Dynamic Fault Tree Analysis Based On The Structure Function

Guillaume Merle, Jean-Marc Roussel, Jean-Jacques Lesage

Dynamic Fault Tree Analysis Based On The Structure Function

Guillaume Merle, PhD, LURPA, ENS Cachan
Jean-Marc Roussel, PhD, LURPA, ENS Cachan
Jean-Jacques Lesage, Habil., PhD, LURPA, ENS Cachan

Key Words: Boolean function, fault trees, probability, reliability, symbol manipulation

SUMMARY & CONCLUSIONS

This paper presents an algebraic approach allowing to perform the analysis of any Dynamic Fault Tree (DFT). This approach is based on the ability to formally express the structure function of DFTs. We first present the algebraic framework that we introduced to model dynamic gates and hence be able to determine the structure function of DFTs. Then, we show that this structure function can be rewritten under a canonical form from which the qualitative analysis of DFTs can be performed directly. We finally provide a probabilistic model of dynamic gates to be able to perform the quantitative analysis of DFTs from their structure function.

1 INTRODUCTION

Fault Tree Analysis (FTA) is one of the oldest, most diffused techniques in industrial applications, for the dependability analysis of critical systems [1-3]. When the interactions between events can be described by means of Boolean OR/AND gates only, so that only the combination of events is relevant, and not their sequence, Fault Trees are called *Static Fault Trees* (SFT). SFTs are commonly analyzed directly from their *structure function*, which is the logical expression between the top event and the basic events of the SFT. The *qualitative analysis* consists in determining the minimal cut sets – the minimal combinations of events which are sufficient to engender the top event – of the SFT. The *quantitative analysis* consists in computing the failure probability of the top event of the SFT. Dugan et al. [4-5] proposed a new model allowing to include various kinds of temporal and statistical dependencies in the SFT model, which is the *Dynamic Fault Tree* (DFT). The DFT is based on the definition of gates Priority-AND (PAND), Functional Dependency (FDEP), and Spare.

Even though the semantics of dynamic gates allows to model complex failure mechanisms that SFTs cannot take into account, DFTs cannot be analyzed by using regular approaches as their structure function cannot be determined. Other types of approaches are hence used to perform the analysis of DFTs. These approaches are mainly based on Zero-suppressed Binary Decision Diagrams (ZBDD) [6], Continuous Time Markov Chains (CTMC) [7], Stochastic Petri Nets (SPN) [8], and dynamic Bayesian Networks (BN) [9]. However, these approaches have limits in the analyses that they allow as well as in the distributions that can be taken into account, even though any distribution can, in most cases, be accommodated by numerical simulation.

In a previous article [10], the authors proposed to extend the approaches commonly used to analyze SFTs to DFTs. We hence proposed an algebraic framework allowing to determine the structure function of DFTs including dynamic gates PAND and FDEP, as well as an analytical approach allowing to perform the analyses from this structure function.

In this paper, we propose to extend the approach considered in [10] to the case of Spare gates. The main approaches allowing to analyze DFTs and their respective limits are presented in Section 2. The algebraic framework allowing to determine the structure function of DFTs is recalled in Section 3, and the behavioural and probabilistic models of dynamic gates are respectively presented in Sections 4 and 5. Finally, we illustrate our approach on a DFT example from the literature in Section 6.

2 STATE OF THE ART

Many approaches have been envisaged to analyze DFTs. In [6], each dynamic gate of the considered DFT is replaced by the static gate corresponding to its logic constraints; the minimal cut sets of the resulting SFT are then generated by using ZBDDs, and these minimal cut sets are expanded to minimal cut sequences by considering the timing constraints. The authors of [7] propose to convert the DFT into a *failure automaton* which models the changing state of the system as failures occur. This failure automaton can then be converted into a CTMC, and the solution of the corresponding set of differential equations allows to determine the failure probability of the top event of the DFT. These two approaches have been implemented in Galileo [5]. Other model-based approaches can be used to perform the quantitative analysis of DFTs. For instance, in [9], the whole DFT is converted into a dynamic BN and the failure probability of the top event of the DFT can be determined by using inference algorithms. In [8], the dynamic subtrees of DFTs are converted into a class of coloured SPNs called *Stochastic Well-formed Net* (SWN). This SWN can be converted into a CTMC to determine the failure probability of the top event of the dynamic subtree, and this failure probability can then be cast back into the original DFT. These

two approaches have been respectively implemented in the Windows [9] and Linux [11] version of Drawnet.

These approaches, as well as the numerous ones which have not been cited in this section, are more of less efficient, but they provide literal quantitative results for exponential distributions only, even though numerical simulation still allows to accommodate any distribution. Because of this limit, we chose to propose an extension of the analysis approaches used for SFTs and based on the structure function. The algebraic framework that we introduced to determine the structure function of DFTs is recalled in Section 3.

3 ALGEBRAIC FRAMEWORK FOR THE MODELLING OF DYNAMIC FAULT TREES

3.1 *Hypotheses*

The hypotheses considered in this work are as follows:

- the DFTs that we consider are the DFTs defined in [4], which include static gates (OR, AND, and K-out-of-N) and dynamic gates (PAND, FDEP, and Spare);
- events are not repairable, in accordance with [3];
- basic events have continuous failure time distributions, as considered in [12], so that independent basic events cannot occur simultaneously; and
- intermediate events of a DFT can still occur simultaneously if the DFT contains repeated events, as explained in [10].

3.2 *Basics and notations of our algebraic framework*

The Boolean model commonly used to model events and gates in SFTs does not allow to take into account the order of appearance of events which is needed to model dynamic gates.

To be able to take into account this temporal aspect and hence model sequences of events, we consider events as Boolean functions defined on the set of positive times and which take Boolean values. As we consider non-repairable events, each non-repairable event *a* can be assigned a unique date of appearance *d(a)*. The timing diagram of a non-repairable event *a* is shown in Figure 1.

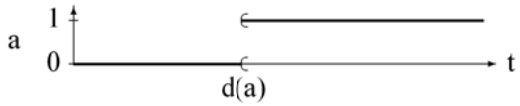


Figure 1 – A non-repairable event

In addition to operators OR (+) and AND (·), we have defined three temporal operators on the set of non-repairable events E_{nr} to model dynamic gates. These operators are operators non-inclusive BEFORE ($\lhd$), SIMULTANEOUS ($\triangle$), and Inclusive BEFORE ($\trianglelefteq$). The definition of these three operators can be found in [10] and is based on the date of appearance of their operands, as illustrated by the definition of the temporal operator Inclusive BEFORE:

$$d(a \trianglelefteq b) = \begin{cases} d(a) \text{ if } d(a) < d(b) \\ +\infty \text{ if } d(a) > d(b) \\ d(a) \text{ if } d(a) = d(b) \end{cases} \quad (1)$$

The exhaustive list of all the theorems verified by these three operators can be found in [13].

4 BEHAVIOURAL MODEL OF DYNAMIC GATES

The three temporal operators defined in Section 3.2 allow to determine the behavioural model of dynamic gates. The behavioural models of gates PAND and FDEP can be found in [10] and are illustrated in Table 1.

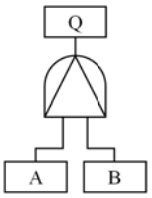
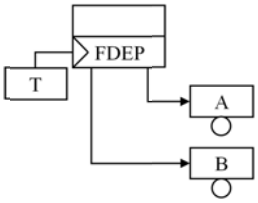
Symbol	Behavioural model
	$Q = A \cdot B \cdot (A \trianglelefteq B)$ $= B \cdot (A \trianglelefteq B)$
	$A_T = (A \trianglelefteq T) + T = A + T$ $B_T = (B \trianglelefteq T) + T = B + T$

Table 1 – Behavioural models of gates PAND and FDEP

Regarding Spare gates, according to [3], Spare gates can be Cold (CSP), Warm (WSP), or Hot (HSP). In this paper, we consider that CSP and HSP gates are specific cases of WSP gates. We hence respectively present the behavioural model of single WSP gates and multiple WSP gates sharing a spare event in Sections 4.1 and 4.2, before presenting the changes that must be made in the behavioural model to take into account the specific case of CSP and HSP gates in Section 4.3. Finally, we show in Section 4.4 how this behavioural model of dynamic gates allows to perform the qualitative analysis of DFTs directly.

4.1 *Behavioural model of a Spare gate with 2 input events*

Let us consider a Spare gate with 2 input events – the primary event A and one spare event B – as shown in Figure 2.

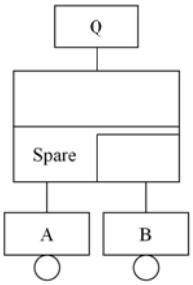


Figure 2 – A single Spare gate with 2 input events

As stated in [3], the output Q of the gate occurs when the primary and all spares have failed, so when A and B have

failed, in this case. A and B are basic events and cannot fail simultaneously, so Q occurs if A and B fail according to sequences $[A,B]$ or $[B,A]$. It is important to note that in sequence $[A,B]$, B fails while in its active mode (denoted by B_a), whereas in sequence $[B,A]$, B fails while in its dormant mode (denoted by B_d). It is essential to distinguish both failure modes by using two different variables, for quantitative analysis purposes. Indeed, B does not have the same failure distribution when it fails during its dormant mode ($B \equiv B_d$) or during its active mode ($B \equiv B_a$). As we aim at making possible the quantitative analysis of DFTs from their structure function, this structure function must hence provide sufficient information to know whether spare events are in their dormant or active mode. Finally, the behavioural model of gate Spare can hence be expressed as

$$Q = B_a \cdot (A \lhd B_a) + A \cdot (B_d \lhd A) \quad (2)$$

Furthermore, as B cannot be both in an active state and in a dormant state, we have

$$B_d \cdot B_a = \perp \quad (3)$$

where $\perp$ is the never-occurring event which corresponds to the additive identity of the set of non-repairable events.

4.2 *Behavioural model of 2 Spare gates with 2 input events sharing a spare event*

Let us consider the specific case of 2 Spare gates – with primary events A and B – sharing a spare event C, as shown in Figure 3.

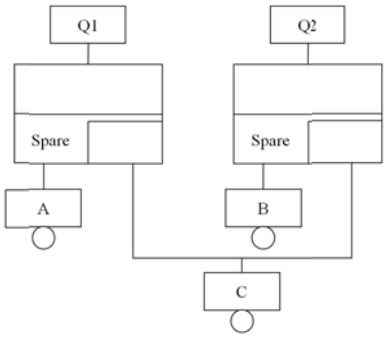


Figure 3 – Two Spare gates sharing a spare event

If we focus on the Spare gate on the left side, Q1 occurs as soon as A and C have failed – as stated in Section 4.1 – or if A fails and C is made unavailable because B has failed before A. As a consequence, the behavioural model of the Spare gate on the left side is

$$Q1 = C_a \cdot (A \lhd C_a) + A \cdot (C_d \lhd A) + A \cdot (B \lhd A) \quad (4)$$

The algebraic expression for the Spare gate on the right side can be determined in the same way by symmetry:

$$Q2 = C_a \cdot (B \lhd C_a) + B \cdot (C_d \lhd B) + B \cdot (A \lhd B) \quad (5)$$

The behavioural model of Spare gates in the general case of *n* Spare gates sharing a Spare event can be found in [13].

4.3 *Specific case of Cold and Hot Spare gates*

The behavioural models presented in Sections 4.1 and 4.2 can be simplified in the specific cases of Cold and Hot Spare events:

- In the case of a cold spare event, the behavioural model in (2) becomes
$$Q = B_a \cdot (A \lhd B_a), \quad (6)$$
as B cannot fail while in its dormant mode.
- In the case of a hot spare event, the behavioural model in (2) becomes
$$Q = B \cdot (A \lhd B) + A \cdot (B \lhd A) = A \cdot B, \quad (7)$$
as B has the same distribution function in its active and dormant mode ($B_a \equiv B_d \equiv B$).

4.4 *Qualitative analysis of DFTs*

This behavioural model of dynamic gates allows to determine the structure function of any DFT. The theorems verified by temporal operators allow to reduce it to a canonical form under the form

$$TE = \sum\left(\prod b_i \prod\left(b_j \lhd b_k\right)\right), j \notin \{i,k\} \quad (8)$$

where TE is the top event of the DFT and b_x are the basic events of the DFT, which may be spare events in their active or dormant state.

Even if it allows to perform the qualitative analysis of DFTs directly, this canonical form of the structure function is not sufficient to perform their quantitative analysis for which a probabilistic model of dynamic gates is necessary. This probabilistic model is presented in Section 5.

5 *PROBABILISTIC MODEL OF DYNAMIC GATES*

The probabilistic models of gates PAND and FDEP can be found in [10], so we only consider the case of Spare gates in this section. Let us first recall the probabilistic formulas which allowed to determine the probabilistic model of gates PAND and FDEP, and which will be needed to determine the probabilistic model of Spare gates. Given an event *x* with cumulative distribution function (Cdf) $F_x(t)$, and probability density function (pdf) $f_x(t)$, the following expressions hold under the hypothesis of statistical independence:

$$Pr\{A \lhd B\}(t) = \int_0^t f_A(u)\left(1 - F_B(u)\right)du \quad (9)$$

$$Pr\{B \cdot (A \lhd B)\}(t) = \int_0^t f_B(u) F_A(u) du \quad (10)$$

5.1 *Probabilistic model of a Spare gate with 2 input events*

The behavioural model of a single Spare gate with 2 input events is given in (2). B cannot fail both before and after A, so both algebraic terms are disjoint and

$$Pr\{Q\}(t) = Pr\{B_a \cdot (A \lhd B_a)\}(t) + Pr\{A \cdot (B_d \lhd A)\}(t) \quad (11)$$

On the one hand, the Cdf and pdf of B_d do not depend on A, so the probability of occurrence of the second term can be determined by means of the expression (10) as

$$Pr\{A \cdot (B_d \lhd A)\}(t) = \int_0^t f_A(u) F_{B_d}(u) du \quad (12)$$

On the other hand, the Cdf and pdf of B_a depend on the failure date of A, so the probability of occurrence of the first term cannot be determined by means of the expression (10) as A and B_a are statistically dependent. The rewriting of this probability by means of *expectation values* and *indicator functions* [14] allows to determine the following expression for $Pr\{B_a \cdot (A \lhd B_a)\}(t)$, as detailed in [13]:

$$Pr\{B_a \cdot (A \lhd B_a)\}(t) = \int_0^t \left(\int_v^t f_{B_a}(u,v)du\right) f_A(v)dv \quad (13)$$

The probabilistic model of a single Spare gate with 2

input events can hence be obtained by summing expressions (12) and (13) according to (11). It can be noted that this probabilistic model does not depend on the distribution considered for basic events.

5.2 *Quantitative analysis of DFTs*

The inclusion-exclusion formula [15] and the probabilistic model of dynamic gates presented in [10] and in Section 5.1 allow to perform the quantitative analysis of any DFT. It can be noted that both the probabilistic model of dynamic gates and the expression for the failure probability of the top event which will be obtained from this probabilistic model can accommodate any distribution for basic events.

We illustrate this approach on a DFT example from the literature in Section 6.

6 *APPLICATION TO A DFT EXAMPLE*

6.1 *A Computer System Example (HECS)*

We are going to illustrate our approach on the DFT of an Hypothetical Example Computer System (HECS) from [3] which is shown in Figure 4.

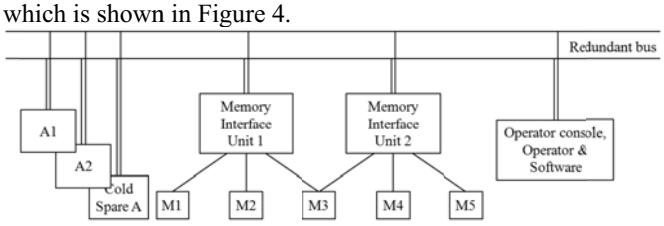


Figure 4 – The Computer System Example

The HECS includes dual-redundant processors A1 and A2 and a cold spare processor A, which can replace either upon failure. Processors A1, A2, and A are all identical processors, running the same operating system. The system can continue to operate until all three processors have failed.

The HECS also includes five memory units of which three are required. These memory units are connected to the redundant bus via two memory interface units. If a memory interface unit fails, the memory units connected to it are unusable. Memory unit 3 (M3) is connected to both interfaces for redundancy; thus M3 is accessible as long as either interface unit is operational. A memory interface unit must hence be operational in order for the memory units which are connected to it to be accessible, thus the memory units are functionally dependent on the interfaces.

There are two identical redundant buses (BUS1 and BUS2), of which one is required for system operation. Thus the bus subsystem fails when both of the buses fail.

The last subsystem to be considered is the application subsystem. The application software runs on the computer system. The operator is a human who interfaces with the computer via a Graphical User Interface (GUI) that runs on an interface device. Thus an application (software (SW)) failure, GUI (hardware (HW)) failure or human operator (OP) error will lead to system failure.

The HECS requires the correct operation of the processing, memory, and bus subsystems, as well as the software application. Thus the HECS will fail if any of these subsystems fail. The DFT which models the potential failure of the HECS is shown in Figure 5.

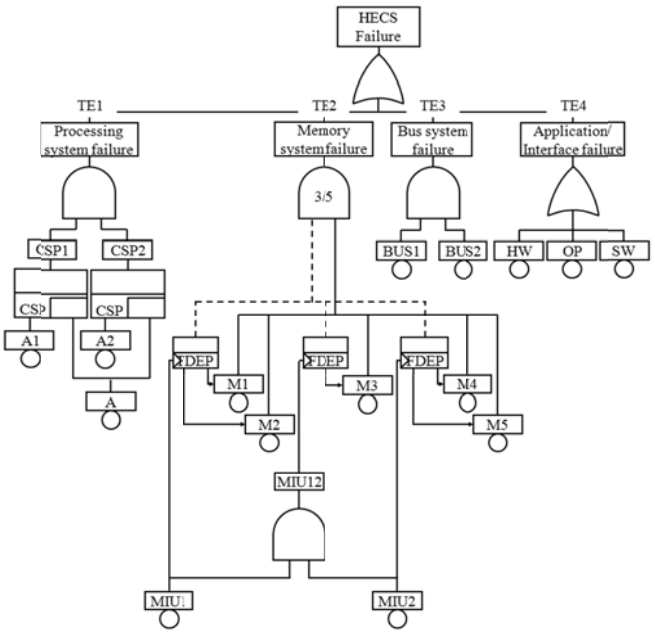


Figure 5 – The DFT of the HECS

6.2 *Canonical form of the structure function*

The DFT in Figure 5 can be divided into the 4 following independent subtrees:

- Subtree 1 corresponds to the processing system failure. It is dynamic and its top event will be denoted by TE_1.
- Subtree 2 corresponds to the memory system failure. It is dynamic and its top event will be denoted by TE_2.
- Subtree 3 corresponds to the bus system failure. It is static and its top event will be denoted by TE_3.
- Subtree 4 corresponds to the application/interface failure. It is static and its top event will be denoted by TE_4.

The structure function of the DFT of the HECS can hence be expressed as

$$TE = TE_1 + TE_2 + TE_3 + TE_4 \quad (14)$$

Subtrees 3 and 4 are static, so their structure function can be determined easily as

$$TE_3 = BUS1 \cdot BUS2 \quad (15)$$

$$TE_4 = HW + OP + SW \quad (16)$$

The structure function of subtree 2 can be obtained from the behavioural model of gate FDEP as

$$\begin{aligned} TE_2 = {} & MIU_1 \cdot MIU_2 + MIU_1 \cdot M_3 + MIU_1 \cdot M_4 \\ & + MIU_1 \cdot M_5 + MIU_2 \cdot M_1 + MIU_2 \cdot M_2 + MIU_2 \cdot M_3 \\ & + M_1 \cdot M_2 \cdot M_3 + M_1 \cdot M_2 \cdot M_4 + M_1 \cdot M_2 \cdot M_5 \\ & + M_1 \cdot M_3 \cdot M_4 + M_1 \cdot M_3 \cdot M_5 + M_1 \cdot M_4 \cdot M_5 \\ & + M_2 \cdot M_3 \cdot M_4 + M_2 \cdot M_3 \cdot M_5 + M_2 \cdot M_4 \cdot M_5 \\ & + M_3 \cdot M_4 \cdot M_5 \end{aligned} \quad (17)$$

Finally, the behavioural model of Spare gates presented in Section 4 allows to determine the structure function of subtree 1 as

$$\begin{aligned} TE_1 = {} & A_a \cdot A2 \cdot (A1 \lhd A_a) \cdot (A1 \lhd A2) \\ & + A_a \cdot A1 \cdot (A2 \lhd A_a) \cdot (A2 \lhd A1) \end{aligned} \quad (18)$$

The canonical form of the structure function of the HECS can hence be determined as the conjunction of these 4 expressions.

6.3 *Qualitative analysis*

The canonical form of the structure function of the HECS contains 23 (2 + 17 + 1 + 3) terms. On the one hand, 21 terms do not contain the temporal operator $\lhd$. They are static and can hence provide the minimal cut sets of the DFT:

$$\begin{gathered} MIU_1 \cdot MIU_2, MIU_1 \cdot M_3, MIU_1 \cdot M_4, MIU_1 \cdot M_5, \\ MIU_2 \cdot M_1, MIU_2 \cdot M_2, MIU_2 \cdot M_3, M_1 \cdot M_2 \cdot M_3, \\ M_1 \cdot M_2 \cdot M_4, M_1 \cdot M_2 \cdot M_5, M_1 \cdot M_3 \cdot M_4, M_1 \cdot M_3 \cdot M_5, \\ M_1 \cdot M_4 \cdot M_5, M_2 \cdot M_3 \cdot M_4, M_2 \cdot M_3 \cdot M_5, M_2 \cdot M_4 \cdot M_5, \\ M_3 \cdot M_4 \cdot M_5, BUS1 \cdot BUS2, HW, OP, SW \end{gathered} \quad (19)$$

On the other hand, 2 terms contain the temporal operator $\lhd$. They are dynamic and can hence provide the minimal cut sequences of the DFT. The algebraic term $A_a \cdot A2 \cdot (A1 \lhd A_a) \cdot (A1 \lhd A2)$ indicates that A1 must fail before A_a and A2 and hence corresponds to the two minimal cut sequences $[A1, A2, A_a]$ and $[A1, A_a, A2]$. The algebraic term $A_a \cdot A1 \cdot (A2 \lhd A_a) \cdot (A2 \lhd A1)$ indicates that A2 must fail before A_a and A1 and hence corresponds to the two minimal cut sequences $[A2, A1, A_a]$ and $[A2, A_a, A1]$. The minimal cut sequences of the DFT hence are

$$[A1, A2, A_a], [A1, A_a, A2], [A2, A1, A_a], [A2, A_a, A1] \quad (20)$$

6.4 *Quantitative analysis*

The probabilistic model of dynamic gates presented in Section 5 allows to determine the failure probability of the top event of the 4 subtrees considered in Section 6.2.

For instance, in the case of subtree 1, the structure function of the subtree is given in (18). We have shown in Section 6.3 that the two algebraic terms of (18) correspond to the four minimal cut sequences given in (20). The structure function for TE_1 can hence be rewritten as

$$\begin{aligned} TE_1 = {} & A_a \cdot (A1 \lhd A2) \cdot (A2 \lhd A_a) \\ & + A2 \cdot (A1 \lhd A_a) \cdot (A_a \lhd A2) \\ & + A_a \cdot (A2 \lhd A1) \cdot (A1 \lhd A_a) \\ & + A1 \cdot (A2 \lhd A_a) \cdot (A_a \lhd A1) \end{aligned} \quad (21)$$

and the probability of these four disjoint algebraic terms can be determined from the probabilistic formulas recalled in Section 5 so as to determine the failure probability of TE1:

$$\begin{aligned} & \Pr\{TE_1\}(t) \\ = {} & \int_0^t \left(\int_w^t \left(\int_w^u f_{A2}(v)dv \right) f_{A_a}(u,w)du \right) f_{A1}(w)dw \\ & + \int_0^t \left(\int_0^w \left(\int_v^w f_{A_a}(u,v)du \right) f_{A1}(v)dv \right) f_{A2}(w)dw \\ & + \int_0^t \left(\int_w^t \left(\int_w^u f_{A1}(v)dv \right) f_{A_a}(u,w)du \right) f_{A2}(w)dw \\ & + \int_0^t \left(\int_0^w \left(\int_v^w f_{A_a}(u,v)du \right) f_{A2}(v)dv \right) f_{A1}(w)dw \end{aligned} \quad (22)$$

The failure probability of the top event of the DFT of the HECS can hence be determined by using the inclusion-exclusion formula. It can be noted that this failure probability does not depend on the distribution considered for basic events as the probabilistic model of dynamic gates can accommodate any distribution for basic events.

In the particular case of exponential distributions with the failure rates given in Table 2, we obtain an unreliability of 95.92% after 100 hours. We have retained this mission time because it is the one retained in [3], even though the quantitative results obtained are different as basic components are considered as repairable in [3].

Basic component	Failure rate (h^{-1})
A1, A2, A	10^{-4}
M1, M2, M3, M4, M5	6×10^{-5}
MIU1, MIU2	5×10^{-5}
BUS1, BUS2	10^{-6}
HW	5×10^{-5}
SW	3×10^{-2}
OP	10^{-3}

Table 2 – Failure rates of the basic events of the DFT of the HECS, from [3]

As the exponential distribution is not necessarily the most suitable to model the failure of components as it does not take into account their aging, the failure probability of the HECS could be computed by considering other more suitable distributions, such as the Weibull distribution, for instance.

7 CONCLUSION & PROSPECTS

In this paper, we presented the behavioural and probabilistic model of Spare gates. On the one hand, the behavioural model allows to take into account any type – Cold, Warm, or Hot – of Spare gate and to determine the structure function of any DFT under a canonical form thanks to the behavioural model of gates PAND and FDEP from [10]. The qualitative analysis of DFTs can then be performed directly from this canonical form. On the other hand, the probabilistic model allows to perform the quantitative analysis of any DFT from the canonical form of its structure function. It can be noted that, as this probabilistic model does not depend on the distribution considered for basic events, any distribution can be accommodated during the quantitative analysis.

Ongoing work is currently addressed to the elaboration of efficient algorithms allowing to determine the structure function of DFTs and to perform their analysis directly from this structure function. Besides, the set of minimal cut sequences obtained with this approach is not necessarily minimal as it may contain redundant minimal cut sequences. We should hence define one or many minimization criterion and develop optimization algorithms allowing to reduce this set of minimal cut sequences.

REFERENCES

1. E. Henley, H. Kumamoto, *Reliability Engineering and Risk Assessment*, Prentice Hall, 1981.
2. N. Leveson, *Safeware: System Safety and Computers*, Addison-Wesley, 1995.

3. *Fault Tree Handbook With Aerospace Applications*, NASA Office of Safety and Mission Assurance, 2002, pp. 1-205.
4. J.B. Dugan, S. Bavuso, M. Boyd, "Fault Trees and Sequence Dependencies," *Proc. Ann. Reliability & Maintainability Symp.*, (Jan.) 1990, pp. 286-293.
5. J.B. Dugan, K.J. Sullivan, D. Coppit, "Developing a low-cost high-quality software tool for dynamic fault-tree analysis," *IEEE Trans. Reliability*, vol. 49, no. 1, (Mar.) 2000, pp. 49-59.
6. Z. Tang, J.B. Dugan, "Minimal cut set/sequence generation for dynamic fault trees," *Proc. Ann. Reliability & Maintainability Symp.*, (Jan.) 2004, pp. 207-213.
7. D. Coppit, K.J. Sullivan, J.B. Dugan, "Formal Semantics of Models for Computational Engineering: A Case Study on Dynamic Fault Trees," *Proc. 11th Int. Symp. on Software Reliability Engineering (ISSRE 2000)*, (Oct.) 2000, pp. 270-282.
8. A. Bobbio, D. Codetta-Raiteri, "Parametric Fault Trees with Dynamic Gates and Repair Boxes," *Proc. Ann. Reliability & Maintainability Symp.*, (Jan.) 2004, pp. 459-465.
9. S. Montani, L. Portinale, A. Bobbio, D. Codetta-Raiteri, "DBNet, a tool to convert Dynamic Fault Trees into Dynamic Bayesian Networks," Technical report TR-INF-2005-08-02, (Aug.) 2002.
10. G. Merle, J.-M. Roussel, J.-J. Lesage, A. Bobbio, "Probabilistic Algebraic Analysis of Fault Trees With Priority Dynamic Gates and Repeated Events," *IEEE Trans. Reliability*, vol. 59, no. 1, (Mar.) 2010, pp. 250-261.
11. V. Vittorini, G. Franceschinis, M. Gribaudo, M. Iacono, N. Mazzocca, "Drawnet: Model objects to support performance analysis and simulation of systems," *Proc. 12th Int. Conf. on Modelling Tools and Techniques for Computer and Communication System Performance Evaluation*, Springer Verlag – LNCS, vol. 2324, 2002, pp. 233-238.
12. J.B. Fussell, E.F. Aber, R.G. Rahl, "On the Quantitative Analysis of Priority-AND Failure Logic," *IEEE Trans. Reliability*, vol. R-25, no. 5, (Dec.) 1976, pp. 324-326.
13. G. Merle, "Algebraic modelling of Dynamic Fault Trees, contribution to qualitative and quantitative analysis," PhD thesis, ENS de Cachan, (Jul.) 2010.
14. G.R. Grimmett, D.R. Stirzaker, *Probability and Random Processes*, Oxford University Press, 2001.
15. K. Trivedi, *Probability & Statistics with Reliability, Queueing & Computer Science Applications*, Wiley, 2001.

BIOGRAPHIES

Guillaume Merle, PhD
LURPA, ENS Cachan
61 avenue du Président Wilson
Cachan, 94230, France

e-mail: merle@lurpa.ens-cachan.fr

Guillaume Merle received the PhD degree in Electrical and automation engineering and the MSc degree in Systems engineering at the Ecole Normale Supérieure de Cachan (France), respectively in 2010 and 2007. His main research interests span the area of algebraic methods, with application to performance evaluation, and reliability. He is a member of IEEE.

Jean-Marc Roussel, PhD
LURPA, ENS Cachan
61 avenue du Président Wilson
Cachan, 94230, France

e-mail: roussel@lurpa.ens-cachan.fr

Jean-Marc Roussel received the PhD degree in 1994. He is currently Associate Professor of Automatic Control at the Ecole Normale Supérieure de Cachan and carries out research at the LURPA (Automated Production Research Laboratory) on the control of Discrete Event Systems with algebraic approaches.

Jean-Jacques Lesage, Habil., PhD
LURPA, ENS Cachan
61 avenue du Président Wilson
Cachan, 94230, France

e-mail: lesage@lurpa.ens-cachan.fr

Jean-Jacques Lesage received the PhD degree in 1989, and the Habilitation in 1994. He is currently Professor of Automatic Control at the Ecole Normale Supérieure de Cachan. His research topics are formal methods and models of Discrete Event Systems (DES), both for modelling synthesis and analysis. The common objective of his works is to increase the dependability of the DES control. He is a member of IEEE.