stochastic Petri Nets: Applications and Analysis," *Performance 84*, E. Gelenbe, ed., North-Holland, 1984.

[16]Geist, R., Trivedi, K., Dugan, J.B., and Smotherman, M., "Design of the Hybrid Automated Reliability Predictor," *Proceedings IEEE/AIAA 5th Digital Avionics Systems Conference*, IEEE, New York, Nov. 1983.

[17]Trivedi, K., Geist, R., Smotherman, M., and Dugan, J.B., "Hybrid Modeling of Fault-Tolerant Systems," *Computers and Electrical Engineering, An International Journal*, Vol. 11, Nos. 2, 3, 1985, pp. 87-108.

[18]Geist, R., Trivedi, K., Dugan, J.B., and Smotherman, M., "Modeling Imperfect Coverage in Fault-Tolerant Systems," *Proceedings IEEE 14th Fault-Tolerant Computing Symposium*, IEEE, New York, June 1984.

[19]Stiffler, J.J. and Bryant, L.A., "CARE III Phase III Report—Mathematical Description," NASA CR 3566, Nov. 1982.

[20]Ng, Y.-W. and Avizienis, A., "A Model for Transient and Permanent Fault Recovery in Closed Fault-Tolerant Systems," *Proceedings IEEE 6th Fault-Tolerant Computing Symposium*, IEEE, New York, June 1977, pp. 182-187.

[21] Makam, S.V. and Avizienis, A., "ARIES 81: A reliability and life-cycle evaluation tool for fault-tolerant systems," *Proceedings IEEE 12th Fault-Tolerant Computing Symposium*, IEEE, New York, June 1982, pp. 267-274.

[22] McGough, J., Smotherman, M., and Trivedi, K., "The Conservativeness of Reliability Estimates Based on Instantaneous Coverage," *IEEE Transactions on Computers*, Vol. C-34, No. 7, July 1985, pp. 602-609.

[23] Hopkins, A.L. Jr., Smith, T.B. III, and Lala, J.H., "FTMP—A Highly Reliable Fault-Tolerant Multiprocessor for Aircraft," *Proceedings of the IEEE*, Vol. 66, Oct. 1978, pp. 1221-1239.

[24] Wensley, J.H. et al. "SIFT: The Design and Analysis of a Fault-Tolerant Computer for Aircraft Control," *Proceedings of the IEEE*, Vol. 66, Oct. 1978, pp. 1240-1255.

[25] Stiffler, J.J., "Computer Aided Reliability Estimation," AIAA Paper, Nov. 1977.

[26] Trivedi, K. and Geist, R., "A Tutorial on the CARE III Approach to Reliability Modeling," NASA CR 3488, 1981.

[27] Trivedi, K., "Reliability Evaluation for Fault-Tolerant Systems," *Mathematical Computer Performance and Reliability*, edited by G. Iazeolla, P.J. Courtois, and A. Hordijk, North-Holland Publishing Co., Amsterdam, 1984, pp. 403-414.

[28] Geist, R.M., Smotherman, M.K., Trivedi, K.S., and Dugan, J.B., "Reliability of Life-Critical Systems," accepted subject to revision, *Acta Informatica*.

[29] Shampine L.F. and Watts, H.A., "Global error estimation for ordinary differential equations," *ACM Transactions on Mathematics and Software*, June 1976, pp. 172-186.

[30] Smotherman, M., "Parametric Error Analysis and Coverage Approximations in Reliability Modeling," Ph.D. Dissertation, Dept. of Computer Science, University of North Carolina, Chapel Hill, 1984.

[31] Smotherman, M., Geist, R., and Trivedi, K., "Provably Conservative Approximations to Complex Reliability Models," to be published in *IEEE Transactions on Computers*, March 1986.

[32] Bavuso, S.J., Petersen, P.L., and Rose, D.M., "CARE III Model Overview and User's Guide," NASA TM 85810, June 1984.

[33] Peterson, J.L., *Petri Net Theory and the Modeling of Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1981.

[34] Molloy, M.K., "Performance Analysis Using Stochastic Petri Nets," *IEEE Transactions on Computers*, Sept. 1982.

[35] Marsan, M.A., Balbo, G., and Conte, G., "A Class of Generalized Stochastic Petri Nets for the Performance Evaluation of Multiprocessor Systems," *ACM Transactions on Computer Systems*, May 1984.

[36] Dugan, J.B., "Extended Stochastic Petri Nets: Applications and Analysis," Ph.D. Dissertation, Dept. of Electrical Engineering, Duke University, Durham, NC, 1984.