

# Reliability Analysis of Dynamic Systems by Translating Temporal Fault Trees into Bayesian Networks

Sohag Kabir, Martin Walker, and Yiannis Papadopoulos

Department of Computer Science, University of Hull, Hull, UK

{s.kabir@2012., martin.walker@, y.i.papadopoulos@}hull.ac.uk

**Abstract.** Classical combinatorial fault trees can be used to assess combinations of failures but are unable to capture sequences of faults, which are important in complex dynamic systems. A number of proposed techniques extend fault tree analysis for dynamic systems. One of such technique, Pandora, introduces temporal gates to capture the sequencing of events and allows qualitative analysis of temporal fault trees. Pandora can be easily integrated in model-based design and analysis techniques. It is, therefore, useful to explore the possible avenues for quantitative analysis of Pandora temporal fault trees, and we identify Bayesian Networks as a possible framework for such analysis. We describe how Pandora fault trees can be translated to Bayesian Networks for dynamic dependability analysis and demonstrate the process on a simplified fuel system model. The conversion facilitates predictive reliability analysis of Pandora fault trees, but also opens the way for post-hoc diagnostic analysis of failures.

## 1 Introduction

Fault Tree Analysis (FTA) is a well-established and widely used method for evaluating system reliability which utilizes graphical representation based on Boolean logic to show logical connections between different faults and their causes [16]. FTA is a deductive analysis method, which means analysis starts with a system failure known as the top event and works backwards to determine its root causes. From a fault tree, it is possible to understand how combinations of failures of different components or certain environmental circumstances can lead to system failure. Qualitative analysis of fault trees are performed by reducing them to minimal cut sets (MCSs) which are the smallest combinations of failure events that are necessary and sufficient to cause system failure.

Increasingly, systems are getting more complex and their behaviour is becoming more dynamic as the behaviour of the system changes, functions and their failure modes vary, as do the flows between components of the architecture and the potential deviations of those flows. Due to this complex behaviour and the many possible interactions between the components, assessing the effects of combinations of failure events is not enough by itself to capture the system failure behaviour; in addition, understanding the order in which they fail is also required for a more accurate failure model. However, classical combinatorial fault trees are unable to capture this sequence-dependent dynamic behaviour [3, 5].

To overcome this limitation, a number of dynamic analysis techniques have been introduced, such as Dynamic Fault Trees (DFTs) [6] and Pandora temporal fault trees (TFTs) [18]. DFTs are traditionally analysed by translating them into Markov chains (MCs), but the major shortcoming of a Markov model is that for large and complex systems, it becomes very large and complicated and therefore difficult to construct and analyse. Alternatives have also been proposed, such as an algebraic framework to model dynamic gates of DFTs; this allows qualitative [11] and quantitative [12] analysis of DFTs.

Pandora introduces temporal gates to capture sequence-dependent behaviour and provides temporal logic to allow qualitative analysis. This technique can be used to determine minimal cut sequences (MCSQs) of temporal fault trees, which are the smallest sequences of events that are necessary to cause the top events, analogous to minimal cut sets of conventional fault trees. As temporal expressions are usually complex when compared to Boolean expressions, temporal laws can be used to minimise the expressions and therefore reduce complexity [17]. Like DFTs, Pandora temporal fault trees can also be solved by mapping them into MCs, but again this can lead to issues with the state-space explosion problem.

Bayesian Networks (BNs) have previously been used in the analysis of combinatorial fault trees, e.g. in [1]. They offer a number of advantages, alleviating some of the typical constraints of FTs, e.g. the need for statistical independence of events and exponential distributions of failure rate of events. They also allow both predictive and diagnostic analysis [13]. BNs can also be used to help minimise the state-space explosion problem by allowing compact representation of temporal dependencies among the system components and sequence-dependent failure behaviours [10], which has led to their use for analysing DFTs [2, 14]. However, so far no attempts have been made to use BNs to solve Pandora temporal fault trees. One of the advantages of Pandora is that it can be integrated well in model based design and analysis techniques. It has been shown, for instance in [17], that Pandora expressions incorporating sequence operators can be used to describe the local failure logic of components and then synthesis into temporal fault trees from systems models, e.g. expressed in popular notations such as SysML, EAST-ADL, AADL or Matlab Simulink, where models have been annotated with Pandora expressions. The principle has been shown within the HiP-HOPS model-based safety analysis tool, which has been connected experimentally to several commonly used modelling notations. Given the increasing importance of model-based design and analysis, and the potential use of Pandora in this context, we believe that it is both theoretically and practically useful to explore possible avenues for improved analysis of Pandora temporal fault trees. Therefore, in this paper, we show how the BNs can also be used to solve Pandora temporal fault trees, and thus enable us to perform quantitative analysis of Pandora TFTs focusing on prediction of reliability.

The rest of paper is organized as follows: Section 2 presents the fundamentals of Pandora temporal fault trees. The basic concepts of Bayesian Networks are presented in Section 3. Section 4 describes the method for converting Pandora temporal fault trees into Bayesian Networks. The method is then illustrated by applying it to a case study in Section 5. Finally, concluding remarks are presented in Section 6.

## 2 Introduction to Pandora temporal fault trees

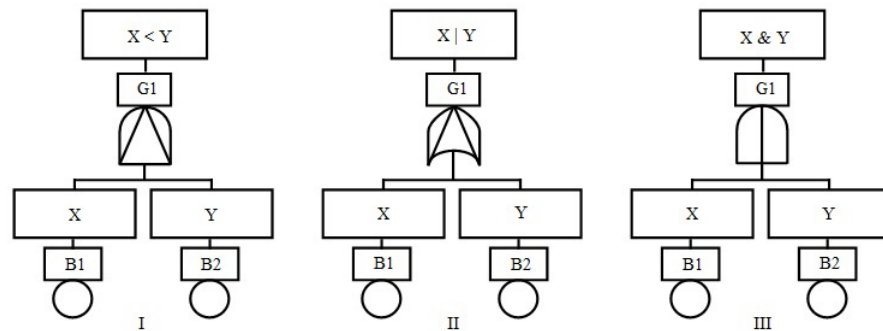
### 2.1 Pandora temporal gates

Pandora extends conventional FTs by defining three temporal gates: Priority-AND (PAND), Priority-OR (POR), and Simultaneous-AND (SAND). These gates allow analysts to represent sequences or simultaneous occurrence of events as part of a fault tree.

**Priority-AND (PAND):** The Priority-AND (PAND) gate is not a new gate and has been used in FTA as far back as the 1970s [8], and also features in the Dynamic Fault Tree methodology. However, it was never properly defined for use in qualitative analysis, resulting in ambiguous behaviour. In Pandora, therefore, the PAND gate is defined as being true only if the following conditions are true:

- All input events occur
- Input events occur from left to right
- No input events occur simultaneously

The symbol ' $<$ ' is used to represent the PAND gate in logical expressions, i.e.,  $X < Y$  means (X PAND Y) where X and Y are both failure events. The fault tree symbol of the PAND gate is shown in Fig.1 (I).



**Fig. 1.** Pandora temporal gates. (I) PAND. (II) POR. (III) SAND

**Priority-OR (POR):** Like the PAND gate, the Priority-OR (POR) gate also defines a sequence, but it specifies an ordered disjunction rather than an ordered conjunction. It is used to indicate that one input event has priority and must occur first for the POR to be true, but does not require all other input events to occur as well. The POR can therefore be used to represent trigger conditions where the occurrence of the priority event means that subsequent events may have no effect. The POR is true only if:

- Its left-most (priority) input event occurs

- No other input event occurs before the priority input event
- No other input event occurs at the same time as the priority input event

The symbol ‘|’ is used to represent the POR gate in logical expressions, thus  $X|Y$  means  $(X \text{ POR } Y)$  and the fault tree symbol of the POR gate is shown in Fig.1 (II).

**Simultaneous-AND (SAND):** The Simultaneous-AND or SAND gate is used to define situations where an outcome is only triggered if two or more events occur simultaneously. For example, this can happen because of a common cause, or because the events have a different effect if they occur approximately simultaneously as opposed to in a sequence. It is true only if:

- All input events occur
- All the events occur at the same time

The symbol ‘&’ is used to represent the SAND gate in logical expressions and the fault tree symbol of the SAND gate is shown in Fig.1 (III).

We further use ‘+’ to represent OR and ‘.’ to represent AND. The SAND gate has the highest priority in a logical expression, then PAND, POR, AND, and OR. Therefore  $P+Q\&R<S|T$  is equivalent to  $P+(((Q\&R)<S)|T)$ .

## 2.2 Pandora temporal logic

As well as the three temporal gates, Pandora also defines a set of temporal laws that describe the behaviour of the gates and how they relate to each other and to the standard Boolean AND and OR gates. These laws can all be proved, e.g. with temporal truth tables as in [18], and form the basis for qualitative analysis of Pandora’s temporal fault trees, allowing reduction and minimisation of the expressions to obtain the minimal cut sequences, or MCSQs.

Most important of these laws are the Completion Laws [18], which relate the temporal gates to the Boolean gates:

- Conjunctive Completion Law:  $X \cdot Y \Leftrightarrow X < Y + X \& Y + Y < X$
- Disjunctive Completion Law:  $X + Y \Leftrightarrow X | Y + X \& Y + Y | X$
- Reductive Completion Law:  $X \Leftrightarrow Y < X + X \& Y + X | Y$

## 2.3 Time representation in Pandora temporal fault trees

Pandora makes few assumptions about the model of time used in any particular system; time can be interval or point-based, discrete or continuous, or some hybrid thereof. The result is a flexible approach that can be adapted to both design-time exploration of possible failures and run-time diagnosis of faults that have already occurred. Pandora does impose three key restrictions, however:

- The model of time used must be linear, not branching;
- Events are persistent (i.e., once occurred, they remain in a ‘true’ state indefinitely)

- Events occur instantly (i.e., go from 'false' to 'true' with no delay).

As long as these restrictions are followed, there are only three possible temporal relations between two events: before, simultaneous, and after. These are all covered by the temporal gates PAND, SAND, and POR introduced above. In Pandora, therefore, the exact time at which an event occurs is not as important as the order in which the events occur; all that matters is whether an event occurs before, after, or at the same time as another event, as this is what defines the sequence.

### 3 Bayesian Networks

Bayesian Networks are based on a well-defined theory of probabilistic reasoning and provide a graphical framework to represent uncertain knowledge in artificial intelligence [15]. BNs consist of a qualitative part and a quantitative part. The qualitative part is a directed acyclic graph of nodes and arcs, like the one shown in Fig.2. The nodes of the graph represent the random variables (events) and arcs represent dependencies or cause-effect relation among nodes.

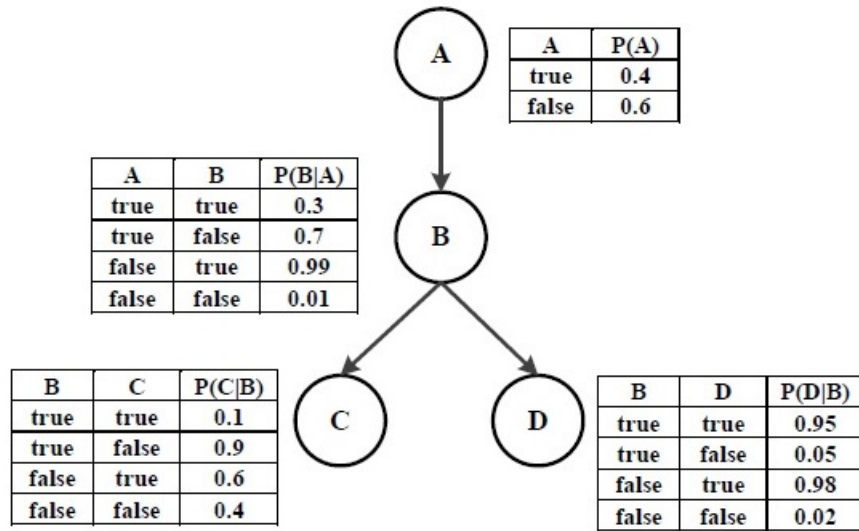


Fig.2. A Simple Bayesian Network

In Bayesian Network, a node X is said to be the parent of another node Y if there is an arrow from node X to node Y, e.g. node A is a parent of node B in Fig.2. A node without parent is known as a root node and node without a child is called a leaf node e.g. node A is a root node and node C and D are leaf nodes in Fig.2.

The quantitative part of BNs consists of a prior probability table for each of the root nodes and a conditional probability table (CPT) for each of the other nodes given the

status of their parents. For example, in Fig.2, the root node A has a prior probability table which shows the probability of A being in any of the permissible states, e.g.  $P(A = \text{true}) = 0.4$ . At the same time all other nodes have their own CPT. The CPT of a node shows the probability of that node being in any of its permissible states given the states of its parent nodes. For example, from the CPT of node D, we can see that the probability of the node D be true given that its parent B is false is 0.98, i.e.,  $P(D=\text{true} | B=\text{false}) = 0.98$ . A set of conditional independence statements are the main consideration while making BN models and conditional independence information can be obtained from the BN model by employing the rules of d-separation [15]. The joint probability distribution of a set of random variables  $\{X_1, X_2, X_3, \dots, X_n\}$  can be determined from conditional independence assumptions and using a chain rule as explained in [15]:

$$P(X_1, X_2, \dots, X_{n-1}, X_n) = \prod_{i=1}^n P(X_i | \text{Parent}(X_i)) \quad (1)$$

If  $A$  and  $B$  are two random events and evidence is found that event  $B$  has occurred, then using the Bayes theorem, the posterior probability of event  $A$  on condition that  $B$  has happened can be defined as:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (2)$$

where  $P(A)$  and  $P(B)$  are the prior probabilities of event  $A$  and  $B$  respectively.

BNs provide a robust probabilistic method of reasoning under uncertainty and they are capable of combining different sources of information to provide a global assessment of dependability attributes such as reliability and safety, therefore they have received much attention in last decade in the area of dependability analysis [1, 9].

## 4 Conversion of Temporal Fault Trees into Bayesian Networks

In order to convert temporal fault trees into BNs, it is required to understand how Pandora represents the sequencing of events. As mentioned earlier, in Pandora “the exact time at which an event occurs is not important—the only thing that matters is when it occurs relative to the other events, i.e. which comes first, which comes second, which comes last etc.” [18]. Therefore, sequence values, an abstraction of the relative time at which an event occurs are used in Pandora instead of a quantitative, absolute metric of time. If an event has not occurred then that event is given a sequence value 0. If an event has occurred then it is given a sequence value greater than 0 to indicate when it occurred relative to other events, i.e., all sequence value greater than 0 represents logical true, but the higher the value the later the events occurred, e.g. sequence value 1 means the event occurred first, 2 means it occurred second and so on. If two or more events occur simultaneously, then they will have the same sequence value. These same sequence values also make up the basis of temporal truth tables used in Pandora, as they combine the truth state (true/false) with the relative time of occurrence for each event. A temporal truth table showing the sequence values for each gate in Pandora given the sequence values of each input event are provided below:

**Table 1.** Temporal Truth Table for all gates in Pandora

<b>X</b>	<b>Y</b>	<b>X OR Y</b>	<b>X AND Y</b>	<b>X POR Y</b>	<b>X PAND Y</b>	<b>X SAND Y</b>
0	0	0	0	0	0	0
0	1	1	0	0	0	0
1	0	1	0	1	0	0
1	1	1	1	0	0	1
1	2	1	2	1	2	0
2	1	1	2	0	0	0
2	2	2	2	0	0	2

The first step in the conversion process is to translate the Pandora TFT into an equivalent discrete-time Bayesian Network, where each root node of the BN represents a basic event of the TFT and each gate (including both Boolean and temporal gates) of the TFT is represented as intermediate node. The root nodes must then each have a prior probability table defined, while the intermediate nodes must each have a conditional probability table (CPT) defined. The conversion of the TFT to BN is a simple one-to-one mapping from basic events and gates to root and intermediate nodes in which the original fault tree connections are maintained as parent/child relationships, but the prior probability table and CPT are populated based on the failure probability distributions of the basic events and the behaviour of the gates used in the temporal fault trees.

As the outcome of each temporal gate is dependent on the relative sequence values of the events involved in those gates, we divide the mission time  $T$  into  $n$  intervals from  $t=0$  to  $t=T$ , where  $n$  must be at least equal to the number of events, following the similar concept used in [2]. Each time interval represents a possible non-zero sequence value used in Pandora, during which the state of an event may change from not occurred (false) to occurred (true). At the beginning of the system operation, a component is considered as not failed therefore given a sequence value 0 (i.e., the component is in State 0) and this value remains the same until the component fails. If a component fails in interval 1, then it will have the sequence value 1, and so on.

The value of the events in each interval can be treated as a specific state. If an event is in State 0, then it means it has sequence value 0 (i.e., it did not occur). If an event is in State 1, then it means it occurred in interval 1, and if it is in State 2, then it occurred in interval 2, and so forth. This allows us to construct a CPT for all possible states of the input events to a gate, showing the probability that the gate will be in a particular state (i.e., has a given sequence value) depending on the states (sequence values) of the input events.

Note that in Pandora the outcome of any gate can be either true or false but every true value is associated with sequence information, i.e., the relative time at which the gate becomes true. Therefore, the probability of a node representing a gate being in a certain state given the state information of its parent nodes can either be 0 or 1. As an example, let us consider a 2 input POR gate. For simplicity, we choose  $n=2$  and thus divide the mission time  $T$  into 2 intervals. This yields three possible states for each

event: state 0, in which an event does not occur at all during the mission time T; state 1, in which the event occurs in the interval  $[0, T/2]$ , and state 2, in which the event occurs in the interval  $[T/2, T]$ . Each state is then associated with a probability, namely the probability of the event being in that state.

Fig.3 illustrates this example and shows the POR gate with two input events and its equivalent BN, where the events are independent. In the BN, nodes X and Y represent the independent events X and Y and their prior probability tables are populated from the failure rate information of the component failure modes X and Y respectively.

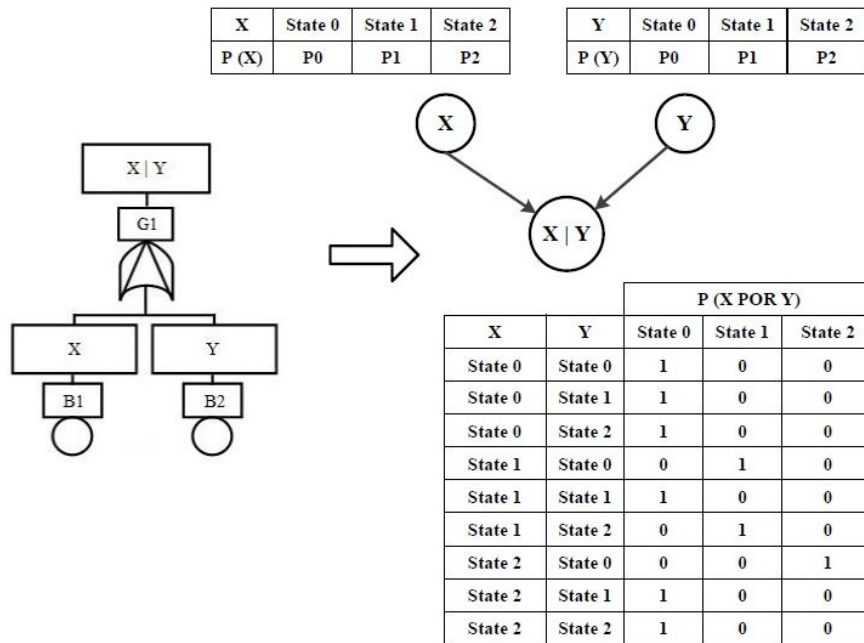


Fig. 3. Two input POR gate and its equivalent BN

The arcs from node X and Y to node (X|Y) shows the dependency of the POR gate outcome on its input events X and Y. The CPT shows the behaviour of the POR gate, i.e., it is true if its left-most (priority) input event occurs and no other events occur before or at the same as the priority event. A 1 in the State 0 column of the CPT means the outcome of POR gate is false — i.e., that it has a 100% probability of not occurring at all during the mission time T given those particular input states. Similarly, a 1 in the State 1 column means that it has a 100% probability of occurring during interval 1 given those particular inputs, and the same for State 2 and interval 2.

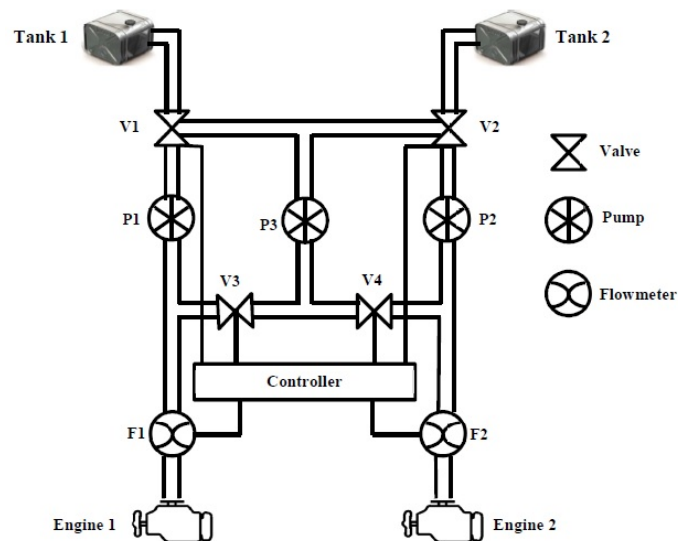
For example, we can say that the probability that the POR gate will be in State 0 given that event X is in State 1 and event Y is State 0 is 0, i.e.,  $P(X \text{ POR } Y = \text{State } 0 \mid X = \text{State } 1, Y = \text{State } 0) = 0$ . This is as we would expect, since the POR gate will have



the same sequence value as its priority event if the left-most (priority) event X occurs and the other event Y does not occur. Instead, the POR outcome should be in State 1, i.e.  $P(X \text{ POR } Y = \text{State 1} \mid X = \text{State 1}, Y = \text{State 0}) = 1$ . This can also be seen in the temporal truth table in Table 1: if X is 1 and Y is 0, then X POR Y should also be 1. This procedure can be repeated for all the gates used in Pandora such that their CPTs will resemble their equivalent temporal truth tables. It is possible to represent a TFT in different logically equivalent forms. In our experiments, translation of equivalent TFTs appear to produce equivalent BNs, although we have yet to prove this formally.

## 5 Case Study

To illustrate the idea of reliability analysis of dynamic systems by converting TFTs into BNs, we use the case study of fault tolerant fuel distribution system of a ship, first presented in [7] but reworked here, and shown in Fig.4.



**Fig. 4.** Fault tolerant fuel distribution system

Under ordinary operation, there are two primary fuel flows: Engine 1 gets fuel from Tank 1 through Pump 1 (P1), and Engine 2 gets fuel from Tank 2 through Pump 2 (P2). Flowmeter 1 (F1) and Flowmeter 2 (F2) monitor the rate of fuel flow to Engine 1 and Engine 2 respectively and report it back to the Controller. On detecting insufficient fuel flow to either of the engine, the Controller introduces dynamic behaviour to this system by activating standby Pump 3 (P3) and redirecting fuel flow accordingly using the valves V1-V4. For example, if insufficient fuel flow to Engine 1 is detected, then the Controller can activate Pump 3 and open Valves 1 and 3 (V1 and V3), and thus fuel

flows to Engine 1 through Pump 3 instead of Pump 1. On the other hand if a problem with fuel flow to Engine 2 is detected then Pump 3 will be activated and Valves 2 and 4 (V2 and V4) will be opened instead. Therefore, Pump 3 can take over the task of either Pump 1 or Pump 2, but not both. A failure of both Pump 1 and Pump 2 will cause at least one engine to be starved of fuel; for example, if Pump 1 fails and Pump 3 replaces it, then Pump 3 will no longer be available to replace Pump 2 if Pump 2 subsequently fails. This results in degraded propulsion functionality for the vessel, as speed and manoeuvrability will be reduced.

Pandora temporal gates can be used to model the dynamic behaviour in this scenario and helps to correctly capture the sequences of events that lead to failure. For simplicity, internal failure of the engines themselves is left out of the scope of this analysis. The Pandora temporal fault tree for the failure behaviour of Engine 1 of the fault tolerant fuel distribution system was constructed via model-based synthesis from Pandora descriptions of local failure logic of components and it is shown in Fig.5. As the failure of Engine 1 and Engine2 are caused by the similar events in the opposite sequences, the TFT of failure behaviour of Engine 2 looks almost identical except that P1 and P2 are reversed, and so is omitted for brevity. The basic events in the following TFT are:

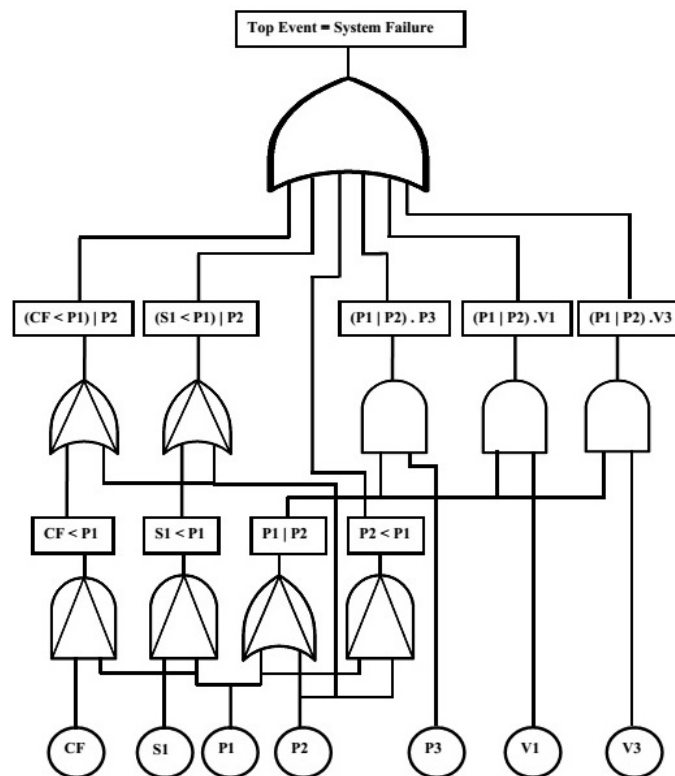
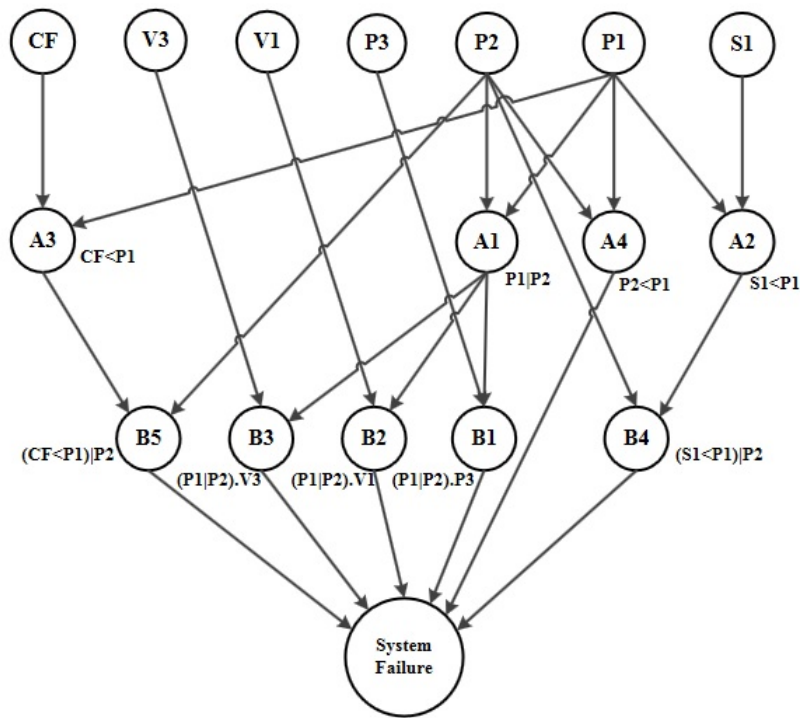


Fig. 5. TFT of failure behaviour of Engine 1

- P1/P2/P3 = Failure of Pump 1/2/3 (e.g. blockage or mechanical failure)
- V1/V3 = Failure of Valve 1/3 (e.g. blockage or stuck closed)
- S1 = Failure of Flowmeter 1 (e.g. sensor readings stuck high)
- CF = Failure of Controller

The TFT of Fig.5 is converted into a discrete-time BN following the procedure described in Section 4, shown in Fig.6. Failure rates of all components are assumed to be exponentially distributed and the values of the failure rates of each component are shown in Table 2. The prior probability table of each root node is populated based on the failure rate of corresponding component and the conditional probability tables of each intermediate node are populated following the same procedure described in Section 4.



**Fig. 6.** Bayesian Network of failure behaviour of Engine 1

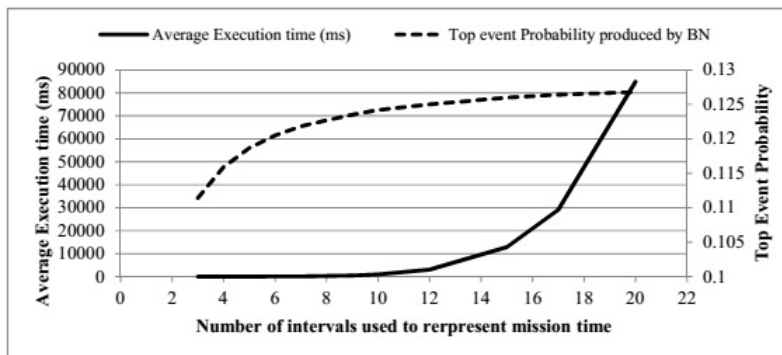
The computation for the system failure probability or unreliability using the BN is performed using the modified version of the JavaBayes tool [4] considering mission time  $T= 10000$  hours. The probability of top event occurrence for different values of  $n$  is shown Table 3. For comparison, an analytical technique [7] for solving the TFT yields a value of 0.1353. We can see that as the value of  $n$  increases, the accuracy of the approximation increases and the value of top event probability converges towards a ceiling value.

**Table 2.** Failure rates of components of fuel distribution system

Component	Failure rate/hour ( $\lambda$ )
Tanks	1.5E-5
Valve1 & Valve2	1E-5
Valve3 & Valve4	6E-6
Pump1 & Pump2 & Pump3	3.2E-5
Flowmeter Sensor	2.5E-6
Controller	5E-7

**Table 3.** Top event probability of redundant fuel distribution system

$n$	Top Event Probability	Average Execution Time (ms)
3	0.1114	15.6
4	0.1159	25.2
5	0.1187	40.6
6	0.1205	74.8
7	0.1218	149.6
8	0.1227	306.0
9	0.1235	493.0
10	0.1242	1011.2
12	0.1250	3070.0
15	0.1260	12879.4
17	0.1264	29162.8
20	0.1268	84889.0



**Fig. 7.** Changes in unreliability and average execution time with the change of value of  $n$

Execution time is taken as an average over 5 runs for each value of  $n$  and increases by a factor of approximately 1.6 for every additional interval, but it also results in greater accuracy. Therefore, the value of  $n$  represents the tradeoff one can make between the execution time and the accuracy. In this case, an acceptable tradeoff can be found at value of  $n = 12$  which could provide reasonable accuracy.

## 6 Conclusion

In this paper, we have discussed how a recent extension of fault tree analysis, Pandora, allows capturing sequence dependent failure behaviour in dependability analysis. An advantage of Pandora is that it is easily integrated in increasingly popular model-based design and analysis techniques. In this paper we identified BNs as a potential framework for improved solving of Pandora temporal fault trees. BNs provide a robust probabilistic method of reasoning under uncertainty and they are capable of combining different sources of information to provide a global assessment of dependability in terms of its various attributes. We have presented a way of translating Pandora TFTs to BNs to perform dynamic dependability analysis focusing on reliability and demonstrated the process on an example case study. After mapping the TFTs into a BN, the BN model was solved to perform predictive analysis to compute the top event probability, i.e. system unreliability. The execution speed and precision of the results obtained from the proposed BN model is dependent on the number time slots used to represent the mission time. Therefore, users can make a tradeoff between execution time and accuracy by choosing the number of time intervals to model in the BN. At present, we have considered the system components as non-repairable, but in the future, we hope to extend this work by considering reparability of components. At the same time we have a plan to perform a numerical comparison between the performance of the BN-based solution to Pandora TFTs and other existing solution techniques, e.g. Markov chains. The conversion presented in this paper also opens the road for demonstrating the ability for post-hoc diagnostic analysis of failures in the context of Pandora-enabled model-based dependability analysis, a process which involves calculating and updating the posterior probability of nodes given observed evidence in other nodes.

## References

1. Bobbio, A., Portinale, L., Minichino, M., Ciancamerla, E.: Improving the analysis of dependable systems by mapping fault trees into Bayesian networks. *Reliability Engineering & System Safety* 71(3), 249–260 (2001)
2. Boudali, H., Dugan, J.: A new bayesian network approach to solve dynamic fault trees. In: *Proceedings of Annual Reliability and Maintainability Symposium*. pp. 451–456 (Jan 2005)
3. Bruns, G., Anderson, S.: *Validating Safety Models with Fault Trees*. In: Górski, J. (ed.) *SAFECOMP 93*, pp. 21–30. Springer London (1993)
4. Cozman, F.: *JavaBayes*. <http://www.cs.cmu.edu/~javabayes/> (2001)
5. Dugan, J.B., Bavuso, S.J., Boyd, M.A.: *Fault Trees and Sequence Dependencies*. In: *Proceedings of Annual Reliability and Maintainability Symposium*. pp. 286–293 (Jan 1990)
6. 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)

7. Edifor, E., Walker, M., Gordon, N.: Quantification of Priority-OR Gates in Temporal Fault Trees. In: Ortmeier, F., Daniel, P. (eds.) *Computer Safety, Reliability, and Security*, pp. 99–110. Springer (2012)
8. Fussell, J., Aber, E., Rahl, R.: On the Quantitative Analysis of Priority-AND Failure Logic. *IEEE Transactions on Reliability* R-25(5), 324–326 (1976)
9. Langseth, H., Portinale, L.: Bayesian networks in reliability. *Reliability Engineering & System Safety* 92(1), 92–108 (2007)
10. Marquez, D., Neil, M., Fenton, N.: Solving Dynamic Fault Trees using a New Hybrid Bayesian Network Inference Algorithm. In: *16th Mediterranean Conference on Control and Automation*. pp. 609–614. IEEE (2008)
11. Merle, G., Roussel, J.M., Lesage, J.J.: Algebraic determination of the structure function of Dynamic Fault Trees. *Reliability Engineering & System Safety* 96(2), 267–277 (2011)
12. Merle, G., Roussel, J.M., Lesage, J.J.: Quantitative Analysis of Dynamic Fault Trees Based on the Structure Function. *Quality and Reliability Engineering International* 30(1), 143–156 (2014)
13. Montani, S., Portinale, L., Bobbio, A., Codetta-Raiteri, D.: Radyban: A tool for reliability analysis of dynamic fault trees through conversion into dynamic Bayesian networks. *Reliability Engineering & System Safety* 93(7), 922 – 932 (2008)
14. Montani, S., Portinale, L., Bobbio, A., Varesio, M., Codetta-Raiteri, D.: A tool for automatically translating Dynamic Fault Trees into Dynamic Bayesian Networks. In: *Annual Reliability and Maintainability Symposium (RAMS'06)*. pp. 434–441. IEEE (2006)
15. Pearl, J.: *Probabilistic reasoning in intelligent systems: Networks of Plausible Inference*. Morgan Kaufmann (1988)
16. Vesely, W., Dugan, J., Fragola, J., Minarick, R., Rallsback, J.: *Fault Tree Handbook with Aerospace Applications*. Tech. rep., NASA office of safety and mission assurance, Washington, DC (2002)
17. Walker, M., Papadopoulos, Y.: Qualitative temporal analysis: Towards a full implementation of the Fault Tree Handbook. *Control Engineering Practice* 17(10), 1115–1125 (2009)
18. Walker, M.D.: *Pandora: A Logic for the Qualitative Analysis of Temporal Fault Trees*. Ph.D. thesis, University of Hull (2009)