Evolved Methods for Risk Assessment

Andrew Jackson
Resilience Engineering, The University of Nottingham, U.K., Email: Andrew.jackson2@nottingham.ac.uk

Silvia Tolo
Resilience Engineering, The University of Nottingham, U.K., Email: Silvia.tolo@nottingham.ac.uk

John Andrews
Resilience Engineering, The University of Nottingham, U.K., Email: John.andrews@nottingham.ac.uk

The foundations of risk assessment tools such as fault tree analysis and event tree analysis were established in the 1970s. Since then, research has made considerable advances in the capabilities of analytical techniques applicable to safety critical systems. Technology has also advanced and system designs, their operation conditions and maintenance strategies are now significantly different to those of the 1970s.

This paper presents an overview of a new methodology developed, retaining the traditional ways of expressing system failure causality, which aims to develop the next generation of risk assessment methodologies. These evolved techniques, appropriate to meet the demands of modern industrial systems, aim to overcome some of the limitations of the current approaches. These new tools and techniques will seek to retain as much of the current methodology features as possible to reduce the learning curve for practitioners and increase the chances of acceptance.

The new approach aims to increase the scope of event tree/fault tree analysis through the incorporation of Petri net, Markov model, and binary decision diagram-based methodologies. Use of these techniques incorporates features such as: non-constant failure rates, dependencies between component failure events, and complex maintenance strategies to boost the capabilities of the methods.

In addition, it considers dedicated routines to analyse the accident risk of transport systems formulated as phased mission models. This type of modelling is demonstrated through the application to an aeronautical system, where the system is modelled as a mission consisting of a series of phases. Mission success requires the successful completion of each of the phases. This approach allows the requirements for success (and therefore failure) to differ from one phase to another. It is also possible to model scenarios whereby a system fault that occurs in one phase of a mission may not affect the system until a later phase of the mission.

Keywords: Risk assessment, Petri net (PN), Binary decision diagram (BDD), Fault tree, Event tree, Phased mission model

1. Introduction

Risk assessment methodologies are being applied to increasingly complex safety critical engineering systems. The complexity of these systems is advancing not just in terms of technology and system design, but also their operational conditions and maintenance strategies. Commonly adopted methodologies for risk assessment of such systems currently include Fault trees (FTs) and Event trees (ETs). These methodologies were developed in the 1970s and have limitations when attempting to accurately model certain features often found in modern systems such as non-constant failure rates, dependencies between component failure events, and complex maintenance strategies. Since the FT and ET methodologies were initially developed, considerable advances have been made in the development of analytical techniques which address some of the shortcomings of these methodologies when applied to modern engineering systems. These techniques often utilise Petri nets (PNs) (Chew et al. (2008), Andrews et al. (2014), Bryant et al. (2017)) or Markov models (MMs) (Bouissou & Bon (2003), Bäckstrom et al. (2016)). However, many of these techniques are often overlooked or under-utilised by risk assessment practitioners who favour more familiar FT & ET-based techniques.

This paper presents an overview of a new risk assessment methodology which aims to overcome some of the limitations imposed by FT & ET-based approaches and meet the demands of modern industrial systems. This new methodology seeks to incorporate and extend some of the enhanced techniques that have been developed to model and assess risk within complex systems; whilst retaining as much of the FT & ET methodology features as possible to reduce the learning curve for practitioners and increase the chances of acceptance.

2. Background

Fault tree analysis is a widely used technique to assess the probability and frequency of system failure in many industries. By providing information which enables the probability of basic events to be calculated, the fault tree can then be

Proceedings of the 31st European Safety and Reliability Conference
Edited by Bruno Castanier, Marko Cepin, David Bigaud, and Christophe Berenguer
Copyright © ESREL 2021.*Published by* Research Publishing, Singapore.
ISBN: 978-981-18-2016-8; doi:10.3850/978-981-18-2016-8_194-cd

quantified to yield reliability parameters for the system. Such parameters typically sought by risk assessment practitioners include top event probability, top event unconditional failure intensity, top event failure rate, expected number of top event occurrences in a specified time period, and total system downtime in a specified time period. Importance measures, indicating the contribution that each component makes to the system failure can also be obtained through fault tree analysis.

Phased missions are a common scenario in engineering, particularly within transport-based systems, whereby a system operates through several sequential and distinct periods of time (phases), during which the modes and consequences of failure can differ. For the mission to be a success the system must operate throughout all the phases. Component failures may occur at any point during the mission, yet not affect the system performance until the phase in which their condition is critical. Fault tree analysis can be adapted for, and is commonly applied to, phased mission analysis. Typically, individual fault trees are constructed each representing one phase of the mission. This approach normally necessitates the use of non-coherent fault trees (NOT gates included), as fault trees representing phases occurring after the first phase must model the successful completion of any earlier phases.

A key objective of the new methodology is that its relevance to practitioners is maximised through its ability to produce all the metrics obtainable through traditional FT analysis, whilst being applicable to more complex scenarios/systems and potentially offering increased accuracy and/or efficiency of calculation.

3. Base Model

3.1 *Base Model Inputs*

The first step towards the generation of models for the proposed risk analysis methodology is the input of:

- component failure/repair information;
- subsystem fault tree structures;
- a system event tree structure;
- component/subsystem dependency information.

Note that, with the exception of the last entry in the list, these inputs are of the same form as those that would be required for a traditional FT/ET risk modelling approach.

3.2 *Base Methodology*

Two of the most significant limitations of traditional FT/ET analysis techniques are: the assumption that failure/repair rates are constant, and the assumption of stochastic independence amongst a system's components. The techniques employed by the new methodology presented in this paper to overcome these limitations will be described within this section.

The inputs described in the previous section are first analysed to identify the presence of any features that cannot be fully captured by traditional FT/ET-based techniques. Such features include component failure dependencies, time-dependent failure modes, or complex maintenance strategies. If any such features are present, the relevant components/systems are encapsulated within independent sub-models, referred to as complex events. The required reliability data for these complex events is then obtained through the application of appropriate simulation strategies, for example via the use of PNs or MMs. When a complex event incorporates dependencies between two or more elements, outputs are recorded in terms of joint probability values covering all possible combinations of component states.

The reliability information obtained from complex event models is then integrated into the analysis of the individual sub-system models described by the FT structures input by the user. The model converts these FT structures to Binary Decision Diagrams (BDDs) for the computation of sub-system failure probability or frequency.

The results obtained from the numerical analysis of the sub-system BDDs are combined according to the event tree structure input by the user to enable the calculation of the overall system reliability.

3.3 *Base Methodology Case Study*

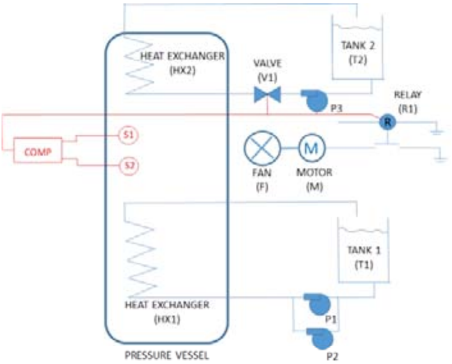


Figure 1- Case study model of a power plant cooling system for demonstration of concepts

The methodology briefly outlined in the previous section is now demonstrated in more detail through its application to the case study system shown in Fig. 1.

The case study models a simplified power plant cooling system consisting of four sub-systems: a primary cooling system, a secondary cooling system, a detection system, and a fan system.

Any combination of subsystem failures has the potential to affect the performance and operation of the overall system, as shown by the event tree in Fig. 2. For instance, the failure of the primary cooling system can result in total loss of cooling only if in combination with the failure of the detection system (preventing the activation of secondary mitigation measures) or with the simultaneous unavailability of the secondary and fan cooling. Conversely, if one of the latter two subsystems operates correctly but not the other, only partial loss of cooling will occur. Finally, if all but the primary cooling subsystem are available, no cooling loss is registered.

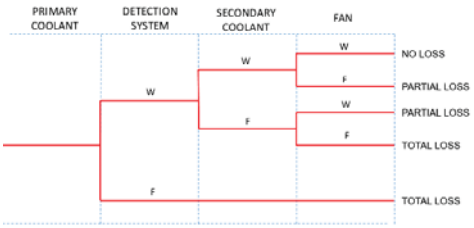


Figure 2- Event tree for the cooling system in Figure 1

The examples presented in this paper will focus on the primary cooling system only. The primary cooling system consists of a heat exchanger (HX1) which is fed cooling water from a storage tank (T1). Circulation of cooling water is ensured by the operation of one of two pumps (P1 & P2), with P2 operating as a warm standby for P1. The failure of either HX1 or T1 prevents the correct functioning of the primary cooling system. Similarly, the simultaneous unavailability of the circulation pumps P1 and P2 leads to the overall failure of the subsystem. Conversely, the failure of only one of the two pumps still results in the correct operation of the primary cooling system. A fault tree summarising the possible failure modes of the primary cooling system is shown in Fig. 3.

The first step of the proposed approach consists of converting individual independent subsystem FTs into BDDs. A BDD is a directed acyclic graph consisting of terminal and non-terminal vertices connected by edges (also referred to as branches). Each non-terminal vertex is associated with a basic event (e.g., the failure of an individual component) and is the origin of two branches: a 0-branch representing the non-occurrence of the basic event (e.g., the working state of the component) and a 1-branch representing the occurrence of the basic event (e.g., component failure). Terminal vertices, in which all paths through the BDD terminate, assume either a 0 value, associated with the working state of the system, or 1, indicating the failure of the system. For instance, considering a system consisting of a single component X and adopting the i*f-then-else* logic structure, this can be expressed as *ite(X,1,0):* if X fails then the entire system fails (terminal vertex equal to 1); on the contrary, if X works the entire system works (terminal vertex equal to 0). One of the advantages associated with BDDs is the ease with which a tree can be converted to represent its complementary top event.

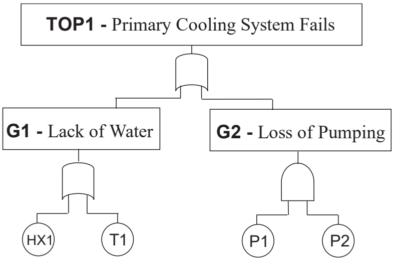


Figure 3- Fault tree for the primary cooling subsystem

The conversion from FT to BDD is carried out following the method developed by Rauzy (1993). The complexity of the BDD produced by applying this procedure, strongly depends on the variable ordering selected. Various variable ordering heuristics are available and can be implemented within the prosed methodology. In the current study, the special ordering suggested by Sinnamon & Andrews (1997) has been adopted: FT gate events are considered in a top-down ordering, with the exception that at each gate the input basic events are listed with the repeated events first (if the gate has more than one repeated event as an input then the most repeated event is placed first).

This results in the ordering *HX1<T1<P1<P2* for the primary cooling FT. Implementing the BDD construction rules, the resulting model for the primary cooling system is computed as follows:

$$\begin{aligned} G1 &\rightarrow ite(HX1,1,0) + ite(T1,1,0) \\ &= ite\big(HX1,1,ite(T1,1,0)\big) \end{aligned} \tag{1}$$

$$\begin{aligned} G2 &\rightarrow ite(P1,1,0).\,ite(P2,1,0) \\ &= ite(P1,ite(P2,1,0),0) \end{aligned} \tag{2}$$

$$\begin{aligned} TOP1 &\rightarrow ite\big(HX1,1,ite(T1,1,0)\big) + \\ &\quad ite(P1,ite(P2,1,0),0) \\ &= ite\Big(HX1,1,ite\big(T1,1,ite(P1,ite(P2,1,0),0)\big)\Big) \end{aligned} \tag{3}$$

Fig. 4 shows the structure of the BDD for the primary cooling system. Once the structure of the BDDs has been generated, their numerical analysis can be carried out based on the component reliability information. Traditional FT/ET-based approaches would dictate that only constant (or assumed constant) component failure/repair rates can be used for this type of analysis. Although largely accepted in engineering practice, the use of constant failure/repair rates is not always justified and may not adequately depict the behaviour of some components (e.g., subject to ageing). In these cases, the adoption of non-constant failure/repair rates may represent a more attractive option and enhance the accuracy of the analysis.

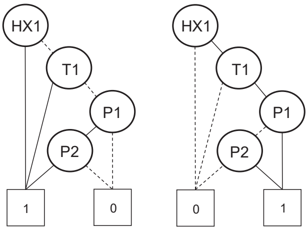


Figure 4– BDD (left) and DBDD (right) for the primary cooling sub-system

For this case study, PNs will be used to allow failure and repair times to be modelled by distributions representing non-constant rates. These are adopted to compute the reliability and failure frequency of components simulating the stochastic alternation of failure events and repairs. For instance, with regards to the primary cooling system, let component HX1 be characterised by non-constant failure and repair rates. Additionally, let us assume that the warm standby operation of P2 results in it failing at a lower rate when P1 is active. The remaining failure/repair rates are assumed to be constant.

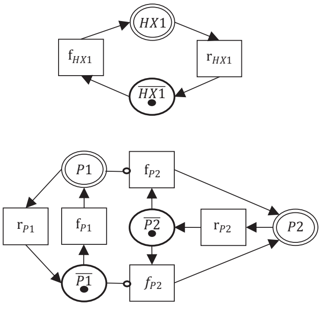


Figure 5- PN models of components HX1 (top), P1 and P2 (bottom) failure-repair cycles

The basic PN models portraying the failure mode of components HX1, P1 and P2 are shown in Fig. 5. For each component there are two PN places: one associated with the correct operation of the component ($\overline{HX1}$, $\overline{P1}$ and $\overline{P2}$ in Fig. 5), the other ($HX1$, $P1$ and $P2$) indicating their failure state. The presence of a token in one or the other place denotes the current state of the component. The firing of the stochastic failure transitions ($\mathrm{f}_{\mathrm{HX1}}$, f_{P1}, and f_{P2} or f_{P2} (where f_{P2} represents the reduced rate of failure when P2 is in standby mode)) causes the movement of the token from the working to the failure state, while the repair transitions ($\mathrm{r}_{\mathrm{HX1}}$, r_{P1} and r_{P2}) simulate the completion of corrective maintenance and hence the restoration of the correct component functionalities. The reliability of the individual elements can then be calculated as the ratio of down time over the system simulated time. Similarly, the failure frequency is computed as the number of failures recorded per unit time.

The unconditional system failure intensity

$$w_{SYS}(t)dt = \sum_{i} G_i(\boldsymbol{q}).w_i(t)dt, \tag{4}$$

can be calculated directly from the BDD structure and component availability data, avoiding the use of approximation. $w_i(t)$ is the failure intensity of component *i*. The criticality function $G_i(\boldsymbol{q})$, is the probability that the system is in a critical state for component *i* such that failure of component *i* causes system failure.

For a component *i*,

$$\begin{aligned} G_i(\boldsymbol{q})w_i(t) &= [Q(1_i,\boldsymbol{q}) - Q(0_i,\boldsymbol{q})]w_i(t) \\ &= Q(1_i,\boldsymbol{q})w_i(t) - Q(0_i,\boldsymbol{q})w_i(t), \end{aligned} \tag{5}$$

where $Q(1_i,\boldsymbol{q})w_i$ is the probability that the system fails with component *i* failed, and $Q(0_i,\boldsymbol{q})w_i$ is the probability that the system fails with component *i* working.

All paths p_n to a terminal 1, representing system failure, are obtained from the BDD. For each path, the contribution that each initiating event makes to $Q(1_i,\boldsymbol{q})w_i$ (i.e., paths that go through the component *i* BDD node 1 branch) and $Q(0_i,\boldsymbol{q})w_i$ (i.e., paths that go through the component *i* BDD node 0 branch) is calculated. (and summed for all paths) The contributions are then summed

$$\forall i \qquad Q(1_i,\boldsymbol{q})w_i = \sum_{p_n=1}^{p_T} (Q(1_i,\boldsymbol{q})w_i)_{p_n} \tag{6}$$

and

$$\forall i \qquad Q(0_i,\boldsymbol{q})w_i = \sum_{p_n=1}^{p_T} (Q(0_i,\boldsymbol{q})w_i)_{p_n}, \tag{7}$$

where $(Q(1_i,\boldsymbol{q})w_i)_{p_n}$ and $(Q(0_i,\boldsymbol{q})w_i)_{p_n}$ are the contribution that an initiating event on path p_n

makes to $Q(1_i, \boldsymbol{q})w_i$ and $Q(0_i, \boldsymbol{q})w_i$ respectively, and p_T is the total number of BDD paths to a terminal 1.

The advantage of this method is that it does not require the use of conditional probabilities to account for the location of dependent components within the BDD. All dependent events can be grouped together (e.g., $P(P1, P2)$) and the appropriate probability extracted from the PN/MM used to model the dependent components. The paths obtained using this method remain disjoint, therefore the calculations are exact.

Once the availability of each subsystem and the frequency of the trigger event are known, the analysis can proceed towards the computation of the event tree (Fig. 2). Since the case under consideration assumes independence between subsystems, the results are calculated as follows:

$$No\ Loss\ = F_P * (1 - Q_D)) * (1 - Q_D) * (1 - Q_F\ ) \tag{8}$$

$$PartialLoss = F_P * (1 - Q_D) * (1 - Q_S) * Q_F + F_P * (1 - Q_D) * (Q_S) * (1 - Q_F\ ) \tag{9}$$

$$TotalLoss = F_P * (Q_D) + F_P * (Q_D) * Q_S * Q_F, \tag{10}$$

where F_P indicates the failure frequency associated with the primary cooling, Q_D the availability of the detection system, Q_F the availability of the fan system, and $1 - Q_S$ the probability associated with the working state of the secondary system.

The proposed approach, based on the computation of individual component reliability using PNs and the subsequent integration of the obtained output in the quantitative analysis of subsystem BDD, can be applied in a similar form for modelling components characterised by complex maintenance strategies.

The methodology may also be adapted to model systems with ‘hard’ dependencies, where common components are shared between sub-systems; and ‘soft dependencies, triggered by secondary procedures or processes, which may be not strictly connected with the hardware function (e.g., maintenance, surrounding conditions, load changes etc.).

4. Phased mission methodology

A phased mission consists of a sequential series of varying time intervals referred to as phases. The success of a mission requires the successful completion of each of the phases. Conversely, the failure of a mission is expressed as the loss of the function of the system during at least one of the phases. The probability of this is the mission unreliability.

Phase and mission failures can be expressed in terms of the various system, sub-system, or component-level (basic event) failures that can cause them. The requirements for success (and therefore failure) differs from one phase to another and will have different failure logic models. Failure can occur during a phase or at the phase change.

Phased mission modelling is typically well suited to transport systems. Most applications are non-repairable during the mission (e.g., aircraft, trains) but for some applications some maintenance is possible (e.g., ships).

4.1 *Base Model Inputs*

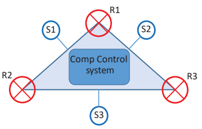
For application of the proposed methodology to phased mission analysis the following user-inputs are required:

- component failure/repair information;
- mission phase fault tree structures;
- mission phase duration and failure mode information;
- component/subsystem dependency information.

Note that, except for the last entry in the list, these inputs are of the same form as those that would be required for a traditional FT/ET-based phased mission modelling approach.

4.2 *Phased mission case study*

Various aspects of the phased mission methodology will be demonstrated through their application to the case study system shown in Fig. 6.



Drone System Components:

Camera sensors that send overlapping images to computer. Full 360 degrees covered by any 2 cameras.

Computer processes information sent by sensors.

Rotors that provide the lift to keep the drone airborne. Fly drone to locations.

Figure 6 – Case study model of an autonomous drone system for demonstration of phased mission methodology

The case study system models an autonomous drone whose mission is to fly to and land at another location. Details of the mission phases and failure scenarios are provided in Table 1. All components are assumed to be non-repairable during a mission.

The user inputs to the model are first analysed to identify the presence of any features that cannot be fully captured by traditional FT/ET-based techniques. If any such features are present, the relevant components/systems are encapsulated as complex events. The required reliability data for these complex events can then be obtained through the application of appropriate modelling strategies, such as PNs or MMs.

Table 1– Autonomous drone system mission phase details

Mission Phases	Mission Failure causes
1.Take off	- Fail to activate rotors (C1) - 2 out of 3 rotors fail (R1-R3)
2. Identify destination	- Fail to identify destination (C2) - 2 out of 3 sensors fail (S1-S3)
3. Fly to destination	- Fail to navigate to location (C3) - 2 out of 3 sensors fail (S1-S3)
4. Land	- Fail to turn off rotors (C4)

The drone's rotor system consists of inter-dependent components whereby the failure rate of the remaining operational components increases following a single component failure. The rotor system components in the drone mission phase 1 (Ph_1) FT are substituted with a complex event (Fig. 7) and for this case study the model generates a PN (Fig. 8) representing the rotor system. PN simulations are run to determine the mean time to failure (MTTF) μ for the complex event. The MTTF is used to generate a failure rate,

$$\lambda_{C2000} = \frac{1}{\mu} \qquad (11)$$

for the complex event representing the rotor system. The reliability information obtained from complex event models is then integrated into the analysis of the mission phase FT structures.

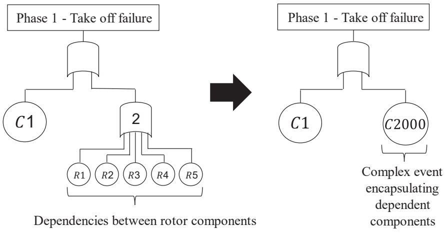

Phase 1 - Take off failure
C1
2
R1
R2
R3
R4
R5
Dependencies between rotor components
Phase 1 - Take off failure
C1
C2000
Complex event encapsulating dependent components

Figure 7 – Substitution of dependent components (R1, …,R3) with a complex event (C2000) in mission phase 1 FT

The reliability of the phased mission cannot simply be calculated by multiplying the reliabilities of each of the individual phases as this involves the false assumptions that the phases are independent, and all components are in the working state at the beginning of each phase. Use of this methodology would result in an appreciable over-prediction of system reliability.

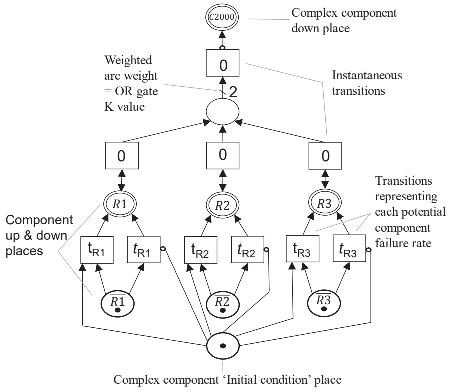

C2000
Complex component down place
Weighted arc weight = OR gate K value
0
2
Instantaneous transitions
0
0
0
R1
R2
R3
Transitions representing each potential component failure rate
Component up & down places
t_R1
t_R1
t_R2
t_R2
t_R3
t_R3
R1
R2
R3
Complex component 'Initial condition' place

Figure 8 – PN for modelling drone rotor system component failures

The individual phase fault trees are combined using a technique that enables the probability of failure in each phase to be determined in addition to the whole mission unreliability. For any phase, the method combines the causes of success of previous phases with the causes of failure for the phase being considered to allow both qualitative and quantitative analysis of both phase failure and mission failure.

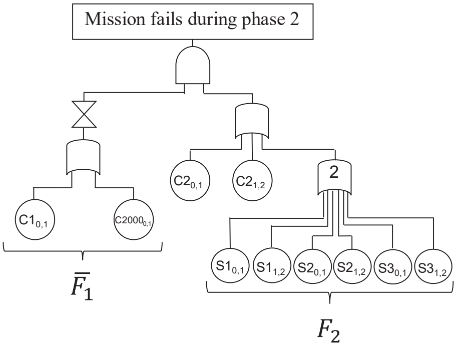


Figure 9 – FT structure for drone mission failure during phase 2. The logic structure representing NOT failure during mission phase 1 ($\bar{F}_1$) is combined with the logic structure representing failure during mission phase 2 (F_2). Events spanning multiple phases (e.g., $S1_{0,2}$ = failure of sensor 1 between 0 (mission start) and end of mission phase 2) are split into single phase failure events (e.g., $S1_{0,2}$ becomes $S1_{0,1}$ and $S1_{1,2}$).

The event of component failure in phase *i* is represented as the event that the component could have failed during any phase up to and including phase *i*. System failure F_i in phase *i* is represented as the success (i.e. NOT failure, $\bar{F}$) of phases *1..i*-1 AND the failure during phase *i*. This method allows for the evaluation of individual phase failures, and accounts for the condition where components are known to have functioned to enable the system to function in previous phases. For any phases after the first phase, the incorporation of the success of previous phases means that the fault trees will be non-coherent and not simply consist of AND and OR gates, as NOT logic is required to represent this success.

The individual phase FT structures are expanded using NOT logic to model non-failure (i.e., success) of the system in earlier phases. Events that could occur across multiple phases are split into multiple events each representing occurrence of the event in a single phase (Fig. 9).

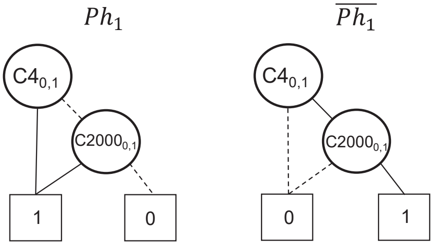


Figure 10 – BDD (left) and DBDD (right) for drone mission phase 1, where C4<C2000

Each of the mission phase fault trees are converted into BDD format and each variable in the BDD is associated with the time interval over which the variable can contribute to phase failure.

DBDDs are used to evaluate the successful completion of a previous phase when considering failure in phase *i > 1* of a phased mission (Fig. 10).

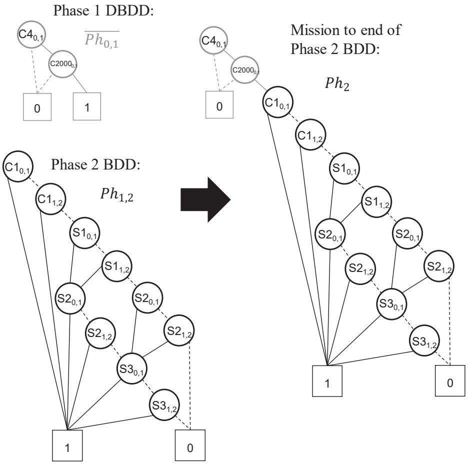


Figure 11 – Union of phase 1 $\overline{Ph_{0,1}}$ DBDD and phase 2 $Ph_{1,2}$ BDD to form mission to end of phase 2 Ph_2 BDD. The variable ordering heuristic of Sinnamon & Andrews (1997) used in section 3.1 has been applied

The phase BDDs and DBDDs can then be connected according to

$$Ph_i = \overline{Ph_{0,1}}.\overline{Ph_{1,2}}.\cdots.\overline{Ph_{i-2,i-1}}.Ph_{i-1,i}, \qquad (12)$$

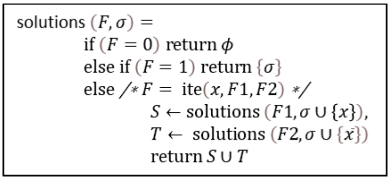
where Ph_i is the BDD representing mission failure during phase i (Fig. 11).

The BDDs are used for the computation of phase failure probabilities or frequencies and overall mission unreliability. Complex events are represented as single vertices within these BDDs; which are assigned the reliability data calculated via the complex event models/simulations. Various variable ordering heuristics can be implemented within the methodology during BDD construction.

Quantifying the phase and mission BDDs involves either tracing along all paths from the root vertex to terminal 1 vertices and calculating the failure probability for each; or tracing along all paths from the root vertex to terminal 0 vertices and calculating the success probability for each. Since the paths are disjoint the failure/success probabilities are added to give the total probability of failure/success. Generally, the expression for success up to the end of a phase is simpler than the expression for its failure. This can be exploited to give a more efficient way of finding the exact conditional phase failure probability using

$$\begin{aligned} Pr(Ph_i) &= 1 - Pr\left(\overline{Ph_i}\middle|\overline{Ph_{i-1}}\right) \\ &= 1 - \frac{Pr\left(\overline{Ph_{0,i}}\right)}{Pr\left(\overline{Ph_{0,i-1}}\right)} \end{aligned} \qquad (13)$$

Paths through the BDDs to ‘0’ terminal nodes (i.e., paths to success) are obtained by applying a modified version of the BDD solutions algorithm developed by Rauzy (1993) (Fig. 12).



```
solutions $(F, \sigma)$ =
        if $(F = 0)$ return $\phi$
        else if $(F = 1)$ return $\{\sigma\}$
        else $/* F =$ ite$(x, F1, F2)$ $*/$
            $S \leftarrow$ solutions $(F1, \sigma \cup \{x\})$,
            $T \leftarrow$ solutions $(F2, \sigma \cup \{\bar{x}\})$
            return $S \cup T$
```

Figure 12 – Modified BDD solutions algorithm for obtaining BDD paths including success events

The modification allows success event information to be retained in the output paths produced by the algorithm. Once all paths through the BDDs have been obtained the following Boolean phase algebra rules (La Band & Andrews, 2003) are applied to remove any impossible paths (i.e., combinations of events that could never occur together), redundancies and repetition of events,

Redundant terms (repetition of events)	$A_i.A_i = A_i$	(14)
Cannot fail in more than one phase	$A_i.A_j = 0$	(15)
Redundant terms	$A_i.A_{ij} = A_i$	(16)
Cannot be failed and working at the same time	$\bar{A_\iota}.A_i = 0$	(17)
Success in phase *i* rules out failure in that phase	$\bar{A_\iota}.A_{ij} = A_{i+1,j}$	(18)
Combination of success events across successive phases	$\bar{A_\iota}\overline{A_{\iota+1}}\dots\bar{A_\jmath} = \overline{A_{\iota\jmath}}$	(19)
Combination of failure events across successive phases	$A_i + A_{i+1} + \dots + A_j = A_{ij}$	(20)

The probability of occurrence, Q_i, or non-occurrence, R_i, is then calculated by summing the probabilities of the disjoint paths through the BDD. The evaluation of the failure or success probabilities of each basic event can be determined using

$$r_{A_{i,j}} = 1 - q_{A_{i,j}} \tag{21}$$

$$q_{A_{i,j}} = q_A\left(t_{i-1}, t_j\right) = \int_{t_{i-1}}^{t_j} f_A\left(t \text{ d}t = F_A\left(t_j\right) - F_A\left(t_{i-1}\right)\right), \tag{22}$$

where $r_{A_{i,j}}$ is the is the probability of success of basic event A in all phases from i to j, $q_{A_{i,j}}$ is the probability of failure of basic event A in any phase from i to j, t_{i-1} the time of initiation of phase i, $f_A(t)$ the probability density function of failure of component A with respect to t, $F_A(t)$ the cumulative probability function of failure of component A with respect to t.

Once mission phase failure probabilities have been calculated for each phase, they can then be added to give the total failure probability for the entire mission,

$$Q_{MISS} = \sum_{i=1}^{n} Q_i \tag{23}$$

where n is the total number of phases that make up the mission.

When calculating mission phase failure/success probabilities, phase criticality functions can also be calculated by setting $q_{A_{i,j}} = 0$ and $q_{A_{i,j}} = 1$ for individual components instead of using Eq. (21). Eq. (4) is then used to determine the system unconditional failure intensity.

5. Conclusions

An overview of a new methodology for risk assessment has been presented with a selection of features illustrated through its application to classic FT/ET and phased mission-based case studies. By incorporating Petri net and/or Markov model-based techniques the methodology offers enhanced risk modelling capabilities compared to classic FT/ET based approaches. Such enhancements include failure rate dependencies, non-constant failure rates, common cause failures, and quantification of the effects of complex degradation and maintenance processes. The new methodology utilises BDD techniques for efficient analysis of user inputs in the form of traditional FT/ET structures.

Acknowledgements

This work was supported by the Lloyd's Register Foundation, a charitable foundation in the U.K. helping to protect life and property by supporting engineering-related education, public engagement, and the application of research.

References

Chew, S. P., Dunnett, S. J., & Andrews, J. D. (2008). Phased mission modelling of systems with maintenance-free operating periods using simulated Petri nets. *Reliability Engineering & System Safety*, 93, 980-994.

Andrews, J. D., Prescott, D., & De Rozières, F. (2014). A stochastic model for railway track asset management. *Reliability Engineering & System Safety*, 130, 76-84.

Bryant, L., Andrews, J. D., & Fecarotti, C. (2017). A Petri net model for railway bridge maintenance. *Proc. IMechE Part O: J. Risk and Reliability*, 1-18.

Bouissou, M., & Bon, J. L. (2003). A new formalism that combines advantages of fault-trees and Markov models: Boolean logic driven Markov processes. *Reliability Engineering & System Safety*, 82(2), 149-163.

Bäckstrom, O., Butkova, Y., Hermanns, H., Krčál, J., & Krčál, P. (2016). Effective static and dynamic fault tree analysis. *Proceedings of the 35th international conference SAFECOMP 2016, 266-280.*

La Band, R. A., & Andrews, J. D. (2003). Phased mission modelling using fault tree analysis. *Proceedings of the 15th advances in reliability technology symposium (ARTS), Mechanical Engineering Publications for IMechE, 2003, 81-97.*

Rauzy, A. (1993). New algorithms for fault trees analysis. *Reliability Engineering & System Safety*, *40*(3), 203-211.

Sinnamon, R. M., & Andrews, J. D. (1997). Improved efficiency in qualitative fault tree analysis. *Quality and Reliability Engineering International*, *13*(5), 293-298.