Quantification of Simultaneous-AND Gates in Temporal Fault Trees

Ernest Edifor, Martin Walker, and Neil Gordon

Department of Computer Science, University of Hull,
Cottingham Road, Hull, HU6 7RX, UK
{e.e.edifor@2007,martin.walker@,n.a.gordon@}hull.ac.uk

Abstract. Fault Tree Analysis has been a cornerstone of safety-critical systems for many years. It has seen various extensions to enable it to analyse dynamic behaviours exhibited by modern systems with redundant components. However, none of these extended FTA approaches provide much support for modelling situations where events have to be "nearly simultaneous", i.e., where events must occur within a certain interval to cause a failure. Although one such extension, Pandora, is unique in providing a "Simultaneous-AND" gate, it does not allow such intervals to be represented. In this work, we extend the Simultaneous-AND gate to include a parameterized interval – referred to as *pSAND* – such that the output event occurs if the input events occur within a defined period of time. This work then derives an expression for the exact quantification of pSAND for exponentially distributed events and provides an approximation using Monte Carlo simulation which can be used for other distributions.

1 Introduction

The effects of technology now pervade almost every sphere of life, increasing human dependency on them. The failures of some of these systems can have devastating effects on human life and the environment. Such systems with catastrophic effects are known as high consequence or safety-critical systems, so assessing their reliability is of increasing importance. Some modern systems depend on duplicated or stand-by components to improve their reliability. However, this feature poses other challenges for system designers who need to model and evaluate system reliability appropriately.

Fault Tree Analysis (FTA) is a popular technique for analysing how faults or combinations of them can cause the total failure, also known as the top-event, of a system or subsystem. Developed in the 1960s, fault trees can be analysed logically (qualitatively) or probabilistically (quantitatively). Traditionally, these analyses are mostly done with the Boolean AND and OR gates. The end products of the logical analysis are the Minimal Cut Sets (MCSs), which are combinations of basic faults necessary and sufficient to cause the occurrence of the top-event. The probabilistic analysis produces numbers representing the probability of the

W. Zamojski et al. (Eds.): *New Results in Dependability & Comput. Syst.*, AISC 224, pp. 141–151.
DOI: 10.1007/978-3-319-00945-2_13

top-event occurrence or the importance of the various terms in the MCSs in relation to their contribution to the top-event probability.

However, classical FTA is considered static [1-2], which is a significant disadvantage in modern dynamic systems: there are often occasions where accurately representing the failure behaviour of the system requires a more flexible and in-depth description than that provided by ordinary AND and OR gates. Not doing so can result in inaccurate evaluation of MCSs and estimation of the top-event probability [3–5]. As a result, FTA has seen various modifications to enable it to model and evaluate modern systems presenting dynamic behaviours. A popular and widely used solution is the Dynamic Fault Tree (DFT) [4]. DFTs make use of a pre-existing definition of Priority-AND (PAND) gates [6] and introduce other dynamic gates -- Spare, Functional Dependency (FDEP) and Sequential Enforcing (SEQ) -- to model and evaluate fault trees with dynamic features.

Apart from DFTs, FTA has seen other modifications. A recent modification of FTA is Pandora [5] [7–8] which analyses fault trees logically with three temporal gates – PAND, POR, and SAND. PAND stands for Priority-AND and it occurs if and only if an input event occurs strictly before another input event; inputs are arranged left-to-right with the leftmost occurring first. POR is for Priority-OR – which represents the situation where an output event occurs if its first input event occurs before its second input event or just the first input event occurs without the occurrence of the second input event. Finally, SAND stands for Simultaneous-AND. A SAND gate is used to represent the situation where all input events to an output event occur at the same time.

Pandora analyses fault trees with its temporal gates by use of its novel temporal laws [5] to generate Minimal Cut Sequences (MCSQs), analogous to Minimal Cut Sets; this enables a form of temporal qualitative analysis. MCSQs represent combinations or sequences that are sufficient and necessary to cause the top event. There are also techniques for the quantitative evaluation of the AND and OR gates [9], PAND gate [6] [10–12] and POR [13]. Taking an ideal situation, the SAND gate evaluates to zero [14] because the probability of exponentially distributed, independent events occurring simultaneously is zero.

One situation which is not covered by either DFTs or Pandora is that of "near simultaneity". In some scenarios we may wish to differentiate between a failure that occurs because two (or more) events occur within a given time of each other, and a different failure that occurs when those events occur further apart. This kind of 'interval' occurrence is comparatively common. For example, if a fire is detected and the sprinklers activate almost immediately to extinguish it, then the damage may be comparatively minimal; conversely, if there is a delay between the alarm and the sprinklers, perhaps because of a blockage in a pipe, the damage may be significantly worse.

Approaches to formalise such scenarios include Duration Calculus [15-16], PLTLP [17], CSDM [18], CTL [19], simplified CTL* [20] and PFTTD [21]. In general, these approaches are intended to formalise the semantics of timing and sequences of events in the context of simulation and formal specification. Only CSDM and Durational Calculus are intended to work with fault trees, and even this is primarily via an initial transformation to other forms, e.g. Petri Nets.

By contrast, DFTs and the Pandora-based approach in this paper are both fault tree-centric approaches designed with analysis, rather than specification, as the primary goal. However, DFTs in general lack good support for qualitative analysis, so we focus here on extending Pandora's existing capabilities to solve this issue.

Pandora's SAND gate provides the closest semantics to this scenario, but needs extending so that it can model a slight time delay between the input events. Therefore this work seeks define a delay-inclusive SAND gate, which is hereafter referred to as the parameterized SAND (pSAND), and to define means of probabilistically evaluating such a gate. The new gate is defined in section 2, and in section 3 we present two new mathematical techniques for evaluating the pSAND gate: Calculus (exact solution) and Monte Carlo Simulation (approximate solution). Both techniques are applied to a small case study in section 4. Discussions of the results and evaluation of proposed techniques are made in sections 5 and in section 6 we present our conclusions.

2 Parameterized SAND (pSAND) Description

Nearly simultaneous events are events that will trigger the occurrence of an output event if they should happen within a relatively short period of time – i.e., within a given *interval*. A classic example is discussed in the case study in Section 4. To model and evaluate the pSAND gate where input events occur within an interval, it is expedient that its symbols for modelling, qualitative and quantitative analysis be altered to accommodate the change. Therefore the original SAND gate is not redefined entirely, but rather slightly extended:

> *Semantics of pSAND*: All input events of the pSAND gate must occur and they must do so within a relatively short interval of duration '*d*', which starts with the first input event to occur. The pSAND is therefore false if any of its inputs do not occur or if they occur outside the interval, i.e., the time between the first and last input to occur is more than *d*.

Since it is not the focus of this work to completely redefine the SAND gate, its original graphical symbol (Fig. 1a) is retained for the situation where $d=0$: input events occur at exactly the same time. Alternatively, Fig. 1b can also be used to represent the same scenario where $d=0$. Fig. 1c on the contrary represents a situation where the input events are nearly simultaneous with a duration, d, between them and $d>0$.

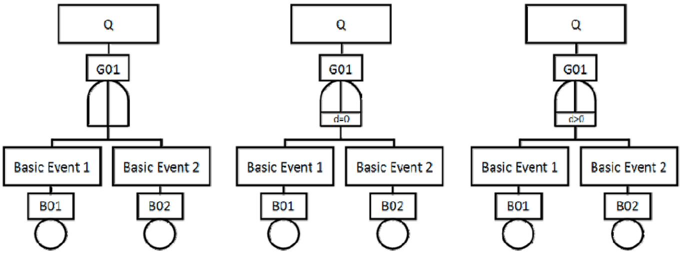

Fig. 1 pSAND graphical representations

SAND's abbreviation and symbol are changed to pSAND and $\&_d$ respectively, where d will be the duration of the interval. In addition, expressing $A\ \&_d\ B$ will be represented as $A\ \&_d\ B\ \{t_0,\ t_1\}$ where $d>0$ and $d=t_1 - t_0$; t_0 being the time of beginning the duration and t_1 the time of end of the interval within which the output event becomes true.

For any two independent events A and B, Fig. 2 represents the timing behaviour for $A\ \&_d\ B$ and equations a-e their corresponding mathematical expressions.

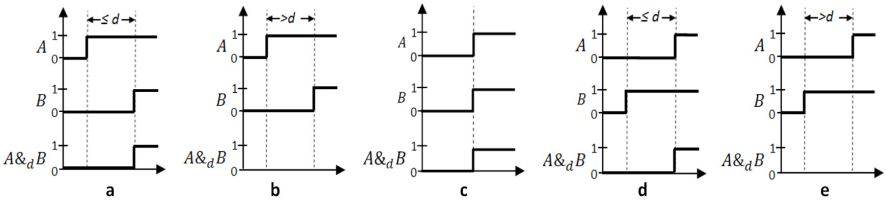

Fig. 2 pSAND timing behaviour

$$\{t(A) < t(B)\}\ \text{AND}\ \{t(B) - t(A) \leq d\} \longrightarrow t(A\ \&_d\ B) = t(B) \tag{a}$$

$$\{t(A) < t(B)\}\ \text{AND}\ \{t(B) - t(A) > d\} \longrightarrow t(A\ \&_d\ B) = \emptyset \tag{b}$$

$$\{t(A) = t(B)\} \longrightarrow t(A\ \&_d\ B) = t(A) \tag{c}$$

$$\{t(A) > t(B)\}\ \text{AND}\ \{t(A) - t(B) \leq d\} \longrightarrow t(A\ \&_d\ B) = t(A) \tag{d}$$

$$\{t(A) > t(B)\}\ \text{AND}\ \{t(A) - t(B) > d\} \longrightarrow t(A\ \&_d\ B) = \emptyset \tag{e}$$

In Fig. 2a A occurs before B and the duration of delay between them is less than or equal to d but greater than zero: thus $A\ \&_d\ B$ occurs/becomes true. In Fig. 2b A occurs before B within the duration which is greater than d and d is greater than zero: $A\ \&_d\ B$ does not occur. In Fig. 2c, if A and B occurred at exactly the same time then the pSAND would still be true; however, the probability of this occurring with independent events is essentially 0 (though if A and B were not independent, the probability could be non-zero). In Fig. 2d B occurs before A within the duration which is less or equal to d but greater than zero: $A\ \&_d\ B$ occurs. In Fig. 2e

B occurs before *A* with a duration of delay which is greater than *d* and greater than zero: $A \&_d B$ does not occur. pSAND occurs when all of its input events occur within a specified interval of duration.

3 Mathematical Model

It is assumed that all events are non-repairable, exponentially distributed with constant failure rates and any system under study is coherent with *F(X)* being the cumulative distributive function of *X*. Fig. 3 is a graphical representation of a pSAND scenario with two input events E_1 and E_2 having constant failure rates λ_1 and λ_2 respectively and a delay '*d*' between the occurrence of E_1 and E_2; E_1 occurring at 't_0' and E_2 occurring anytime between 't_0' and 't_1'.

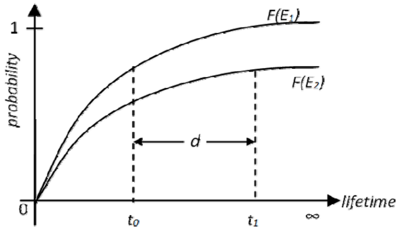

Fig. 3 pSAND Mathematical graph for two events

The probability of E_1 occurring at t_0 and E_2 occurring anytime between t_0 and t_1 is the pSAND probability of E_1 and E_2. pSAND is commutative therefore event sequence does not matter. Algebraically this can be expressed as:

$$E_1 \&_d E_2 \{t_0,t_1\}=E_2 \&_d E_1 \{t_0,t_1\}=E_1 \{0,t_0\}*E_2 \{t_0,t_1\}+E_2 \{0,t_0\}*E_1 \{t_0,t_1\} \quad (1)$$

$$Pr(E_1 \&_d E_2)\{t_0,t_1\}=Pr(E_1 \{0,t_0\}*E_2 \{t_0,t_1\}+E_2 \{0,t_0\}*E_1 \{t_0,t_1\}) \quad (2)$$

$$=Pr(E_1 \{0,t_0\}*E_2 \{t_0,t_1\})+Pr(E_2 \{0,t_0\}*E_1 \{t_0,t_1\})-Pr((E_1 \{0,t_0\}*E_2 \{t_0,t_1\})*(E_2 \{0,t_0\}*E_1 \{t_0,t_1\})) \quad (3)$$

However, $Pr((E_1 \{0,t_0\}*E_2 \{t_0,t_1\})*(E_2 \{0,t_0\}*E_1 \{t_0,t_1\}))$ equals 0, assuming E1 and E2 are independent and a continuous model of time is used, therefore,

$$Pr(E_1 \&_d E_2)\{t_0,t_1\}=Pr(E_1 \{0,t_0\}*E_2 \{t_0,t_1\})+Pr(E_2 \{0,t_0\}*E_1 \{t_0,t_1\}) \quad (4)$$

$$=Pr(E_1 \{0,t_0\})*Pr(E_2 \{t_0,t_1\})+Pr(E_2 \{0,t_0\})*Pr(E_1 \{t_0,t_1\}) \quad (5)$$

$$=Pr(E_1 \{0,t_0\})*Pr(E_2 \{t_0,t_1\})+Pr(E_2 \{0,t_0\})*Pr(E_1 \{t_0,t_1\}) \quad (6)$$

$Pr(X\{a,b\})=F(X)\{a,b\}=Exp(-a*x) - Exp(-b*x)$, *F(X)* being the Cumulative Distributive Function (CDF) of *X*. Therefore,

$$Pr(E_1 \&_d E_2)\{t_0,t_1\}=F(E_1)\{0,t_0\}*(F(E_2)\{t_0,t_1\})+F(E_2)\{0,t_0\}*(F(E_1)\{t_0,t_1\}) \quad (7)$$

Given that, $Pr(X\{t_0,t_1\})=Pr(X\{0,t_1\})-Pr(X\{0,t_0\})$

$$Pr(E_1 \&_d E_2)\{t_0,t_1\}=F(E_1)\{0,t_0\}*(F(E_2)\{0,t_1\}-F(E_2)\{0,t_0\})+ F(E_2)\{0,t_0\}*(F(E_1)\{0,t_1\}-F(E_1)\{0,t_0\}) \quad (8)$$

It follows that for any n independent events $E_1, E_2, \ldots, E_{n-1}, E_n$ the pSAND probability is

$$Pr(E_1 \&_d E_2 \&_d \ldots \&_d E_{n-1} \&_d E_n)\{0,t_0,t_1\} = \sum_{i=1}^{n}\left(F(E_i)\{0,t_0\} * \left(\prod_{\substack{j=1 \\ j\neq i}}^{n} F(E_i)\{t_0,t_1\}\right)\right) \quad (9)$$

Monte Carlo (MC) simulation is used to understand and control complex stochastic real world systems. It has been employed in weather forecasting, insurance, engineering, financial market, chemical processes, telecommunication networks and the like. Its popularity in reliability engineering has increased over the past decade. It has been employed in qualitative [22–23] and quantitative [10] analysis.

A typical MC implementation commences by mathematically modelling known and unknown variables of the system under study. This model is run a large number of times – called trials – based on the outputs expected from the model with randomised values of appropriate input variables. This simulated random behaviour of the system model allows the estimation of complex real world probabilities provided it has been well modelled and randomised an appreciable number of times.

An algorithm to estimate the pSAND probability for n independent events using Monte Carlo simulation is:

1. Generate random numbers for the failure rates of all events.
2. If, for all events, the first event occurs between $\{0,t_0\}$ while all other events occur between $\{t_0,t_1\}$ then keep count
3. Repeat steps 1 and 2 for a large number of trials.
4. Evaluate the pSAND probability by dividing the number of counts by the trials.

4 Case Study

The pSAND is useful in differentiating the effects of failures that occur within a small interval from effects of more widely spaced failures. To demonstrate the relevance of pSAND and prove the proposed techniques, in Fig. 4 we present an automotive brake-by-wire (BBW) system, which is adapted from [8].

The BBW system contains individual brake actuators with rotation sensors connected at each of its four wheels. The central bus serves two major functions: it is a medium for controlling the actuators and also carries the signals from the sensors. The signals from the sensors are inputs to a pair of Electronic Control Units (ECUs) that control the brakes. To prevent inadvertent braking caused by an error in one ECU, the comparator determines the output of both ECUs; if they agree, commands are sent to the actuators to activate braking. 'Vehicle dynamics' is a virtual component representing the effect the brakes have on the handling of the vehicle and can be thought of as the output of the braking system. It should be emphasized that the vehicle dynamics is not a physical component of the system but rather a way of representing the success (or failure) of the braking effect on the vehicle. For the sake of demonstrating the pSAND concept, the case study presented in this work is simplified. The focus is solely on the vehicular dynamics based only on the front wheels braking.

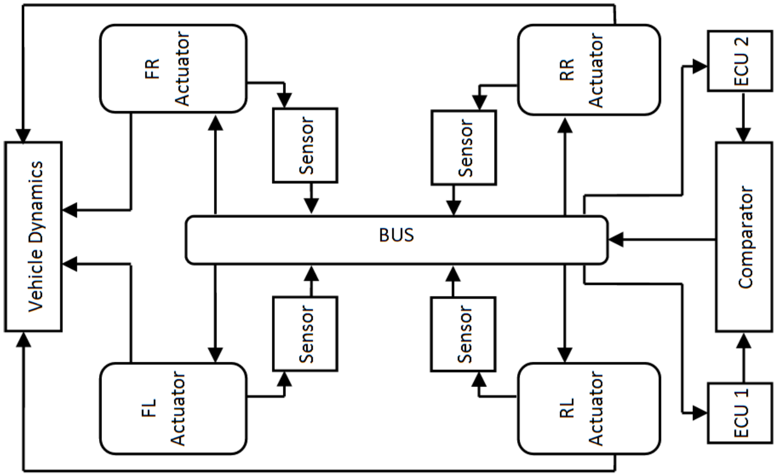

Fig. 4 An automotive brake-by-wire system

The failure behaviour of the actuators coupled with its failure data are modelled below. ‘IF’ represents the Internal Failure of a component, 'V' represents a value deviation (i.e., an error in a signal), and ‘C’ represents inadvertent commission of the component. {X} means that a failure can refer to any of FR (front-right), FL (front-left), RL (rear-left), or RR (rear-right), e.g. C_ActuatorFL is the front-left braking actuator. “+”, “.” and “|” are for logical OR, AND and POR respectively.

C_Actuator{X}	= IF_Actuator{X} + C_BusCommand{X}
C_BusCommand{X}	= C_Comparator{X} + IF_Bus
C_Comparator{X}	= C_ECU1{X} . C_ECU2{X} + IF_Comparator
C_ECU{X}	= V_BusSignal{X} + IF_ECU
V_BusSignal{X}	= V_SensorData{X}
V_SensorData{X}	= IF_Sensor{X}

A thorough qualitative analysis on the entire BBW system produces approximately 75 MCSQs. A full quantification of the entire MCSQs is outside the scope of this paper. However, to demonstrate the nearly simultaneous scenario explained in this work we focus on the effects of inadvertent commission of the front right and left actuators.

The vehicle dynamics section provides the overall failures of the system. In this work, we focus only on the front two wheels, in which case we have three possible scenarios:

- The left wheel brakes either alone or before the right wheel, causing the vehicle to veer suddenly to the left.
- The right wheel brakes either alone or before the left wheel, causing the vehicle to veer to the right.
- Both wheels brake at approximately the same time.

How critical these failures are depends on whether the vehicle is in a left-hand or right-hand drive country, but for the sake of the example let us assume it is left-hand drive (i.e., vehicles drive on the left). In this case, veering to the right may cause the vehicle to veer into oncoming traffic, which is the most severe possible failure. Veering to the left may cause the vehicle to go off road, which is still dangerous but less so than a head-on collision with another vehicle. Finally, if both wheels brake at the same time, the car is likely to brake in a roughly straight line, which is the least severe of the three types of failure.

These failures can be represented using temporal gates like so:

VeerIntoOncomingTraffic = C_Actuator_FR | C_Actuator_FL
VeerOffRoad = C_Actuator_FL | C_Actuator_FR
StraightBraking = C_Actuator_FL $\&_d$ C_Actuator_FR

Note that here we use the pSAND for the straight-line braking; as long as the brakes fail within an interval of about 0.1 seconds, the result will be the less severe straight braking rather than a dangerous veer to the side. The actuators need not fail at exactly the same moment.

Table 1 Failure Probabilities

Component	Failure rate/hr
C_Actuator_FL	1E-3
C_Actuator_FR	1E-3

Table 2 Results of StraightBraking (*C_Actuator_FL* $\&_{0.1}$ *C_Actuator_FR)*

Time (hrs)	Analytical Solution	Monte Carlo Solution	Percentage Error
1E1	5.0194E-5	4.9000E-5	2.37
1E2	4.3873E-4	4.2600E-4	2.90
1E3	1.1849E-3	1.2110E-3	2.21

5 Discussion and Evaluation

Table 2 contains results of the application of both analytical and Monte Carlo solution described in this work on the StraightBraking failure described in the case study. Results for both techniques were achieved by modelling them in Mathematica 8; an application for performing many kinds of computations [24]. By using small realistic failure data with no dynamic stopping techniques or importance measures, we use a large number of trials (10^6) to increase the accuracy of results. MC simulation is also simplified in this work. It is used just for the purpose of demonstration.

From Table 2, it is clear that both techniques produce results close to each other at least to the magnitude. The percentage error is less than 3% which can be considered to be good considering the fact that we have not used a dynamic stopping technique on the Monte Carlo approximation technique.

In general, the results show that the probability of both C_Actuator_FR and C_Actuator_FL increases with time. Meaning the more time given, the more likely it is for the two events to occur within d.

From equation (9), it can be seen that if d (which is equal to $t_1 - t_0$) is zero, the pSAND probability is also zero which means it technically becomes a SAND.

Although we have provided an analytical solution in this work, we have also provided a simulative alternative. This is not to necessarily evaluate the analytical solution but to provide a framework for estimating the pSAND probability of events with distributions other than an exponential distribution.

For future work, one may consider the possibility of parameterising the PAND and POR gates to achieve some form of completion [5] [9].

6 Conclusion

In modern dynamic systems, it is often necessary to be able to represent scenarios involving events which are "nearly simultaneous", i.e., they occur together within a short period of time. It is common for the effects of such failures to be different if they occur nearly simultaneously compared to occurring further apart, as in the braking system presented earlier. However, it is difficult to model this situation in current FTA approaches. Pandora, a modification to FTA, provides the SAND gate to represent simultaneous occurrence of events, but does not cater for any delay between occurrences. This paper extends Pandora’s definition of SAND gates to provide "parameterised SAND" gates or pSANDs, which can be used in modelling and evaluating nearly simultaneous scenarios. We have proposed new logical representations for pSAND and provided its exact and approximated calculations using analytical and Monte Carlo solutions respectively. The result is a more flexible analytical approach that enables Pandora-based FTA to be applied to a greater variety of situations in modern safety critical systems.

References

[1] Gulati, R., Dugan, J.B.: A modular approach for analyzing static and dynamic fault trees. In: Reliability and Maintainability Symposium (1997)

[2] Merle, G., Roussel, J.: Algebraic modelling of Fault Trees with Priority AND gates. In: 1st IFAC Workshop, pp. 175–180 (2007)

[3] Dugan, J.B., Doyle, S.A.: New results in fault-tree analysis. In: Tutorial Notes of the Annual Reliability and Maintainability Symposium (1997)

[4] Dugan, J.B., Bavuso, S.J., Boyd, M.A.: Dynamic fault-tree models for fault-tolerant computer systems. IEEE Transactions on Reliability 41(3), 363–377 (1992)

[5] Walker, M., Papadopoulos, Y.: Synthesis and analysis of temporal fault trees with PANDORA 2: The time of Priority AND gates. Nonlinear Analysis: Hybrid Systems 2(2), 368–382 (2008)

[6] Fussell, J.B., Aber, E.F., Rahl, R.G.: On the quantitative analysis of Priority-AND failure logic. IEEE Transactions on Reliability R-25(5), 324–326 (1976)

[7] Walker, M., Papadopoulos, Y.: Pandora 2: The time of priority-OR gates. In: IFAC Workshop on Dependable Control of Discrete Event Systems (2007)

[8] Walker, M.: Pandora: a logic for the qualitative analysis of temporal fault trees. Dissertation, University of Hull (2009)

[9] Vesely, W.E., Stamatelatos, M., et al.: Fault tree handbook with aerospace applications. NASA Office of Safety and Mission Assurance, Washington DC (2002)

[10] Durga, R., Gopika, V., et al.: Dynamic fault tree analysis using Monte Carlo simulation in probabilistic safety assessment. Reliability Engineering & System Safety 94(4), 872–883 (2009)

[11] Yuge, T., Yanagi, S.: Quantitative analysis of a fault tree with Priority AND gates. Reliability Engineering & System Safety 93(11), 1577–1583 (2008)

[12] Long, W., Sato, Y., Horigome, M.: Quantification of sequential failure logic for fault tree analysis. Reliability Engineering & System Safety 67(3), 269–274 (2000)

[13] Edifor, E., Walker, M., Gordon, N.: Quantification of priority-OR gates in temporal fault trees. In: Ortmeier, F., Lipaczewski, M. (eds.) SAFECOMP 2012. LNCS, vol. 7612, pp. 99–110. Springer, Heidelberg (2012)

[14] Merle, G., Roussel, J.: Probabilistic algebraic analysis of fault trees with priority dynamic gates and repeated events. IEEE Transactions on Reliability 59(1), 250–261 (2010)

[15] Chaochen, Z., Hoare, C., Ravn, A.: A calculus of Durations. Information Processing Letters 40(5), 269–276 (1991)

[16] Hansen, K.M., Anders, P.R., Stavridou, V.: From safety analysis to software requirements. IEEE Transactions on Software Engineering 24(7), 573–584 (1998)

[17] Palshikar, G.: Temporal fault trees. Information and Software Technology 44(3), 137–150 (2002)

[18] Gorski, J., Wardzinski, A.: Deriving real-time requirements for software from safety analysis. In: Real-Time Systems, pp. 9–14 (1996)

[19] Schellhorn, G., Thums, A., Reif, W.: Formal fault tree semantics. In: Proceedings of The Sixth World Conference on Integrated Design & Process Technology (2002)

[20] Güdemann, M., Ortmeier, F., Reif, W.: Computation of Ordered Minimal Critical Sets. In: Proceedings of the 7th Symposium on Formal Methods for Automation and Safety in Railway and Automotives (2008)

[21] Babczyński, T., Lukowicz, M., Magott, J.: Time coordination of distance protections using probabilistic fault trees with time dependencies. IEEE Transactions on Power Delivery 25(3), 1402–1409 (2010)
[22] Rocco, C., Muselli, M.: A machine learning algorithm to estimate minimal cut and path sets from a Monte Carlo simulation. In: Probabilistic Safety Assessment and Management (PSAM7–ESREL), pp. 3142–3147 (2004)
[23] Chan, J.C., Kroese, D.P.: Rare-event probability estimation with conditional Monte Carlo. Annals of Operations Research 189(1), 43–61 (2009)
[24] Wolfram Research, What Is Mathematica? (2013), `http://www.wolfram.co.uk/mathematica` (accessed January 07, 2013)