2012 International Conference on Information and Computer Networks (ICICN 2012)
IPCSIT vol. 27 (2012) © (2012) IACSIT Press, Singapore

A Uniform and Expanded Quantification Approach for Cut Sequence Set Model

Bo Wang [1+], Dong Liu [1] and Yi Li [2]

[1] National Laboratory of EIES, Academy of Equipment, Beijing, China

[2] Department of Scientific Research, Academy of Equipment, Beijing, China

Abstract. Cut Sequence Set (CSS) model is an analytical method to overcome the shortcomings of the state-space-based and simulation-based methods for dynamic fault trees (DFTs) analysis. Temporal failure logic is an accurate, unambiguous formal characterization for sequence of basic events in DFTs. However, the original quantitative analysis of the CSS model is not defined using temporal failure logic, which brings about the ambiguity and inconsistency for quantitative methods. To vanquish these drawbacks, a formal description of temporal failure logics for dynamic gates in DFTs is proposed firstly, including priority failure logic, sequential failure logic and spares' failure logic. Then, using proposed temporal failure logics, this paper presents a uniform and expanded quantification approach of CSS model via employing the concept of temporal intervals. An example demonstrates the effectiveness of the proposed approach.

Keywords: Dynamic fault trees, cut sequence set, temporal failure logic, quantification

1. Introduction

Fault trees (FTs) are widely used in system reliability analysis, which provides a simple and intuitive way to describe the failure behaviour of the system. However, with the improvement reliability requirements, there have been more and more dynamic systems, whose components embodied with the order of dependency. The traditional FTs methods cannot model such dynamic systems. Dugan et al [1] proposed dynamic FTs (DFTs) in 1992 to capture the dynamic failure mechanism of a system having sequence-depending components. DFTs have witnessed a large attention in the past few years.

As considering the timing dependencies between events, combinatorial logic is not valid for DFTs, the typical solution is converting the DFTs to a continues time Markov model [2]. However, Markov model has to face with state space explosion problem, which greatly limits the application. Recently, an effective way to alleviate the state space explosion is Cut Sequence Set (CSS) model [3].

In a dynamic system, the system's failure depends not only the combinational logic of the basic events, but also the sequence relationship of basic events. To address the sequence, CSS model extends the concept of cut set in FTs into cut sequence in DFTs. Cut sequence is the set of ordered basic events which could lead to the failure of system (i.e., top event). CSS model includes two processes, namely the generation of quantification of cut sequence. Temporal failure logic is an accurate, unambiguous formal characterization for sequence of basic events. However, the quantitative analysis of the original CSS model is not defined using temporal failure logic, but proceed directly from the natural language-based definition, which resulting in limitedness to the simple, general situations. Moreover, it brings about the ambiguity and inconsistency for quantitative methods. In order to overcome these drawbacks, in this paper, the temporal failure logic of dynamic gates failure logic (including the event's starting point, end point, the state switching time, etc.) is described precisely. Then the definitions of the priority failure logic, sequential failure logic and spares'

[+] Corresponding author. Tel.: +8613701103586; fax: +8610-82738397.
E-mail address: wowbob@139.com.

failure logic are given. At last, an expanded and unified quantitative approach based on temporal failure logic is proposed and verified.

2. Hypothesis

The following are some hypothesis in this paper:

- Failure times of basic events are mutually statically independent.
- $f_x(t)$ denotes probability density function of basic event x, which is integrable or generalized integrable.
- The probability of two or more failures during the period $[t, t+\mathrm{d}t)$ is assumed to be $o(\mathrm{d}t)$ for any input.

3. Description of Temporal Failure Logic

By analyzing sequence relationships of basic events in dynamic gates for DFTs, we abstract them into three categories:

(1) *Priority failure logic*, including priority-AND gate (PAND), functional trigger gate (FDEP), AND and OR;

(2) *Sequential failure logic*, including the Cold spare gate (CSP), Sequence enforcing gate (SEQ);

(3) *Spares' failure logic*, including warm spare doors (WSP) and hot spare gates (HSP).

We will describe them in detail. In the following analysis, we assume that the input basic events are $X=\{<x_1, x_2, \ldots, x_{n-1}, x_n>\}$ (n is a positive integer), basic events' start time (age point), life (age) and the failure time are correspondingly to the $\boldsymbol{\tau}^0 = \{\tau_1^0, \tau_2^0, \cdots, \tau_{n-1}^0, \tau_n^0\}$, $\boldsymbol{\tau} = \{\tau_1, \tau_2, \cdots, \tau_{n-1}, \tau_n\}$ and $\boldsymbol{\tau}^F = \{\tau_1 + \tau_1^0, \tau_2 + \tau_2^0, \cdots, \tau_{n-1} + \tau_{n-1}^0, \tau_n + \tau_n^0\}$

3.1. Priority Failure Logic

Priority failure logic is parallel state dependencies (with the same starting point), which satisfies the definitions and natures:

(1) $x_1, x_2, \ldots, x_{n-1}, x_n$ begin to work at t=0 with normal failure rates; (2) x_1 fails at $t = \tau_1$, and the failure state lasts over $t = \tau_n$; (3) x_2 fails after x_1 at $t = \tau_2$, and also, the failure state lasts over $t = \tau_n$; (4) In accordance with (3), $x_3, x_4, \ldots, x_{n-1}$ fail by sequence, similarity, the failure state lasts over $t = \tau_n$; (5) x_n fails at $t = \tau_n$, the output (top event) occurs; (6) The life of the system is τ_n.

3.2. Sequential Failure Logic

Sequential failure logic reflects the input events of serial dependency on the state (with different starting points), and can be defined as follow:

(1) $\boldsymbol{\tau}^0 = \{\tau_1^0, \tau_2^0, \cdots, \tau_{n-1}^0, \tau_n^0\} = \{0, \tau_1, \cdots, \sum_{k=1}^{n-2}\tau_k, \sum_{k=1}^{n-1}\tau_k\}$, namely, x_1 begins to work at t=0 with normal failure rates, the left basic evens $x_2, \ldots, x_{n-1}, x_n$ don't work; (2) x_1 fails at $t = \tau_1$, the failure state lasts over $t = \tau_n$; x_2 replaces x_1 ,starting to work with normal failure rates; (3) after a time period τ_2, x_2 fails , then x_3 replaces x_2 and works normally; (4) In accordance with (3), $x_3, x_4, \ldots, x_{n-1}$ fail at time series $\tau_1 + \tau_2 + \tau_3, \tau_1 + \cdots + \tau_4, \cdots, \tau_1 + \cdots + \tau_{n-1}$ by sequence, similarity, the failure state lasts over $t = \tau_1 + \tau_2 + \cdots + \tau_{n-1} + \tau_n$; (5) x_n fails at $t = \tau_1 + \tau_2 + \cdots + \tau_{n-1} + \tau_n$, the output (top event) occurs; (6) The life of the system is $\tau_1 + \tau_2 + \cdots + \tau_{n-1} + \tau_n$.

3.3. Spares' Failure Logic

Spares' failure logic is more complex. There are primary components and one or more replaceable (reserved) components for spare gates. As System's start up, primary components are running, while reserved components not, or dormant running, reserved components replace those failed components. When all the input events occur, the spare gates' output generates. Since the spares change the status from dormant to active, and the conversion depends on the state of the previous parts, so we recursively define this logic.

Let $\boldsymbol{\tau}' = \{\tau_1', \tau_2', \cdots, \tau_{n-1}', \tau_n'\}$ denotes ages of each spares, and $T_k (k = 1, 2, \cdots, n)$ denotes the k-input spare gate. Firstly, we introduce indicator function.

Definition (*indicator function* I (A)): I (A) = 1, if A is not empty, otherwise I (A) = 0.

Definition (*spares' failure logic*):

(1) $\boldsymbol{\tau}^0 = \{\tau_1^0, \tau_2^0, \cdots, \tau_{n-1}^0, \tau_n^0\} = \{0, T_1, \cdots, T_{n-1}\}$; (2) $T_1 = \tau_1$; (3) If $\tau_2' < T_1$, then $T_2 = T_1$, otherwise $T_2 = T_1 + \tau_1$; (4) $T_i = T_{i-1} + I(\tau_i' > T_{i-1})\tau_i = \tau_1 + I(\tau_2' > T_1)\tau_2 + I(\tau_3' > T_2)\tau_3 + \cdots + I(\tau_i' > T_{i-1})\tau_i (i = 3, \cdots, n)$;(5) Life of the system is $T_n = \tau_1 + I(\tau_2' > T_1)\tau_2 + I(\tau_3' > T_2)\tau_3 + \cdots + I(\tau_n' > T_{n-1})\tau_n$.

4. An uniform Quantification method

Temporal logic depicts accurately events' temporal intervals in fact, we denote L_1, L_2, … , L_{n-1}, L_n as temporal intervals of x_1, x_2, …, x_{n-1}, x_n correspondingly. As a starting point, we define the uniform quantification process:

(1) Using temporal failure logic, generate temporal intervals of $x_1, x_2, \ldots, x_{n-1}, x_n$, *i.e.* $L_1, L_2, \ldots, L_{n-1}, L_n$;

(2) Calculate the following multiple integrals

$$\Pr\{X \mid t\} = \int_{L_n}\int_{L_{n-1}} \cdots \int_{L_2}\int_{L_1} f_{x_n}(t_n) f_{x_{n-1}}(t_{n-1}) \cdots f_{x_2}(t_2) f_{x_1}(t_1) dt_1 dt_2 \cdots dt_{n-1} dt_n \quad (1)$$

Where, for a given time, $\tau_n = t$.

For the simple distribution function (such as the exponential distribution) of events, we can obtain an analytical solution; for more complex distribution function, mathematical software can be used to get a numerical approximation. It is noteworthy that, in most cases, temporal logic generated from step (1) is a blend of different logic, such as priority failure logic, sequential failure logic, while the original method proposed in [4] cannot deal with these problems. So the approach presented in this paper improves and expand the existing methods.

5. Application

We consider this method to solve an instance - HCSE (Hypothetical Computer System Example) system [5], the system's structure is shown in Figure 1.

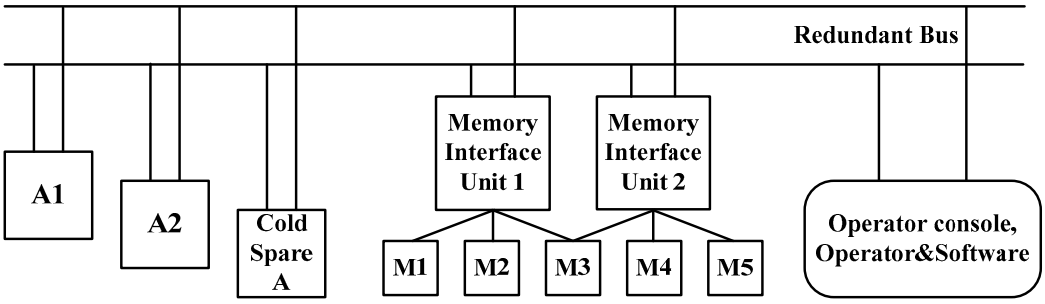

Fig 1 System structure of HCSE

There are four subsystems (corresponding to four sub-trees) of HCSE, i.e. PSF (Processing system failure), MSF (Memory system failure), BSF (Bus system failure), AF (Application failure).

- *T1:* Processor subsystem has two redundant processors A1, A2 and a cold backup processor Aa, once any one of redundant processors fails, Aa will replace the failed processor. A1, A2 and Aa are ideal processors, using the same operating system.
- *T2:* Memory subsystem has five memories, system works at least three of them. Memories are connected through the memory interface units. If the memory interface fails, the memories of their connection will not work. M3 connects to the memory interface unit 1 and 2, which means as long as there is a memory interface unit normal, M3 can be used normally.
- *T3:* Bus subsystem is relatively simple, containing only two redundant buses, both the two fail will result in bus subsystem's failure.
- *T4:* Application subsystems take into account the operator (OP), hardware (HW) and software (SW) impacts on the system. OP, HW and SW failures will lead to a system failure.

From the above analysis, HCSE systems' DFT is shown in Figure 2.

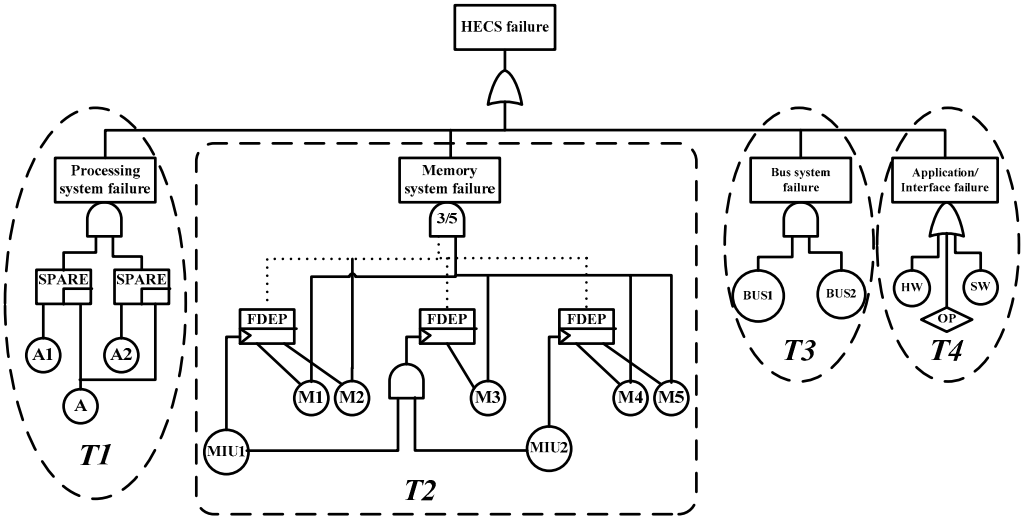

Fig 2 DFT of HCSE

We first consider *T1*, the processor subsystem. With methods proposed in [3], its cut sequence can be generated as:

$$
\begin{aligned}
&T1 = SPARE_{\ 1} \cdot SPARE_{\ 2} \\
&= \{(A_1 \to A_a) + (A_2 \to A_1)\} \cdot \{(A_2 \to A_a) + (A_1 \to A_2)\} = \{(A_1 \to A_a)(A_2 \to A_a) + (A_1 \to A_a)(A_1 \to A_2) + \\
&\quad (A_2 \to A_1)(A_2 \to A_a) + (A_2 \to A_1)(A_1 \to A_2)\} = \{(A_1 \to A_2 \to A_a) + (A_2 \to A_1 \to A_a) + \\
&\quad (A_1 \to A_a \to A_2) + (A_1 \to A_2 \to A_a) + (A_2 \to A_1 \to A_a) + (A_2 \to A_a \to A_1) + \varnothing\} \\
&= \{(A_1 \to A_2 \to A_a) + (A_2 \to A_1 \to A_a) + (A_1 \to A_a \to A_2) + (A_2 \to A_a \to A_1)\} \\
&= \{(A_1 \to A_2 \to A_a) + (A_2 \to A_1 \to A_a) + (A_1 \to A_a \to A_2) + (A_2 \to A_a \to A_1)\} \triangleq CS_1 + CS_2 + CS_3 + CS_4
\end{aligned}
$$

There are four cut sequences in T1, take $A_1 \to A_2 \to A_a$ for instance, from the description in Section 3, we know : A_1 and A_2 are in line with priority failure logic, for that they are combined by AND gate, while A2 and A_a coincides with spares' failure logic. The original quantification method of CSS cannot handle with this mixed logic, however, it can be solved easily via the proposed approach in this paper, which is:

Step 1: Generating of temporal intervals, which is shown in Table I.

TABLE 1 TEMPORAL INTERVALS OF EVENTS

	A_1	A_2	A_c
CS_1	$L_{11}=[0, \tau(A_1)]$	$L_{12}=[0, t]$	$L_{1c}=[\tau(A_1), \tau(A_c)]$
CS_2	$L_{21}=[0, \tau(A_1)]$	$L_{22}=[0, \tau(A_2)]$	$L_{2c}=[\tau(A_2), t]$
CS_3	$L_{31}=[0, \tau(A_1)]$	$L_{32}=[0, \tau(A_2)]$	$L_{3c}=[\tau(A_1), t]$
CS_4	$L_{41}=[0, t)]$	$L_{42}=[0, \tau(A_2)]$	$L_{4c}=[\tau(A_2), \tau(A_c)]$

Step 2: Using equation (1) in Section 4, we obtain multiple integral form of quantitative models as:

$$
\begin{aligned}
\Pr\{T1\} &= \sum_{i=1}^{4} \int_{L_{i1}} \int_{L_{i2}} \int_{L_{ic}} f_{A_1}(t_1) f_{A_2}(t_2) f_{A_c}(t_c) dt_1 dt_2 dt_c \\
&= \int_0^t \left(\int_0^{\tau(A_1)} \left(\int_{\tau(A_1)}^{\tau(A_c)} f_{A_1}(t_1, t_c) dt_1 \right) f_{A_c}(t_c, t_2) dt_c \right) f_{A_2}(t_2) dt_2 + \int_{\tau(A_2)}^t \left(\int_0^{\tau(A_2)} \left(\int_0^{\tau(A_1)} f_{A_1}(t_1, t_2) dt_1 \right) f_{A_2}(t_2, t_c) dt_2 \right) f_{A_c}(t_c) dt_c + \\
&\quad \int_{\tau(A_1)}^t \left(\int_0^{\tau(A_1)} \left(\int_0^{\tau(A_2)} f_{A_2}(t_2, t_1) dt_2 \right) f_{A_1}(t_1, t_c) dt_1 \right) f_{A_c}(t_c) dt_c + \int_0^t \left(\int_{\tau(A_2)}^{\tau(A_c)} \left(\int_0^{\tau(A_2)} f_{A_2}(t_2, t_c) dt_2 \right) f_{A_c}(t_c, t_1) dt_c \right) f_{A_1}(t_1) dt_1
\end{aligned}
\tag{2}
$$

If the failure distribution of A_1, A_2 and A_a is known, using equation (2), we can quickly get the analytical or numerical solution. Quantitative model of the original CSS are good at handling events with complex failure logic, while our method is clear and unambiguous to resolve it.

Other subsystems can be handled using traditional FTs and will not repeat here. Finally, we get the CSS model of HECS:

$$HECS = PSF + MSF + BSF + AF$$
$$= (A1 \to A \to A2) + (A1 \to A2 \to A) + (A2 \to A1 \to A) + (A2 \to A \to A1) + M_1 \cdot M_2 \cdot M_3 + M_1 \cdot M_2 \cdot M_4 +$$
$$M_1 \cdot M_2 \cdot M_5 + M_1 \cdot M_3 \cdot M_4 + M_1 \cdot M_4 \cdot M_5 + M_1 \cdot M_3 \cdot M_5 + M_2 \cdot M_3 \cdot M_4 + M_2 \cdot M_4 \cdot M_5 + M_2 \cdot M_3 \cdot M_5 +$$
$$M_3 \cdot M_4 \cdot M_5 + MIU_1 \cdot MIU_2 + MIU_1 \cdot M_3 + MIU_1 \cdot M_4 + MIU_2 \cdot M_1 + MIU_2 \cdot M_3 + BUS_1 \cdot BUS_2 + HW + SW + OP$$

And quantitative model is:

$$\Pr\{HECS\} = \Pr\{T1\} + \Pr\{T2\} + \Pr\{T3\} + \Pr\{T4\} - \Pr\{T1\} \cdot \Pr\{T2\} - \Pr\{T2\} \cdot \Pr\{T3\} - \Pr\{T3\} \cdot \Pr\{T4\} -$$
$$\Pr\{T4\} \cdot \Pr\{T1\} + \Pr\{T1\} \cdot \Pr\{T2\} \cdot \Pr\{T3\} + \Pr\{T2\} \cdot \Pr\{T3\} \cdot \Pr\{T4\} + \Pr\{T3\} \cdot \Pr\{T4\} \cdot \Pr\{T1\} -$$
$$\Pr\{T1\} \cdot \Pr\{T2\} \cdot \Pr\{T3\} \cdot \Pr\{T4\}$$

Consider the special case of exponential distribution, the relevant failure rates is shown in Table II. Simulation time interval is [0,400] h. As Figure 3 shows, when the simulation carried out to 200h, HECS unreliability reached 99.87%.

TABLE 2 FAILURE RATES OF EVENTS

Basic events	Failure rates (h^{-1})
A_1, A_2, A	10^{-4}
M_1, M_2, M_3, M_4, M_5	6×10^{-5}
MIU_1, MIU_2	5×10^{-5}
BUS_1, BUS_2	10^{-6}
HW	5×10^{-5}
SW	3×10^{-2}
OP	10^{-3}

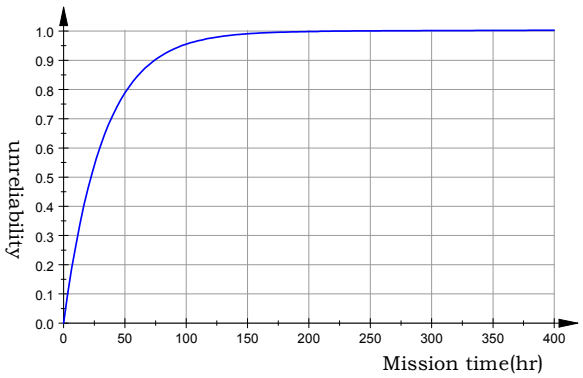

Fig 3 System Unreliability

6. Conclusion

To overcome the fuzziness, inconsistency and limitation in Cut Sequence Set (CSS) model for DFTs, a precise, formal description and characterization for dynamic gates is proposed, including priority failure logic, sequential failure logic and spares failure logic of CSS model. Using temporal failure logic, this paper presents a uniform and expanded quantification approach of CSS model via employing the concept of temporal intervals. An example demonstrates the effectiveness of the proposed approach. In the future, we will focus on related software tools' development.

7. Acknowledgment

This work was supported by National Natural Science Foundation of China (No. 60904082).

8. References

[1] J. B. Dugan, S. Bavuso, M. Boyd. Dynamic fault tree models for fault tolerant computer systems. *IEEE Transactions on Reliability*, vol. 41, 1992, pp.363-377.

[2] L. Meshkat, J. B. Dugan, J. D. Andrews. Dependability analysis of systems with on-demand and active failure modes, using dynamic fault trees. *IEEE Transactions on Reliability*, vol.51, 2002, pp. 240-251.

[3] L. Dong, X. Weiyan, Z. Chunyuan, L. Ri, and H. Li. Cut sequence set generation for fault tree analysis. *Proc. of International Conference on Embedded Software and Systems*. Daegu, South Korea, 14-16 May 2007,LNCS 4523, pp.58-69

[4] L. Dong, Z. Chunyuan, X. Weiyan, L. Ri,and H. Li. Quantification of cut sequence set for fault tree analysis. *Proc. of International Conference on High Performance Computing and Communication*. Houston, USA, 2007, LNCS 4782, pp.755 -765.

[5] W.E. Vesely, M. Stamatelatos, J. B. Dugan, et al. Fault tree handbook with aerospace applications. *NASA Office of Safety and Mission Assurance*, USA. August, 2002, pp.157-161