System Safety Assessment with the Markov Chain Model and Ternary Decision Diagram

Gandi Satyanarayana
Department of Computer Science&
Engineering
Dadi Institute of Engineering &Technology
Visakhapatnam, Andhrapradesh, India
satyanarayanagandi@gmail.com

P. Seetharamaiah
Department of Computer Science&
System Engineering
Andhra University
Visakhapatnam, Andhrapradesh, India
psrama@gmail.com

Abstract

In the „safety-critical' computer systems, the present existing methodologies consist of the two-failure State safety assessments that have restrictions. Based on the Markov's chain state explosion problem, Markov chain modeling technique is restricted to design only small size systems. Some imaginary assumptions are also attached to the „safety assessment' in the safety-critical computer systems. Actually to the present existing physical fault parameters, namely, physical fault failure rate and physical fault coverage, additionally design fault failure rate and design fault coverage are being attached, which is used to estimate the parameters safety and MTTUF. Design faults are included to avoid the over-optimistic estimations of safety and MTTUF. The techniques used to estimate the four input parameters are reviewed.

Keywords- Safety-critical computer systems, Faults, hazard, Errors, Failures, Physical Fault, Design Failures, Safe Failures, Unsafe Failures, Coverage, Failure rate.

I. INTRODUCTION

Safety-critical system and computer system are the two concepts concerned in a „safety-critical' computer system. A „safety-critical' system is a system whose faulty function could lead serious effects such as the huge environmental spoil, sustaining injuries, human life, or large cost-effective penalties, A computer system (IEEE Std.729,2005) is a system composed of computer(s), peripherals, and the software essential to make them work together. Safety-critical computer systems may consist of up to five components: the sensor, the application, the operator, the effector, and the computer. The pattern and the plan of safety-critical computer systems comprises of two issues 1) To state and design a „perfect' system, which work perfectly since there are no errors in it, and to prove that there are no errors in it.2) In order to achieve the exactness, to identify the errors, and to include error detection methods and recovery capabilities to prevent errors from actually causing a hazard to safety. At present computer systems are becoming more complex even after careful authentication and confirmation procedure is done we cannot give assurance for a system to work without an error.

A. Hazard, Faults, Errors and Failures

A system may not always achieve the desired aim .The factors of reliability of a system arises due to causes and effects of deviation from the system functioning. Leveson (Leveson, 2001) defines the following terms.

Definition 1.1: „Hazard' is the potential to cause harm to people, Environment, Asset and Reputation of an organization.

Definition1.2: (Johnson1989) „Fault' is a physical defect, imperfection, or flaw that occurs within some hardware or software component.

Definition1.3: „Error' is a deviation from a desired or expected state.

Definition1.4: „Failure' is the unable state of a system to perform its expected function or to reach its target in a defined time under defined environmental conditions.

A layer model is classified into Physical Universe, Information Universe and External Universe covering the concepts of Faults, Errors and Failures.

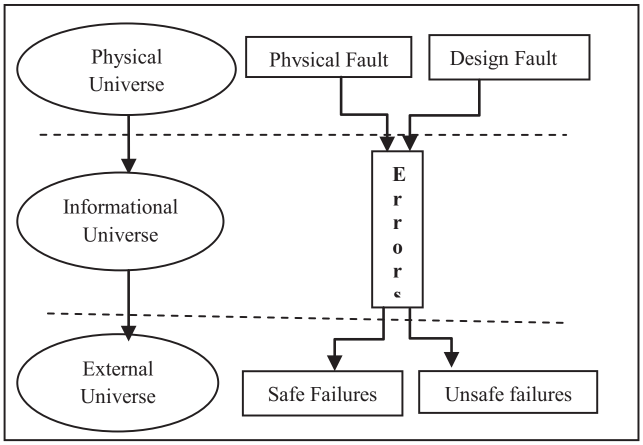

Figure 1-1   Layer model

While carrying out a service, if any erroneous data influence it, a „failure' occurs.

In this paper, we study two types of failures say safe failure and unsafe failure; and also two types of faults, namely, physical faults and design faults (Johnson, 2006).

Definition1.5:„Physical faults' occur due to contrary physical phenomena, like external conditions such as environmental perturbations, electro-magnetic perturbations, temperature, vibration etc or also like internal conditions such as physical-chemical disorders, open circuits, short circuits etc, (Johnson, 2006).

Definition1.6 „Design faults' are consigned during specified modifications or during the systems initial design or during the establishment of managing or maintenance procedures (Johnson, 2006).

B. Fault Characterizations

978-9-3805-4421-2/16/$31.00 ©2016 IEEE

To characterize the system failure behaviors, two varieties of attributes are used, namely Failure rate and coverage are defined as follows:

Definition 1.7 Failure Rate is the intended number of failures of a model of a system for a given time period.

Definition 1.8 Coverage is defined mathematically as conditional probability during the existence of a fault that the system recovers.

II. SAFETY ASSESSMENT

Parameters in engineering are used to measure the performance of a system for chosen attributes, and it is considered as an analytical model which explains the overall performance of a system related with the attributes with the functioning of parameters. In reliability engineering Reliability and MTTF (Mean Time To Failure) are the two parameters researched to measure and assess the Reliability of a system. Reliability and MTTF are not enough to design and describe the attributes of a safety-critical computer system. So in safety related parameters, Safety (Steady-State Safety) and MTTUF (Mean Time To Unsafe Failure) are used here to evaluate the system safety of safety-critical computer systems quantitatively.

Definition 2.1 Safety is the possibility of a system either to perform its functions accurately or to stop its functioning in a way that interrupt its operation of other systems or to conciliate the safety of public related with the system.

Definition 2.2 Steady-State Safety S_{ss} (Johnson 1997) is safety as time approaches infinity.

$$S_{ss} = \lim_{t \to \infty} S(t)$$

Definition 2.3: MTTUF is the probable time that a system will work before the existence of the first unsafe failure.

A. Problem Statement

Design faults must be considered to get a more realistic assessment of safety for safety-critical computer systems. we discriminate the failure state of a system into two different failure states; they are the fail safe state and the fail unsafe state. So far we have studied the system that any (safe or unsafe) failure leads to stop the system operation. The second problem in the use of the Markov chain modeling technique avoids a negative safety assessment for this type of system. Since large size Markov Chain models have the problem of state space explosion, so as to solve the second problem precisely and efficiently a Markov chain modular approach is going to be developed. In this paper scope of a frame work for safety assessment for safety-critical computer systems is depicted in figure 2-1

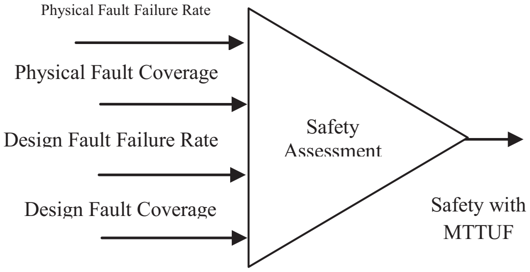

Figure 2-1 A frame work for Safety Assessment

The two problems I am going to deal with are inside of the box of a frame work for Safety Assessment, and so the contributions of the research are inside the box. It will not supply the techniques to advance the estimations of failure rates or coverage. Reviews of the failure rate of design fault, design fault coverage, physical fault failure rate, and the physical fault coverage are provided. Safety and MTTUF parameters are calculated based on the modeling techniques developed in this paper by assuming the availability of the failure rate and coverage parameters.

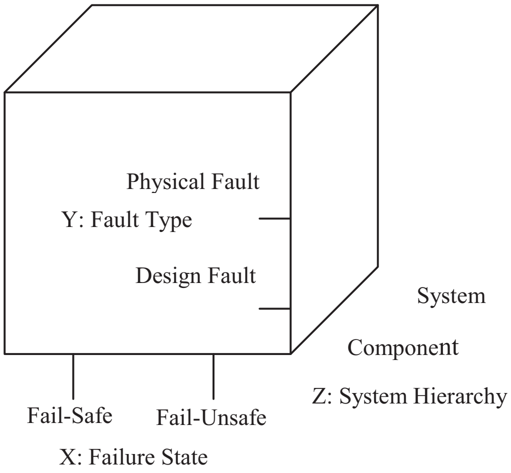

Figure 2-2 investigates Dimension Diagram

Let us consider the Y axis of my paper as the fault type axis. Reviews of design faults and physical faults are conducted. At the component level and at system level Safety models are built, incorporating both physical faults and design faults. Let us consider Z axis of my paper as the system hierarchy axis. Here the assumption made for the modeling of components is that the component does not utilize redundancy..

B. Fault Severity

Based on IEC 61508, the International standard in the functional safety of programmable/electronic/electrical electronic safety related systems, explains four severity categories known as safety integrity levels as shown in table A fault space can be partitioned into two subspaces: Subspace 1 contains the safe faults, and Subspace 2 contains the unsafe faults. By using the fault severity information, based on Carter’s coverage definition, we use the safe fault and unsafe fault theory to differentiate whether a fault is able to be recovered.

Severity Integrity Level	Importance of Safety - Related System Failure
1	Minor property and production protection
2	Minor property and production protection, Possible employee Injury
3	Employee and community protection
4	Catastrophic community impact

Table 2-1 Faulty Severity levels

	Computer Control System	Computer Safety System
Severity Integrity Level	Continuous /high – demand mode of operation (probability of dangerous failure per hour)	Low Demand mode of operation (probability of failure to perform its safety functions on demand)
1	$\geq 10^{-6}$ to $< 10^{-5}$	$\geq 10^{-2}$ to $< 10^{-1}$
2	$\geq 10^{-7}$ to $< 10^{-6}$	$\geq 10^{-3}$ to $< 10^{-2}$
3	$\geq 10^{-8}$ to $< 10^{-7}$	$\geq 10^{-4}$ to $< 10^{-3}$
4	$\geq 10^{-9}$ to $< 10^{-8}$	$\geq 10^{-5}$ to $< 10^{-4}$

Table 2-2 Acceptable Mishap Risk Figures

Thus, we have: the incident of the occurrence of safe fault at time t is independent of the event of the occurrence of fault at time t. Let us assume a fault occurs at a time, we are able to derive:

$$€ = \frac{\text{Occurrence of a safe at time t}}{\text{Occurence of a fault at time t}}$$

D. Markov Chain Model

A collection of random variables is known as stochastic process indexed by a parameter such as time, {X (t), t ≥0}. A Markov's chain is a particular type of stochastic process that has the Markov property. A stochastic process will have the Markov's property when the conditional probability distribution of future states of the process, given the current state, depends only upon the current state. A Markov chain consists of a finite number of states and transition probabilities „P_{ij}' which is probability of moving from state i into state j,

Chapman-Kolmogorov equations are defined as:

$$P_{ij}(t+s) = \sum_{r=0}^{k} P_{ir}(t) P_{rj}\ (s)$$

Where t, s≥0 and i , j ϵ ψ

$\boldsymbol{\Omega}_{ij}$: Denotes the „Transition rate' is the rate of a transition the process makes from state **i** into state **j.**

Ί$_{ij}$: Denotes the time a process spends in state **i** before entering state **j.** **Ί**$_{ij}$ is exponentially distributed with rate $\boldsymbol{\Omega}_{ij}$. Ross has proved (Ross, 1996):

$$\lim_{\Delta t \to \infty} \frac{P_{ij}\ (\Delta t)}{\Delta t} = \lim_{\Delta t \to \infty} \frac{P(\text{Ί}_{ij} < \Delta t)}{\Delta t} = \Omega_{ij}$$

Using the Chapman-Kolmogorov equations and the transition rate definition, the Kolmogorov Differential Equations can be derived as follows:

$$P^{I}{}_{ij}(t) = \sum_{r=0}^{k} \Omega_{ij} P_{ij}\ (t)$$

Ŧ denotes the Transition rate matrix related with a Markov chain as follows

$$\text{Ŧ} = \begin{pmatrix} \Omega_{11} & \cdots & \Omega_{1k} \\ \vdots & \ddots & \vdots \\ \Omega_{k1} & \cdots & \Omega_{kk} \end{pmatrix}$$

A „Transition rate matrix' is independent and it can be partitioned so that the diagonal contains two or more occurrences of the same block, where as the off-diagonal blocks must have (approximately) the same structure but they are not for small entries (Courtois, 1977).

E. Ternary Decision Diagram

In reliability analysis BDD (Binary Decision Diagram) modeling approach is used since system reliability analysis examines a system, The variable combined with each node is a Boolean variable in a BDD. The ternary Decision diagram is an enlargement of BDD of the three-state case (Sasao, 1997).In a TDD variable combined with non-sink node is a three-valued variable. For safety assessment TDD is considered as an efficient modeling approach, since it consists of three sink nodes describing fail-safe state, fail-unsafe state and operational state. Each of the non-sink nodes comprises of three edges shown in Figure 2-3.Here left outgoing edge is represented as "0" edge; and middle one is represented as "1" edge; and right one is represented as "2" edge. In order to denote the non-sink nodes in TDD three-valued variables are used.

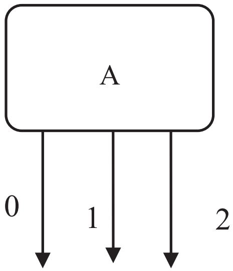

Figure 2-3, Non – Sink Node with Numbered Outgoing Edges

In this paper all the references to „Ternary Decision Diagram' means TDD with ordered node variables, represents each path beginning from root node to sink node visiting nodes by variables, in a rising order.

F. Three-state Markov Model

In order to form „safety-critical' systems a three-state Markov model consists of three-states, they are operational state, fail-safe state, and fail-unsafe state. When a system is operating correctly it is in the operational state and system always starts at operational state. When a safe-failure of the system causes it to go to fail-safe state, the transition probability from operational state to fail-safe state is P_{FS} (t). When an unsafe failure of the system causes it to go to fail-unsafe state and the transition probability from operational state to fail-unsafe state is P_{FU} (t). P_O (t): denotes probability of the system staying in operational state at time t. At any time system obeys the following equation $P_O(t) + P_{FS}(t) + P_{FU}(t) = 1$

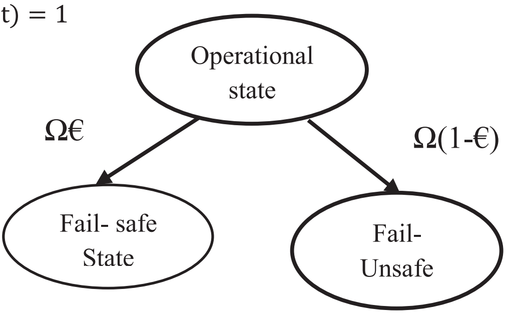

Figure 2-4, Markov Model with Transition Probabilities

The three-state homogeneous Markov model and probabilities information can be calculated as:

$$P_0(t) = e^{-\Omega t} \qquad 2\text{-}1$$

$$P_{FS}(t) = \text{€}e^{-\Omega t} \qquad 2\text{-}2$$

$$P_{FU}(t) = (1 - \text{€})e^{-\Omega t} \qquad 2\text{-}3$$

Where Failure rate: Ω and the coverage: €

G. System-Coverage

If a failure occurs, the ability of a system to fail safely is measured by system coverage. For safety-critical computer systems, this can be represented by three-state Markov model as shown in Figure 2-4, consists of two absorbing states, a system can prefer to stay in as time t go to infinity: fail-safe state and fail-unsafe state. From the three-state Markov model, safety metric is derived as given in the following equation:

$$S(t) = R(t) + P_{FS}(t) = P_0(t) + P_{FS}(t) \qquad 2\text{-}4$$

As time approaches to ∞, we get the steady-state safety:

$$S_{ss} = \lim_{t\to\infty} S(t) = \lim_{t\to\infty} P_0(t) + \lim_{t\to\infty} P_{FS} = \lim_{t\to\infty} P_{FS}(t)$$

For the three-state Markov model, S (t) is sum of the probability that a system stays in operational state at t, which is also reliability of the modeled system R(t), and probability that a system goes to the fail-safe state by t.

$$S_y\text{€}(t) = \frac{P(\text{Fails Safe state at t})}{P(\text{Fail at t})} = \frac{P_{FS}(t)}{P_{FS}(t)+P_{FU}(t)} = \frac{S(t)-R(t)}{1-R(t)}$$

Let time t tends to infinity,We obtain Steady-state system coverage

$$\lim_{t\to\infty} S_y\,\text{€}(t) = \lim_{t\to\infty} \frac{S(t)-R(t)}{1-R(t)} = \frac{\lim_{t\to\infty} S(t)-\lim_{t\to\infty} R(t)}{1-\lim_{t\to\infty} R(t)} = \frac{\lim_{t\to\infty} S(t)-0}{1-0} = S_{ss}$$

By the above derivation we can observe that the value of steady-state system coverage was actually equal to the value of steady-state safety. System coverage was a conditional probability where a system transitions to the fail-safe state if the system has failed (Johnson, 2005). Coverage measures ability that a system is able to recover if a fault occurs, while system coverage measures the ability that a system fails safely if a failure occurs.

III. Assessment of the Physical-faults and the Design-faults

The fault space is partitioned into physical-fault space and design-fault space.

A. Assessment on Physical Faults

Let T denotes the time to failure, and the „Failure distribution function’ is probability of an item failing in the time interval [0, t]. The Failure distribution function, H (t),

$$H(t) = P(T \leq t) \qquad 0 < t < \infty \qquad 3\text{-}1$$

The reliability function is probability of an item that does not fail during the time interval [0, t). The reliability function is defined by

$$R(t) = P(T > t) \qquad 0 < t < \infty \qquad 3\text{-}2$$

The probability density function refers to a probability distribution in the form of integrals. It is non-negative all over and its integral ranges from -∞ to +∞ is equal to one. When a probability distribution has a density h (t), then the interval [t, t + Δt] has probability h (t).Δt to fail. Thus, we are able to derive:

$$h(t) = -\frac{H(t+\Delta t)-H(t)}{\Delta t} \qquad 3\text{-}3$$

The probability for an item to fail during time interval [t, t Δt] given an item is functioning at time T is

$$P(t < T < t + \Delta t \mid T > t) = \frac{P(t < T < t+\Delta t)}{P(T>t)} = \frac{H(t+\Delta t)-H(t)}{R(t)} \quad 3\text{-}4$$

On distributing the probability for the length of the time interval Δt, and letting Δt →0, we get the item’s failure rate function Ω(t) [Rausand2004] as follows,

$$\Omega(t) = \lim_{\Delta t\to 0} \frac{P(t<T<t+\Delta t|T>t)}{\Delta t} = \lim_{\Delta t\to 0} \frac{H(t+\Delta t)-H(t)}{R(t)\Delta t} \qquad 3\text{-}5$$

$$\Omega(t) = -\frac{h(t)}{R(t)}$$

If the homogeneous population of an item is very large, Ω(t). Δt is roughly equal to the number of items that are operating at time t but will fail in (t, t + Δt], the failure rate at any time t is conditioned on the components still operating at t. We define a random variable *Y* that follows a Bernoulli distribution. *Y* is defined as

$$Y = \begin{cases} 1 & \forall \text{ detected physical fault} \\ 0 & \forall \text{ undetected physical fault} \end{cases}$$

$y_{j,}$ j∊ [1, n] are a series of Bernoulli trails. y_j has the value of 1 when a physical fault occurs the physical fault can be detected; y_j has the value of 0 when a physical fault. Occurs the physical fault cannot be detected. Thus, the physical fault coverage can be represented as

$$\text{€}_p = \frac{\sum_{j=1}^{N} y_j}{N} \qquad 3\text{-}6$$

B. Assessment on Design Faults

Design faults in the form of procedural deficiencies and design inadequacies can be introduced in the design process as the designer tries to meet the “expected action” of the system in terms of the user’s expectations and needs (Dunn,2002), Design faults exist as hidden faults in written files, like design specification, program code, and system design execution.

Let us study the Jelinski-Moranda (J-M) model as an example. The set of additional assumptions made for the J-M model is:

1) Let N be an unknown permanent number of faults, in the software at the starting of the time in which the software is observed.

2) If a fault will come it is removed immediately.

3) A software failure rate remains constant above the intervals between failure occurrences and is proportional to the current fault content of the software.

Le*t* N is the total number of software faults in the software at *t* = 0; φ is the proportionality constant; and *t* is a point in time between the occurrence of the (k -1)*th* and the kth fault occurrence, and the expected time between failures at time *t* is $\frac{1}{\varphi.(N-(k-1))}$. If the time between-failure intervals are $X_k = T_k - T_{k-1}, k = 1 \ldots. n$ where X_k ' s are independent exponentially distributed random variables and E[X_k]= $\frac{1}{\varphi.(N-(k-1))}$. The probability density function for X_k is

$$h(X_k) = \frac{e^{\frac{-X_k}{E(X_k)}}}{E(X_k)} = \varphi.\big(N-(k-1)\big)e^{-\varphi.(N-(k-1))X_k} \qquad 3\text{-}7$$

If there is only one software fault at *t* = 0. Thus, N=1 and k = 1. The probability density function of the happening of the first failure is

The corresponding cumulative function is

$$h(t) = \varphi . e^{-\varphi t} \qquad 3\text{-}8$$

Cumulative function is $\mathrm{H(t)} = 1 - \mathrm{e}^{-\varphi \mathrm{t}}$ 3-9

Let us consider the total number of software faults at *t* = 0, N is the expected number of software faults occurred in the interval (0, t], m (t)*,* can be obtained:

$$\mathrm{m}(t) = N.H(t) = N(1 - e^{-\varphi t}) \qquad 3\text{-}10$$

Maximum Likelihood Estimation (MLE) can be used to approximate values of N and

φ statistically. Thus, the estimated $\widehat{N}$ and the estimated $\widehat{\varphi}$ from MLE can then be used in place of *N* and φ.

Corollary 3.1.

Let us assume that all design faults have an equal probability of occurrence and the failure density function of a design fault is $h_D(t)$ with $\int_0^\infty h_D(t)dt = 1$. $\tilde{N}_D$ (t) represents the estimated number of design faults at t, $\tilde{N}_{DS}(t)$ represents the estimated number of safe design faults at t .

So, P {a design fault becomes active at t} and P {a safe design fault becomes active at t} is determined by Equation 3.11 and Equation 3.12 respectively.

P {a design fault becomes active at t}

$$= \tilde{N}_D\ (t).\ (h_D\ (t).\ \Delta t)(1 - (h_D\ (t).\ \Delta t))^{\tilde{N}_D\ (t)-1} \qquad 3\text{-}11$$

P {a safe design fault becomes active at t}

$$=\tilde{N}_{DS}\ (t).\ (h_D\ (t).\ \Delta t)(1 - (h_D\ (t).\ \Delta t))^{\tilde{N}_D\ (t)-1} \qquad 3\text{-}12$$

Proof.

By using a binomial distribution, Binomial (N_D, $h_D(t)$) , to model the process. Each design fault is regarded as an independent trial. The activation of a design fault is clear as a success of a trial. N_D, the number of design faults, now represents the number of independent trials. During [t, t + Δ t], $h_D(t).\Delta t$, the probability of a design fault becoming active at t, will be the probability of a success of a trial. Suppose *S* is a random variable, and the value of *S* denotes the number of successes in the N_D independent trials with the probability of success of each trial at t being h_D (t).Δt. We estimate the probability that one of N_D design faults becoming active at t, so assigning the value of 1 to S:

P(S=1) = P {a design fault becomes active at t}

$$=\left[\frac{\tilde{N}_D(t)}{1}\right]((h_D\ (t).\ \Delta t)(1 - (h_D\ (t).\ \Delta t))^{\tilde{N}_D\ (t)-1} \qquad 3\text{-}13$$

$$=\tilde{N}_D(t)((h_D\ (t).\ \Delta t)(1 - (h_D\ (t).\ \Delta t))^{\tilde{N}_D\ (t)-1} \qquad 3\text{-}14$$

To estimate the probability that a safe design fault becomes active at t, let us define a random variable Z, which follows a Bernoulli distribution. Y is defined as

$$\mathrm{Y} = \begin{cases} 1 & \forall \text{safe design faults} \\ 0 & \forall \text{unsafe design faults} \end{cases} \qquad 3\text{-}15$$

At given time t, we estimate the probability that a design fault is a safe fault, and thus P (Y = 1). We define a space of elementary events, £, which contains $\tilde{N}_D$ (t) design faults. $\tilde{N}_{DS}$ (t) of the $\tilde{N}_D$ (t) design faults are safe faults. So, on the space of elementary events £, the probability that a design fault is a safe fault is:

$$\mathrm{P(Y = 1)} = \frac{\tilde{N}_{DS}\ (t)}{\tilde{N}_D\ (t)} \qquad 3\text{-}16$$

Let us consider Y and S is two independent random variables. If the probability that a design fault is a safe fault, P (Y = 1), and the probability that one of N_D design faults becomes active at t, P(S = 1), we estimate the probability that a safe design fault becomes active at t:

P {a safe design fault becomes active at t}

=P(Y=1∩S=1)

=P(Y=1).P(S=1)

$$= \frac{\tilde{N}_{DS}\ (t)}{\tilde{N}_D\ (t)}.\ \tilde{N}_D(t)((h_D\ (t).\ \Delta t)(1 - (h_D\ (t).\ \Delta t))^{\tilde{N}_D\ (t)-1}$$

$$=\tilde{N}_{DS}(t)((h_D\ (t).\ \Delta t)(1 - (h_D\ (t).\ \Delta t))^{\tilde{N}_D\ (t)-1} \qquad 3\text{-}17$$

IV. ANALYSIS OF EXAMPLE

The processing support to work out command and control over flight operations is given by Space Shuttle Ground System is given by the flight controllers at the Johnson space center. Actually the real-time software system of the Space Shuttle Ground System has more than one-half million lines of source code. During 2656.9 testing hour record of 39 time intervals was released in Misra (Misra, 1983). For the real-time software system the software testing information is provided by the dataset up to 2656.9 hours. Let us consider the real-time software system as one component of the Space Shuttle Ground System, and we perform a framework for safety assessment for this component. All software faults are considered as design faults. The design-fault failure rate and design-fault coverage of the component can be predicted using information shown in Table 4-1. Let us suppose physical-fault failure rate of the real-time software system is zero.

Safety can be estimated by using the parameterized three-state homogeneous Markov model. Let us suppose the time to eliminate a safe fault after it is discovered is small and can be neglected. Thus, MTTUF

Interval	Test	Critical	Major	Minor
1	62.5	0	6	9
2	44.0	0	2	4
3	40.0	0	1	7
4	68.0	1	1	6
5	62.0	0	3	5
6	66.0	0	1	3
7	73.0	0	2	2
8	73.5	0	3	5
9	92.0	0	2	4
10	71.4	0	0	2
11	64.5	0	3	4
12	64.7	0	1	7
13	36.0	0	3	0
14	54.0	0	0	5
15	39.5	0	2	3
16	68.0	0	5	3
17	61.0	0	5	3
18	62.6	0	2	4
19	98.7	0	2	10
20	25.0	0	2	3
21	12.0	0	1	1

22	55.0	0	3	2
23	49.0	0	2	4
24	64.0	0	4	5
25	26.0	0	1	3
26	66.0	0	2	2
27	49.0	0	2	0
28	52.0	0	2	2
29	70.0	0	1	3
30	84.5	1	2	6
31	83.0	1	2	3
32	60.0	0	0	1
33	72.5	0	2	1
34	90.0	0	2	4
35	58.0	0	3	3
36	60.0	0	1	2
37	168.0	1	2	11
38	111.5	0	1	9
39	200.0	0	5	9

Table 4-1 the Software Fault Dataset of the Space Shuttle Ground System

Example 1: Safety Assessment at t = 2656.9.

The last update shown in Table 4-1 is made at t = 2656.9. By applying the Goel- Okumoto model to the fault dataset, the component failure rate is estimated to be 0.06419 per testing hour at t = 2656.9.

Time	Failure Rate	Number of Faults	Number of Safe Faults	Coverage (€)
t=2656.9	0.06419	493.98	474.59	0.960758

Table 4-2 Estimated Parameters at t=2656.9

At t=2656.9, the safety of the component is estimated using the following equation:

$$S(t) = e^{-\Omega(t-2656.9)} + €\left(1 - e^{-\Omega(t-2656.9)}\right)$$

$$= \mathrm{e}^{-0.06419\,(\mathrm{t}-2656.9)} + 0.96075\left(1 - \mathrm{e}^{-0.06419\,(\mathrm{t}-2656.9)}\right)$$

For example, the estimated safety value at t = 2657.9, one testing hour after the last update, is

$$\mathrm{S}(\mathrm{t} = 2657.9) = \mathrm{e}^{-0.06419} + 0.96075(1 - \mathrm{e}^{-0.06419}))$$

=0.9975597

At t = 2656.9, the MTTUF of the component is estimated to be

$$\mathrm{MTTUF} = \frac{\mathrm{MTTF}}{1-€} = \frac{1}{\Omega(1-€)}$$

$$=\frac{1}{0.06419\,(1-0.96075)} = 396.406$$

Since we assume a fault of a component causes a failure of the component, the MTTUF can be estimated to predict the testing hours required to discover the next unsafe fault. The estimated MTTUF at t = 2656.9 means that the average time to discover the next unsafe fault requires about 400 testing hours if the testing method and the testing personals do not change. Both the estimated safety and the estimated MTTUF are valid until the next software fault is found.

V. CONCLUSION

In this paper, the concept of fault space is introduced and the partition of this space based on safe faults and unsafe faults is discussed. The safety modeling fundamentals using Markov chain models and using TDD models are overviewed. Using the Splitting Poisson Process, the failure rate from the operational state to the fail-safe state and the failure rate from the operational state to the fail-unsafe state can be determined by the failure rate Ω and the coverage €. Safety and reliability are briefly compared and their relationship is discussed. The reviews on the physical fault failure rate, the physical fault coverage, the design fault failure rate, and the design fault content are supplied in the paper. state Markov chain model reviewed in 3.By using the parameterized three-state Markov model Component safety assessment is conducted by the assumption that component does not utilize redundancy.

References

[1] T. Anderson and P. A. Lee, Fault Tolerance Principles and Practice, Prentice-Hall International, 1981.

[2] T. Anderson and J. C. Knight, A Framework for Software Fault Tolerance in Real-Time Systems, IEEE Transactions On Software Engineering, vol. SE-9, no. 3, pp. 355-364, 1983.

[3] Paul Anderson, Detecting Bugs in Safety Critical Code , Dr. Dobbs Journal, February, 2008 [Avizienis76] A. Avizienis, Fault Tolerant Systems, IEEE Transactions On Computers, vol. C-25, pp. 1304- 1312, 1976.

[4] T. Bedford and R. Cooke, Probabilistic Risk Analysis: Foundations and Methods, Cambridge Univ. Press, 2001.

[5] J. V. Bukowski and W. M. Goble, "Defining mean time-to-failure in a particular failure-state for multi-failure-state systems," IEEE Trans. Reliab, vol. 50, no. 2, pp. 221–228, Jun. 2001.

[6] C. Y. Choi, B. W. Johnson, and J. A. Profeta III, "Safety issues in the comparative analysis of dependable architectures," IEEE Trans. Reliab., vol. 46, no. 3, pp. 316–322, Sep. 1997.

[7] H. Choi, W. Wang, and K. S. Trivedi, "Conditional MTTF and its computation

[8] in Markov reliability models," in Proc. 1993 Annu. Reliability and Maintainability Symp., Jan. 25–28, 1993, pp. 55–63.

[9] W. M. Goble, Control Systems Safety Evaluation and Reliability: Instrument Society of America, 1998.

[10] IEEE 100, "The Authoritative Dictionary of IEEE Standard Terms," IEEE Press, 2000.

[11] Arlat, J., Y. Crouzet, and J.-C. Laprie, "Fault Injection for the Experimental Validation of Fault Tolerance," LAAS Report 90415,1990.

[12] Avizienis, A., "The Four-Universe Information System Model for the Study of Fault Tolerance," Proceedings of the 12th Annual International

[13] Bowen, J., "The Ethics of Safety-Critical Systems," Communications of ACM, Vol. 43, Issue 4, 2000, pp. 91-97.