An Efficient Algorithm To Analyze New Imperfect Fault Coverage Models

Suprasad V. Amari, PhD, Relex Software Corporation
Albert Myers, Northrop Grumman Corporation
Antoine Rauzy, PhD, Institut de Mathématiques de Luminy

Key Words: fault tolerant systems, imperfect fault coverage, combinatorial models, system reliability

SUMMARY & CONCLUSIONS

Fault tolerance has been an essential architectural attribute for achieving high reliability in many critical applications of digital systems. Automatic recovery and reconfiguration mechanisms play a crucial role in implementing fault tolerance because an uncovered fault may lead to a system or subsystem failure even when adequate redundancy exists. In addition, an excessive level of redundancy may even reduce the system reliability. Therefore, an accurate analysis must account for not only the system structure but also the system fault and error handling behavior. The models that capture the fault and error handling behavior are called coverage models. The appropriate coverage modeling approach depends on the type of fault tolerant techniques used.

Recent research emphasizes the importance of two new categories of coverage models: Fault Level Coverage (FLC) models and One-on-one Level Coverage (OLC) models. However, the methods for solving FLC and OLC models are much more limited, primarily because of the complex nature of the dependency introduced by the reconfiguration mechanisms. In this paper, we propose an efficient algorithm for solving FLC and OLC models.

1 INTRODUCTION

A system is called a fault tolerant system if it can tolerate some faults and functions successfully even in the presence of these faults [1]. In many critical applications of digital systems, fault tolerance has been an essential architectural attribute for achieving high reliability [2]. Fault tolerant designs are particularly important for computer and communication systems that are used in life-critical applications such as flight control, space missions, and data storage systems [3]. Fault tolerance is generally achieved by using redundancy concepts that utilize such techniques as error correcting codes (ECC), built-in tests (BIT), replication, and fault masking [1, 4]. Automatic recovery and reconfiguration mechanisms (detection, location, and isolation) play a crucial role in implementing fault tolerance because an uncovered fault may lead to a system or subsystem failure even when adequate redundancy exists [5]. Hence, systems subjected to imperfect fault coverage may fail prior to the exhaustion of redundancy due to uncovered component failures [2]. In addition, an excessive level of redundancy may reduce the system reliability [6]. Therefore, an accurate reliability analysis of these systems is important. An accurate analysis must consider the fault and error handling behavior [7] in addition to the system structure and its provision of redundancy [8].

1.1 *Acronyms & Abbreviations*

BIT	Built-In Test
CCF	Common Cause Failure
CF	Coverage Factor
ECC	Error Correcting Codes
ELC	Element Level Coverage (model)
FEHM	Fault/Error Handling Model
FLC	Fault Level Coverage (model)
HARP	Hybrid Automated Reliability Predictor
IPC	Imperfect Coverage
OLC	One-on-one Level Coverage (model)
PFC	Perfect Fault Coverage (model)
RBD	Reliability Block Diagram
SDP	Sum of Disjoint Products
XHARP	Extended HARP

1.2 *Terminology and Definitions*

In this paper, we borrow some terminology from [9, 10].

- *System:* A collection of components arranged according to a specific design in such a manner to perform a specific desired function or functions.
- *Subsystem:* A subdivision of the system that performs a specific function or functions that are needed for the overall system functionality.
- *Component:* A self-contained element of an entire system or its subsystem that performs a function necessary to the operation of the system or the subsystem.
- *Failure:* A deviation from the required functionality. A system failure occurs as a result of its component or subsystem failures.

- *Error:* A part of the system state which is liable to lead to failure.
- *Fault:* The cause of an error, i.e., a short circuit, electromagnetic perturbation, etc. Upon occurrence, a fault creates a latent error, which becomes effective when it is activated. When the error affects the delivered service or functionality, a failure occurs.
- *Fault/Error Handling Model (FEHM):* A model that describes the behavior of the system in response to a fault. It is also called coverage model. If the offending fault is transient, and it can be handled without discarding the component, a transient restoration is taking place, and the component returns to its normal working state. If the fault is determined to be permanent, and the offending component is discarded, a permanent coverage is taking place, and the component is considered to be in the covered failure mode (safe failure mode). If the recovery mechanism is unable to detect, locate, or isolate the fault, the fault may lead to an uncovered failure. If the fault by itself causes the system to fail, the single-point failure is taking place. However, depending on the type of recovery mechanisms used, some systems can tolerate multiple undetected or non-isolated faults. If the system can tolerate only one non-isolated fault at a time, the occurrence of a second fault that ~~interfaces~~ interferes with the recovery process of the first fault can cause an uncovered failure, which is called a near-coincident failure.
- *Covered Failure:* A failure that is mitigated by the successful recovery of faults. A component fails in the covered mode if the fault is determined to be permanent and the component is removed from the system. A system fails in the covered mode due to exhaustion of redundancy as a result of the covered failures of its components.
- *Uncovered Failure:* A failure that results from the unsuccessful recovery of faults. A successful recovery of a fault includes both the transient restoration and permanent removal of the faulty component. An unsuccessful recovery of a fault is the situation that leads to the system or the subsystem failure in the uncovered mode. This happens either due to single or multiple non-isolated faults that defeat the redundancy. If the effects of the uncovered failures are local to a subsystem, then the subsystem can be analyzed independently and it can be replaced by a logically equivalent component. Therefore, only the uncovered failures that lead to the system failure need to be considered. Hence, without loss of generality, we can assume that the system fails in the uncovered mode if at least one of its components fails in the uncovered mode.
- *Coverage:* A factor used to account for the efficiency of fault tolerance mechanisms. More specifically, coverage is the [conditional] probability of successfully covering a fault, i.e., avoiding fault propagation, given that the fault has occurred. It is also called coverage factor or coverage probability. $\boldsymbol{coverage \equiv CF = \Pr\{system\ recovery \mid fault\ occurs\}}$ If the coverage value is less than 1, then it is called imperfect coverage (IPC).
- *Single-Point Failure:* A system failure caused by a single fault without interference of other faults.
- *Near-Coincident Failure:* A system failure resulting from the occurrence of two coexisting (not simultaneously occurring) faults. The near-coincident failure condition occurs when the system has already experienced one fault and is in the process of recovering from it when a second statistically independent fault occurs in another unit that is critically coupled to the unit experiencing the first fault.
- *Single-Fault Model:* The effectiveness of recovery mechanisms depend on the occurrence of individual faults. The system failures that are caused by uncovered single-faults are known as single-point failures.
- *Multi-Fault Model:* The effectiveness of recovery mechanisms depend on the occurrence of multiple faults.
- *Combinatorial Model:* A model that represents the system state (success or failure) as a combination of the states of its components (success or failure). Examples include RBDs, fault trees, diagraphs, and reliability networks.

1.3 *Coverage Models*

The models that consider the effects of imperfect fault coverage are known as imperfect fault coverage models or simply fault coverage models or coverage models [2]. Depending on the type of fault tolerant techniques used, the models are classified as [4]:

- Perfect Fault Coverage (PFC). The coverage factor is 1. Hence, the system can be analyzed using classical reliability analysis techniques, which do not consider the effects of coverage.
- Element Level Coverage (ELC). A particular coverage value is associated with each component. This value is independent of the status of other components.
- Fault Level Coverage (FLC). The coverage value depends on the number of good components that belong to a specific group (i.e., the status of other components).
- One-on-one Level Coverage (OLC). OLC is a special case of FLC where the coverage factor is 1 when the number of good components in a specific group is greater than 2.

The ELC model is appropriate when the selection among the redundant elements is made on the basis of a self-diagnostic capability of the individual elements. Such systems typically contain a built-in test (BIT) capability. The FLC

model is appropriate for modeling systems in which the selection among redundant elements varies between initial and subsequent failures. An example is a majority voting system among the currently known working components. In the HARP terminology, ELC models are known as single-fault models, whereas the OLC and FLC models are known as multi-fault models.

Perfect fault coverage models, which are extensively studied in the literature, are actually special cases of the other three model types. Because OLC is a special case of FLC, there are effectively two classes of models (ELC and FLC) that need special attention. In the fault tolerant literature, ELC models have been studied for a long time. The landmark developments in solving these models include decomposition techniques (1983) [11], Markov chain-based solutions (1985-1995) [9, 12], multi-state combinatorial techniques (1995) [13], and a separable method (1999) called the Simple and Efficient Algorithm (SEA) [8], which uses conditional probabilities. As its name suggests, SEA provides the most simple and efficient method available for solving ELC models.

Methods for solving FLC models are much more limited, primarily because of the complex nature of the dependency introduced by reconfiguration mechanisms. The only ~~existing~~ published methods are Markov chain-based solutions [12] and sum of disjointing products methods (2006) [4]. Both of these methods require ~~huge~~ significant computer space and time for systems with large n. In this paper, utilizing the concepts in [2, 4], we propose an efficient method for solving both OLC and FLC models by introducing a new implicit Common Cause Failure (CCF) model and a new extension of the separable method used in SEA.

2 SYSTEM DESCRIPTION AND ASSUMPTIONS

The system description and assumptions are:

- The system consists of several components.
- The system is subjected to imperfect fault coverage. The uncovered failure of any component causes immediate system failure, even in the presence of adequate redundancy.
- The fault coverage = Pr{system recovers | fault occurs} depends on the faulty component and system state. Specifically, the system consists of several groups of components, and the fault coverage of a component depends on the number of working components in its group (called an FLC group).
- Component failures are s-independent. The only dependency among the component failures is due to the uncovered failures caused by imperfect fault coverage mechanisms.
- An s-coherent combinatorial model (RBD, fault tree, digraph, or reliability network) can be used to represent the combinations of covered component failures (or successes) that lead to system failure (or success).
- Fault occurrence probabilities are given either (a) as fixed probabilities (for a given mission time), or (b) in terms of a lifetime distribution. They are s-independent of the system state.

Therefore, the inputs for the reliability analysis are:

- A combinatorial model describing the combinations of covered failures (or successes), which lead to system failure (or success).
- A set of parameters describing the component time to failure behavior.
- A set of FLC groups. Each component belongs to only one group. For each group, the set of FC probabilities depend on the number of working components. If a set contains n components, it needs n FC probabilities. In [4], the coverage probability when all components have failed is considered to be zero. Although this assumption is valid for most systems, our proposed method removes this restriction while still providing the ability to support this case.

3 AN EXISTING COMBINATORIAL METHOD

3.1 *k-out-of-n System with Identical Components*

In this section, we describe the method provided in [4] for analyzing k-out-of-n systems subjected to FLC models. The method has the following steps:

- Compute the probability of exactly m-out-of-n functioning components.
- Multiply this probability by an appropriate coverage factor.
- Sum these updated probabilities over the values of m, ranging from k and n, to find the system reliability.

Consider a 2-out-of-4 system with identical components, where all components belong to a single FLC group. Assume that p is the reliability of each component and c_i is the coverage probability at the i^{th} failure. Because the system fails after three failures (irrespective of fault coverage), both c_3 and c_4 are not applicable and can be considered as $c_3=c_4=0$. The values of c_3 and c_4 are required only if we want to distinguish the failures as covered or uncovered. Such a distinction is required when analyzing systems consisting of several k-out-of-n subsystems that are subjected to imperfect fault coverage.

Under the perfect coverage, the reliability of a 2-out-of-4 system is:

$$R = \sum_{i=0}^{n-k} P_i = P_0 + P_1 + P_2 \tag{1}$$

where P_i is the probability of exactly i components failed. For an identical component case, P_i can be calculated using the binomial distribution.

$$P_i = \binom{n}{i} q^i p^{n-i} \tag{2}$$

where $q=1-p$ = unreliability of each component. Now consider the imperfect fault coverage case. Here, we should multiply P_i with the coverage probabilities associated with the 1^{st}, 2^{nd}, ...,

i^{th} failures. Define:

$$r_i = \prod_{j=0}^{i} c_j \tag{3}$$

Note: By definition, we have: $c_0 = r_0 = 1$. Therefore, system reliability is:

$$R = \sum_{i=0}^{n-k} r_i P_i = P_0 + r_1 P_1 + r_2 P_2 \tag{4}$$

This is exactly the procedure used in [2, 4]. The coverage probabilities c_i can be calculated using an appropriate coverage model. Although the method in [4] can be used for generic multi-fault coverage models, the coverage probabilities in [4] are calculated using the near-coincident fault model, where an uncovered failure occurs if the system experiences a second fault during the recovery from a first fault. For example, let τ be the recovery time for a fault. During the first failure, any one of the remaining *(n-1)* components can fail during τ and can cause the system failure. Extending the same logic for other cases, we have:

$$c_i = \exp[-(n-i)\lambda\tau] \tag{5}$$

where λ is the failure rate of each component. It should be noted that [4] uses a different approach, which is based on BIT coverage, for calculating c_{n-1} and c_n=0.

3.2 *k-out-of-n System with Non-Identical Components*

Extending the k-out-of-n System method for a non-identical component case is simple. The only difference is that P_i should be computed using an appropriate formula. For simplicity, consider a 2-out-of-3 system. We have:

$$R = P_0 + r_1 P_1 + r_2 P_2 \tag{6}$$

The probabilities P_i*'s* can be calculated in several ways. The algorithms with complexity *O(kn)* or better are available in the literature and are discussed in [14]. However, if we use the classical truth-table approach, we have:

$$P_0 = p_1 p_2 p_3 \tag{7}$$

$$P_1 = q_1 p_2 p_3 + p_1 q_2 p_3 + p_1 p_2 q_3$$

$$P_2 = p_1 q_2 q_3 + q_1 p_2 q_3 + q_1 q_2 p_3$$

The calculations can be generalized for the case where c_i is dependent on a set of components that already failed instead of on just the number that have failed. The details of these calculations are discussed in [14].

3.3 *General System Configurations*

The reliability of a general system is computed in a similar fashion, where each term in the system reliability expression represents exactly a specific number of working components from each of the k-out-of-n group. The probability of each term is multiplied with an appropriate coverage factor to find the overall system reliability. The main contribution of Myers' approach [4] is that as long as the system failure logic is represented using a combinatorial model, the inclusion of uncovered failures, resulting from either single-fault or multi-fault coverage models, does not require a complex Markov chain-based solution. Hence, we can solve both single-point failures and near-coincident failures using combinatorial models, which is a major break through in the analysis of coverage models.

The disadvantage of this method is its computational complexity. To produce correct results, the system reliability expression should be expressed in a sum of disjoint products (SDP) form that is grouped according to a specific combination of "number of good units from each FLC group". Due to this restriction, it cannot be combined with efficient algorithms available for combinatorial reliability analysis. In this paper, combining the concepts of Myers' approach, SEA and CCF, we propose an efficient algorithm to solve FLC models and its special case OLC models. In addition to this, we also propose an approximate algorithm that produces the results quickly.

4 *PROPOSED SOLUTION*

In this section, we provide two efficient algorithms for computing the reliability of general system configurations subjected to FLC and OLC models.

- The first algorithm is based on an implicit CCF model, and it produces exact results.
- The second algorithm converts the problem into an equivalent perfect coverage model, and it produces an approximate result with a very small deviation from the exact results.

4.1 *Algorithm 1: Exact Solution*

This algorithm uses SEA-based calculations and CCF analysis concepts. The basic idea of this algorithm is that the conditional reliability of the system can be computed using implicit CCF analysis given that there are no uncovered failures in the system. To apply implicit CCF analysis, we should know the joint probabilities of events that belong to an *s*-dependent group (FLC group). The procedure is explained through a simple example of 2-out-of-3 system with non-identical components subjected to imperfect fault coverage. To apply this method, first we should compute the uncovered failure probability (*U*) of each FLC group *j* as in equation (8).

$$U_j = \sum_{i=1}^{n(j)} (1-c_i) r_{i-1}(j) P_i(j) \tag{8}$$

Because this example contains only one FLC group, for simplifying notation, we omit the subscript for the FLC group. For this example, we have:

$$U = (1-c_1)P_1 + (1-c_2)r_1 P_2 + (1-c_3)r_2 P_3 \tag{9}$$

Let x_i be the probability that only the i^{th} component in the FLC group has failed and that failure is covered. Similarly, x_{ij} represents the probability that only the i^{th} and j^{th} components have failed. Hence, we have:

$$x_1 = r_1 q_1 p_2 p_3 / (1-U) \tag{10}$$

$$x_{12} = r_2 q_1 q_2 p_3 / (1-U)$$

$$x_{123} = r_3 q_1 q_2 q_3 / (1-U)$$

Similarly, we can compute (1) x_2 and x_3, and (2) x_{13} and x_{23}. Let y_{ij} be the probability that at least components i and j have failed in the covered mode, and assume that there are no uncovered failures in the system. Hence, we have:

$$y_1 = x_1 + x_{12} + x_{13} + x_{123} \quad (11)$$
$$y_{12} = x_{12} + x_{123}$$
$$y_{123} = x_{123}$$

Similarly, we can compute (1) y_2 and y_3, and (2) y_{13} and y_{23}. Once we know these x and y values, it is straightforward to compute the system reliability. Most algorithms use the y values. For example, the conditional unreliability of 2-out-of-3 system is:

$$Q^c = y_{12} + y_{13} + y_{23} - y_{123} \quad (12)$$

Finally, the overall system reliability is: $(1-U)(1-Q^c)$. These results match the results in section 3.2. However, this method is not required for the simple k-out-of-n systems. This method is developed for the most complex systems considered in [4], where the system consists of several FLC groups. The algorithm for complex systems follows:

- Using equation (8), for each FLC group j in the system, find the uncovered failure probability, (U_j).
- Using U_j, for each FLC group, find the x and y values required for the common cause analysis as shown in equations (10) and (11).
- Using these x and y values, compute the system conditional reliability, R^c.
- Compute the probability of "no uncovered failure in the system," (C).

$$C = \prod_{j=1}^{m} (1 - U_j) \quad (13)$$

- Finally, the overall system reliability is: $R = C.R^c$

It should be noted that it is straightforward to convert an implicit method into an equivalent explicit method that involves only independent components [15]. Therefore, the system reliability can be solved using any existing combinatorial reliability algorithm, including BDD [16]. The details are given in [14].

4.2 *Algorithm 2: An Efficient Approximation*

This algorithm also uses SEA-based calculations. The basic idea of this approximate algorithm is "the conditional reliability of the system, given that no uncovered failure in the system (R^c)," is almost equivalent to the "unconditional reliability of the system with perfect coverage (R^p)." In fact, $R^p \leq R^c$. Hence, this algorithm produces provably conservative results for the system reliability. The algorithm follows:

- Using equation (8), for each FLC group j in the system, find the uncovered failure probability, (U_j).
- As in equation (13), compute the probability of "no uncovered failure in the system," (C).
- Assuming perfect coverage, i.e., using unconditional component reliabilities, compute the system reliability, R^p, using any combinatorial algorithm.
- Finally, the overall system reliability is: $R = C.R^p$.

5 *EXAMPLE*

In this section, we demonstrate the exact and approximate algorithms proposed in section 4.1 and section 4.2. We consider two cases of a simple hypothetical quadruplex redundant real time control system discussed in [4]. The system consists of four sets of components.

- Four electric power sources: P1, P2, P3, and P4.
- Four power distribution buses: B1, B2, B3, and B4.
- Four feedback sensors: S1, S2, S3, and S4.
- Four control computers: C1, C2, C3, and C4.

The buses are considered to be perfect and never fail. All components within each group are considered to be identical. The coverage of power sources is perfect. The parameters of the system are shown in Table 1.

Parameter	Value
Computer frame rate	100 hz = 10 ms
Fault window	3 frames = 30 ms
Power source failure rate	500 fpmh
Sensor failure rate	250 fpmh
Computer failure rate	750 fpmh
Sensor BIT coverage	0.99 = $c_3(S)$
Computer BIT coverage	0.999 = $c_3(C)$
Mission time	1 hour

Table 1 – Parameters for the Quadruplex System

The Boolean expression that represents the system success (SS) in terms of component success is:

$$SS = (P1 + P4) \cdot S1 \cdot C1 + (P1 + P2) \cdot S2 \cdot C2 + (P2 + P3) \cdot S3 \cdot C3 + (P3 + P4) \cdot S4 \cdot C4 \quad (14)$$

The sensors and computers form two FLC groups. Hence, *FLC-S = {S1, S2, S3, S4}* and *FLC-C = {C1, C2, C3, C4}*. The coverage values c_1 and c_2 are calculated using equation (5). The c_3 is equivalent to the corresponding BIT coverage, and $c_4 = 0$. The corresponding coverage values are shown in Table 2. Refer to [4] for more details on this system.

CF	Sensor	Computer
c_1	$c_1(S)$ = 0.99999999375	$c_1(C)$ = 0.99999998125
c_2	$c_2(S)$ = 0.99999999583	$c_2(C)$ = 0.9999999875
c_3	$c_3(S)$ = 0.99	$c_3(C)$ = 0.999
c_4	$c_4(S)$ = 0.0	$c_4(C)$ = 0.0

Table 2 – Coverage Factors

We consider two cases: (1) all components are independent – the only dependency among the component failures is due to the uncovered failures caused by imperfect fault coverage, and (2) dependency among component failures in addition to the imperfect coverage dependency. The analysis presented in this paper is applicable for case 1. However, it can also be extended for the case 2.

5.1 *Case 1: Independent Component Failures*

The system reliability is computed by utilizing the fact that all components within each group are identical. Therefore, the reliability of the system is calculated considering 4 distinct cases of $i \in \{1, 2, 3, 4\}$: *e*xactly *i*-out-of-4 power sources are working. For each of this case, we computed the system reliability using Sylvester-Poincare's expansion method. The overall system reliability is calculated using the total probability theorem.

The exact answer for unreliability using the implicit CCF analysis is 6.583E-11. The unreliability of the perfect coverage system is 1.062E-12. The uncovered failure probability of the system is 6.502E-11. Therefore, the upper bound on the system unreliability is 6.608E-11. Hence, the percentage error with the unreliability approximation is 0.38, whereas the percentage error in the reliability approximation is only 2.55E-11. This small error indicates that algorithm 2 is nearly as accurate as algorithm 1 and requires much less computational time.

5.2 *Case 2: Dependent Component Failures*

The analysis presented in this paper can be integrated with the PFC analysis methods applicable for the dependent component failures to compute the reliability IPC models with dependent failures. For example, consider the following dependencies:

- Both sensors and computers in a specific channel are forced to fail when associated power is failed, i.e., failure of both P1 and P4 force S1 and C1 to fail. Similarly, failure of both P1 and P2 force S2 and C2 to fail, and so on.
- Failure of a computer in any channel forces its associated sensor, i.e., failure of C1 forces the failure of S1. Similarly, failure of C2 forces the failure of S2, and so on.
- Therefore, the coverage factors of sensors and computers are dependent on all other components.

Because the coverage factors of components are not independent, we cannot use equation (13) to compute *C*: probability of "no uncovered failure in the system". However, the equation (13) can be modified using conditional probabilities. For case 2 of this system, we first computed the probability of exactly i-out-of-4 power sources are working. Then using the condition that exactly i-power sources are working, we computed the reliability j-out-of-4 computers are working and multiplied this probability with its coverage factor. Similarly, we performed the same analysis for computing the k-out-of-4 sensors are working. For each combination of i, j, and k, these probabilities are multiplied. Finally, the probability of "no uncovered failure in the system" is calculated as a sum of these probabilities. The corresponding uncovered failure probability of the system is 1.238E-10. Therefore, the upper bound on the system unreliability is 1.248E-10. However, the exact answer using truth-table approach is 1.239E-10. The results indicate the SEA based algorithm can also be used for analyzing IPC models with dependent failures. The exact results for this case are also presented in [4]. A more detailed analysis of case 2 is presented in [14].

6 CONCLUSIONS

In this paper, we propose an efficient algorithm for solving FLC and OLC models by introducing a new implicit common cause failure model and an extension of the separable method used in the Simple and Efficient Algorithm (SEA). The proposed algorithm can solve a wide range of coverage models that include both single-point faults and near-coincident faults in an efficient way. In addition to this, we also propose an approximate algorithm that produces the results quickly. The proposed algorithms are demonstrated using an example of a quadruplex control computer system.

REFERENCES

1. M. L. Shooman, *Reliability of Computer Systems and Networks: Fault Tolerance, Analysis, and Design*, John Wiley & Sons, 2002.
2. S. V. Amari, *Reliability, Risk and Fault-Tolerance of Complex Systems*, PhD Dissertation, Indian Institute of Technology, Kharagpur, 1997.
3. Y. R. Chang, S. V. Amari, S. Y. Kuo, "OBDD-based evaluation of reliability and importance measures for multistate systems subject to imperfect fault coverage," *IEEE Trans. Dependable and Secure Computing*, vol. 2, 2005, pp 336-347.
4. A. Myers, "k-out-of-n:G system reliability with imperfect fault coverage," accepted in *IEEE Trans. Reliability.*
5. T. F. Arnold, "The concept of coverage and its effect on the reliability model of a repairable system," *IEEE Trans. Computers*, vol. C-22, (Mar.) 1973, pp 325–339.
6. S. V. Amari, J. B. Dugan, R. B. Misra, "Optimal reliability of systems subject to imperfect fault-coverage," *IEEE Trans. Reliability*, vol. 48, (Sep.) 1999, pp 275–284.
7. W. G. Bouricius, W. C. Carter, P. R. Schneider, "Reliability modeling techniques for self-repairing computer systems," *ACM Nat'l Conf.*, 1969, pp 295-309.
8. S. V. Amari, J. B. Dugan, R. B. Misra, "A separable method for incorporating imperfect fault-coverage into combinatorial models," *IEEE Trans. Reliability*, vol. 48, (Sep.) 1999, pp 267–274.
9. J. B. Dugan, K. S. Trivedi, "Coverage modeling for dependability analysis of fault-tolerant systems," *IEEE Trans. Computers*, vol. 38, (Jun.) 1989, pp 775-787.
10. A. Avizienis, J.-C. Laprie, B. Randell, C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Trans. Dependable and Secure Computing*, vol. 1, (Jan.) 2004, pp 11-33.
11. K. S. Trivedi, R. Geist, "Decomposition in reliability analysis of fault tolerant systems", *IEEE Trans. Reliability*, vol. R-32, (Dec.) 1983, pp 463-468.
12. S. J. Bavuso, et. al., *HiRel: Hybrid Automated Reliability*

Predictor (HARP) Integrated Reliability Tool System (Version 7.0), 4 vol.s, NASA TP 3452, (Nov.) 1994.
13. S. A. Doyle, J. B. Dugan, F. A. Patterson-Hine, "A combinatorial approach to modeling imperfect coverage," *IEEE Trans. Reliability*, vol. 44, (Mar.) 1995, pp 87-94.
14. S. V. Amari, A. Myers, "Imperfect fault coverage models", in *New Trends in Performability Engineering,* Editor: K. B. Misra.
15. J. K. Vaurio, "Treatment of general dependencies in system fault-tree and risk analysis," *IEEE Trans. Reliability*, vol. 51, (Sep.) 2002, pp 278-287.
16. Y. R. Chang, S. V. Amari, S. Y. Kuo, "Computing System Failure Frequencies and Reliability Importance Measures Using OBDD," *IEEE Trans. Computers*, vol. 53, 2004, pp 54-68.

BIOGRAPHIES

Suprasad V. Amari, PhD
Relex Software Corporation
540 Pellis Road
Greensburg, PA 15601 USA

e-mail: suprasad.amari@relex.com

Suprasad V. Amari is a Senior Reliability Engineer at Relex Software Corporation. He received both his MS and PhD in Reliability Engineering from the Indian Institute of Technology, Kharagpur. He has published over 35 research papers in reputed international journals and conferences. He is an editorial board member of the *International Journal of Reliability, Quality and Safety Engineering,* area editor of the *International Journal on Performability Engineering*, and management committee member of *RAMS (2005).* He is a senior member of ASQ, IEEE, and IIE; and a member of ACM, ASA, SSS, SRE, and SOLE. He is also an ASQ-certified Reliability Engineer.

Albert Myers
Corporate Vice President, Strategy and Technology
Northrop Grumman Corporation
Los Angeles, USA

e-mail: Al.Myers@ngc.com

Albert F. Myers is corporate vice president of strategy and technology for Northrop Grumman Corporation. He also served as B-2 chief project engineer, deputy program manager, and vice president of test operations. Myers earned BS and MS degrees in Mechanical Engineering from the University of Idaho. He was a Sloan Fellow at the Massachusetts Institute of Technology. In 2006, Myers was elected as a member of the National Academy of Engineering. Myers served from 1989 through 1998 on the NASA Aeronautics Advisory Board. He received the NASA Exceptional Service Medal and the 1981 Dryden Director's Award and was elected to the University of Idaho Alumni Hall of Fame in 1997.

Antoine Rauzy
Institut de Mathématiques de Luminy
163 avenue de Luminy, Case 907
13288 Marseille CEDEX 9, FRANCE

e-mail: arauzy@iml.univ-mrs.fr

Antoine Rauzy received his Ph.D. in computer sciences in 1989 and a "habilitation à diriger des recherches" in 1996. He joined the "Centre National de la Recherche Scientifique" in 1991 and the "Institut de Mathématiques de Luminy" in 2000. His topics of research are reliability engineering, formal methods, and algorithms. He has authored more than 100 articles in international conferences and journals. He designed various software products including the fault tree assessment tool Aralia. Since 2001, he has been the president of the ARBoost Technologies Company. http://iml.univ-mrs.fr/~arauzy/.