Improved Reliability modelling using Bayesian Networks and dynamic discretisation

David Marquez, Martin Neil and Norman Fenton
Department of Computer Science, Queen Mary, University of London

Abstract

This paper shows how recent revolutionary Bayesian Network (BN) algorithms can be used to model very complex reliability problems in a simple unified way. The algorithms work for so-called hybrid BNs, which are BNs that can contain a mixture of both discrete and continuous variables. Such hybrid BNs enable us to model failure times and reliability together. The approach allows a compact representation of the event-dependent failure behaviours characteristic of fault-tolerant systems, avoiding the state space explosion problem of the Markov Chain based approaches. The BN framework presented is able to solve any configuration of static and dynamic gates with general time-to-failure distributions, without using numerical integration techniques or simulation methods. Unlike other approaches (which tend to be restricted to using exponential distributions) we can use as input any parametric or empirical failure rate distribution. The approach offers a powerful framework for analysts and decision makers to successfully perform robust reliability assessment. Sensitivity, uncertainty, diagnosis analysis, common cause failures, and warranty analysis can also be easily performed within this framework.

Keywords: Bayesian networks; Systems Reliability; Dependability; Dynamic Fault Trees; Dynamic discretisation.

1. Introduction

Most published reliability analysis methods are based on parametric and non-parametric statistical models of time-to-failure data and its associated metrics [38]. The underlying assumption of these methods is that a coherent, statistical model of system failure time can be developed that will prove stable enough to accurately predict a system's behaviour over its lifetime. However, given the increasing complexity of the component dependencies and failure behaviours of today's real-time safety-critical systems, the statistical models may not be feasible to build or computationally tractable. This has led to an increasing interest in more flexible modelling frameworks for reliability analysis. The most notable such frameworks are combinatorial models such as fault trees (FTs), space state based approaches such asdynamic fault trees (DFTs), which we describe in Section 2.1; and Bayesian Networks, which we describe in Sections 2.2 and 2.3.

While the DFT approach is very flexible, in practice it has severe limitations, such as the problem of state based explosion and the inability to handle non-standard statistical distributions. To date the Bayesian Network (BN) framework has only partially addressed these limitations.

The main characteristic of real real-world applications is the combination of discrete and continuous components with static and time-dependent life distributions. Previous attempts to apply BN models to reliability assessment have not adequately handled the necessary 'hybrid' models required, i.e. models containing both continuous and discrete variables, with non-Gaussian distributions.

In this paper we present (in Section 3) a simple event-based hybrid BN modelling method for reliability assessment that scales up to large, complex dynamic systems, and overcomes the limitations of both dynamic fault trees and previous BN approaches. The new approach incorporates a recent powerful approximate inference algorithm for hybrid BNs, based on a process of dynamic discretisation of the domain of all continuous variables in the BN.

The significant novel research contributions provided in this work are:

- Modelling of very complex reliability problems in a simple unified way based on a single revolutionary algorithm.
- Modelling system state and failure times together (because of the ability to combine discrete and continuous nodes in the BN model)
- Overcoming most of the complexity problems inherent to space-state based reliability models
- Solving any configuration of static and dynamic gates with general time-to-failure distributions, without using numerical integration techniques or simulation methods
- Producing Time To Failure distributions for all gates and models.
- Ability to use as input any parametric or empirical failure rate distribution (so unlike previous approaches we are not: restricted to Exponential).

The power and flexibility of the approach is demonstrated (in Sections 3 and 4) by comparing the results with traditional space-state approaches, like DFTs, used in a number of popular reliability tools. We test the accuracy of the algorithm on a range of classical dynamic fault trees constructs, allowing the system components to adopt any time to failure distribution occurring in practical applications. In each case we compare the results with the analytical solution of the Markov chain representation or the approximated solutions generated by numerical integration schemes, as appropriate. The results are very close to the analytic solutions and are achieved with much less effort. In several cases our approach provides predictions of situations that simply cannot be modelled by the alternative approaches.

All the example models shown in this paper are built and executed using the commercial general-purpose Bayesian Network software tool AgenaRisk [1], in which our dynamic discretisation algorithm is now implemented.

Glossary of terms used

BDD	Bayesian Decision Diagram
BN	Bayesian Network
CPD	Conditional Probability Distribution
CPU	Central Processing Unit
CSP	Cold Spare Gate
DBN	Dynamic Bayesian Network
DFT	Dynamic Fault Tree
FDEP	Functional Dependency Gate
FFT	Fast Fourier Transform
FT	Fault Tree
HCAS	Hypothetical Cardiac Assist System
HSP	Hot Spare Gate
JT	Junction Tree
KL	Kuback-Leibler
MC	Markov Chain
MCMC	Markov Chain Monte Carlo
MTE	Mixtures of Truncated Exponential
MTTF	Mean Time To Failure

NPT	Node Probability Table
PAND	Priority AND
PDEP	Probability Dependency Gate
SEQ	Sequential Enforcing Gate
WSP	Warm Spare Gate

2. Background

This section reviews the most relevant previous work. In Section 2.1 we review the work on reliability modelling that has evolved from FT and DFT type analysis. This includes Monte Carlo methods. The rest of the section covers BNs. A brief overview of BNs is presented in Section 2.2 and a review of previous BN work in reliability modelling is presented in Section 2.3.

2.1. Reliability modelling using Fault Tree and Markov chain based approaches

The most popular methods for addressing the kind of complex reliability analysis problem described in the Introduction are based on two main frameworks.

1. Combinatorial models, like Static FTs; and
2. State-space models, like DFTs

In the Static FT framework Boolean constructs are used to model how combinations of components' failures can cause the failure of subsystems or of the whole system [39], [45]. Efficient qualitative and quantitative analysis of FTs can be performed using Binary Decision Diagrams (BDD) [7], [9].

The problem with static FTs is that they cannot capture complex event-dependent behaviours (sequence-dependent failures, functional dependencies, and stand-by spares) of fault-tolerant systems. This is the problem that the state-space models, like DFTs [10], [11] were developed primarily to address. DFTs have increased the modelling power of FTs by taking into account not only the combinations but also the sequential ordering of occurrence of component failures' that led to system failure. Analytical solutions of DFTs are obtained by automatic conversion to the equivalent continuous time Markov process, with state-space given by the combination of occurrence of all possible events, and transition probabilities characterised by the components' hazard rates [10].

While the DFT approach will model nearly any sequence-dependent system, representing dynamic tree model failures as states of a Markov process is a daunting, error-prone task that also has two major limitations:

1. The state-space generated grows exponentially with the size of the system.
2. There are limitations on the modelling of spares (such as warm or cold spares with non-exponential time-to-failure distributions)

Several methods have been developed to deal with these limitations. For the first limitation, modularisation algorithms have been introduced to help to break down the size of large systems into smaller independent subtrees that do not share basic events. These subtrees are then solved separately using a suitable technique according to its classification as static (Boolean) or dynamic ([11], [12], [18]). However, if the top-level gate of the fault tree is dynamic, the modularisation technique cannot be applied since it does not provide an exact solution ([11]). Solving several Markov processes corresponding to the independent modules is computationally more efficient than solving the single Markov model for the entire system fault tree. However, the state space explosion and resulting computational complexity remains a major limitation in using the Markov representation of DFTs, as even a relatively simple DFT can give rise to a very large Markov model.

In order to overcome the second limitation, numerical integration methods have been used to obtain approximate solutions of DFTs without converting them to a Markov model [3]. Alternatively, Monte-Carlo simulation techniques have been adopted to solve DFTs. These approaches have extended the modelling capabilities of DFTs by allowing the inclusion of spares components with general time-to-failure distributions (Lognormal, Weibull), previously not feasible using Markov-based techniques [11], [12]. However, the well-known trade-off between long computational times and high accuracy represents a major drawback in the use of simulation and the models are overly complex.

2.2. Bayesian Networks overview

A BN ([19], 0, [36]) consists of two main elements.

1. *Qualitative*: This is given by a directed acyclic graph (DAG), with nodes representing random variables, which can be discrete or continuous, and may or may not be observable, and directed arcs (from parent to child) representing causal or influential relationship between variables.
2. *Quantitative*: Conditional Probability Distributions (CPDs) that define the probabilistic relationship of each node given its respective parents. Nodes without parents, called root nodes, are described according to their marginal probability distributions.

Together, the qualitative and quantitative parts of the BN encode all relevant information contained in a full probability model. The conditional independence assertions about the variables, represented by the lack of arcs, allow decomposition of the underlying joint probability distribution as a product of CPDs. This significantly reduces the complexity of inference tasks on the BN [25], [43]. If the variables are discrete, the CPDs can be represented as Node Probability Tables (NPTs), which list the probability that the child node takes on each of its different values for each combination of values of its parents.

BNs are widely recognised as being an effective and robust decision support framework for problems involving uncertainty and probabilistic reasoning. BNs are mathematically sound and at the same time flexible and simple enough to allow the interaction with domain experts and decision makers. The graphical view, as presented in tools like AgenaRisk [1], is especially powerful. BNs enable us to express our complete state of knowledge about a problem and handle the associated uncertainties. The BN formalism offers powerful algorithms for both predictive (cause to effect) type reasoning and diagnostic (effect to cause) type reasoning. So, for example, predictive reasoning enables us to calculate the reliability of any part of a system based on the prior probability distributions of the basic components and the conditional dependence assertions. In diagnostic reasoning we can perform statistical inference, i.e. revising prior probabilities in the light of actual observations of events.

One of the most important benefits of BNs is that they enable us to integrate information from different sources, including experimental data, historical data, and prior expert opinion. We have applied BNs to a range of real-world dependability-type problems [31], [32], [33], [35] and have convincingly demonstrated both the feasibility and usefulness of the technology. In the area of software system reliability, we have shown the advantages of BNs over traditional methods for predictive and diagnostic reasoning [13], [14], [15], [16].

To incorporate the time dimension into BNs so that dynamic systems can also be modelled, different extensions have been suggested, mainly by adding a direct mechanism for representing temporal dependencies among the variables [2], [17], [29].

2.3. BNs in reliability analysis

A number of recent studies have attempted to use BNs to provide a unified framework for reliability modelling and analysis of complex systems. In particular, BNs have been used as an alternative representation to combinatorial based models like reliability block diagrams and (static) FTs. BNs have been shown to increase both the modelling capabilities and analysis power by including new modelling features - like multi-state variables, noisy gates, common cause failures, and simple sequentially dependent failures - and general a posteriori diagnosis analysis [4], [24], [37], [39], [44].

In space-state based reliability models, like DFTs, two different formalisms have been adopted to represent temporal (and functional) dependencies among the system components:

1. The *time-slice* approach; and
2. The *event-based* approach

The *time-slice* approach uses a Dynamic Bayesian Network (DBN) to perform DFT analysis [27] [46]. A DBN consists of a sequence of identical BNs indexed by the discretised time line. Each time-indexed BN represents a snapshot of the corresponding DFT at a particular point in time (time slice). In particular, each time slice contains a set of time-indexed discrete random variables representing the state of the associated DFT components at a particular time instance. Directed arcs between nodes within a slice represent 'instantaneous' causal or statistical relationships, whereas directed arcs interconnecting nodes from different time slices reflect temporal dependency between the variables. Although this approach allows modelling many complex configurations, there are some drawbacks:

1. Computing and filling the intra-slice conditional and marginal multidimensional NPT, and the inter-slice transition probabilities can be a tedious job, especially in the case where the components' time-to-failure distributions are non-exponential.
2. As pointed out in [27], the DBN formalism corresponds to a discrete time Markov chain representation of the DFT, which differs from the usual continuous time Markov chain adopted in reliability analysis. As a consequence, some assumptions have to be made in the discrete approach regarding contemporary faults, which are not allowed in a continuous time model.

In this paper we adopt the *event-based* approach, in which the DFT is translated into an equivalent BN with continuous random variables representing the time-to-failure of the components of the system. These can be either the time-to failure of elementary components of the system, or the time-to-failure of the fault tree constructs. In the latter case, the nodes in the BN are connected by means of incoming arcs to several components' time-to-failures and are defined as deterministic functions of the corresponding input components time-to-failure as shown in Figure 1**Error! Reference source not found.**.

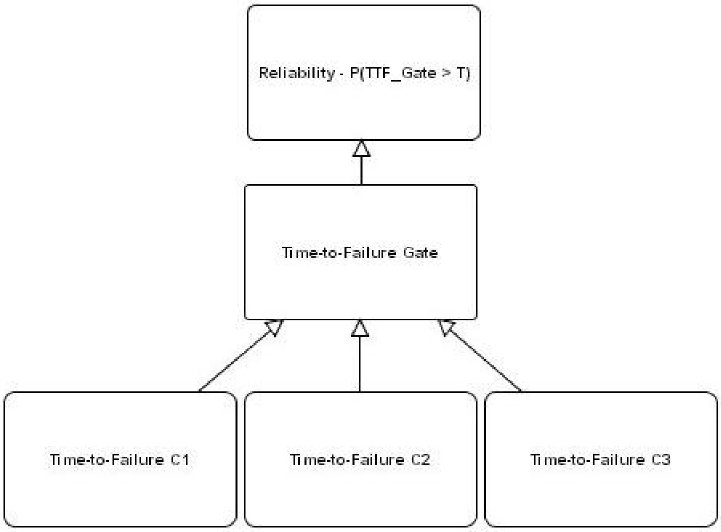

Figure 1– An event-based reliability BN example <NEED TO REPLACE DIGRAM WITH ONE CONTAINING THE MATHS>

In order to specify the probability distribution of the BN, we must give the marginal probability density functions of all root nodes and the CPDs of all non-root nodes. If the time-to-failure nodes corresponding to elementary components of the system (or some subsystem) are assumed statistically independent, (as is the case in static FT analysis, or dynamic gates with independent inputs), these are characterised by their marginal probability distributions. The marginal time-to-failure distributions of the root nodes are generally given by standard parametric probability density functions. The values of the parameters of these density functions can be either obtained as prior information according to expert knowledge, or estimated in a previous reliability data analysis step if some failure data is available (See 0 for an example of parameter learning in BNs).

The CPDs for both static and dynamic gates are probability distributions of variables that are a deterministic function of its parents, and are determined according to the types of constructs of the corresponding DFT. For some simple configurations, such as static gates or dynamic gates with exponential time-to-failure components distributions, an exact closed-form analytical expression can be derived for the CPDs. However, for general components' failure distributions, a closed-form expression for the CPDs of dynamic gates may not be feasible. In this case, numerical approximation methods need to be applied, as we shall show in Section3. The BN structure is created in a modular way by combining a predefined set of BN substructures designed to capture the failure mechanisms of the different (existing or new) DFT constructs. This approach is limited to events that can happen at most once (e.g., component failures of a non-repairable system).

A full discretised version of the event-based BN reliability model is given in [5], where the time line is partitioned into a finite number of time intervals. In this case, the nodes states are interpreted as the time intervals at which the failure of the corresponding system component can take place. The advantage of the discretised space state approach is that exact inference can be carry out on the resulting discrete BN using standard propagation algorithms [20], [26], [41]. However, despite its simplicity, this approach presents some drawbacks:

1. In order to model temporal ordering among a set of nodes, all nodes need to have the same time granularity n.

2. For large values of n, computing and filling the corresponding multidimensional NPTs can be a tedious or even unfeasible job, especially in the case where the components' time-to-failure distributions are non-exponential.
3. As with any static discretisation approach, there is a trade-off between long computational times and high accuracy controlled by the time granularity n.

A full continuous version of the event-based BN reliability model is shown in [6], where the time-to-failure density functions for several static and dynamic gates are expressed in terms of the unit-step and Dirac delta functions. Again, exact closed-form analytical expressions are only possible for some simple configurations.

In conclusion, despite the advances summarised above, the application of BNs as mainstream technology for reliability modelling problems remains modest. In addition to the problems already highlighted, a major impediment to progress has been the limitations of the previous generation BN tools: Designing, implementing and applying BNs in reliability analysis using these tools and technology was near impossible, especially where we need to incorporate continuous variables and inference from large data sets. In what follows we explain how we have solved most of these problems.

3. The new BN approach to reliability modelling

Our approach to solving DFTs uses a general framework, where continuous nodes represent the failure time of the system components and constructs, and discrete random variables are also included in the model to represent the state of the system (or any subsystem) at a particular time instance. Specifically, if the continuous random variable t_C represents the time-to-failure of a component (system, subsystem) *C*, then, a discrete child node *C*, with an incoming arc from t_C, may be included in the model to represents the state of the component. The NPT for the discrete node *C*, which define the probability distribution of the component states at a given time *t*, can be automatically computed from the component time-to-failure distribution (e.g., $P_t\left(C = fail\right) = P\left(t_C \leq t\right)$). The resulting model (Figure 1**Error! Reference source not found.**) is a hybrid BN containing both continuous as well as discrete variables, with general static and time-dependent failure distributions.

Unfortunately, for hybrid BNs containing mixtures of discrete and continuous nodes with non-Gaussian distributions, exact inference becomes computationally intractable. The traditional approach to handling (non-Gaussian) continuous nodes is static: you have to discretise them using some pre-defined range and intervals. However, this approach is unacceptable for critical type systems where there is a demand for reasonable accuracy. To overcome this problem we have developed a new and powerful approximate algorithm for performing inference in hybrid BNs. We use a process of dynamic discretisation of the domain of all continuous variables in the BN and using entropy error as the basis for approximation. This dynamic discretisation technique is described in the next section.

Several alternatives to discretisation have been suggested for approximate inference on hybrid BNs. Those include the use of mixtures of truncated exponential (MTE) distributions to approximate the nodes distributions [8], [28], combinations of MTE approximations with direct sampling methods [21], variational methods [30], and Markov Chain Monte Carlo (MCMC) approaches [39].

3.1. Dynamic Discretisation

Let t be a continuous random node in the BN. The range of t is denoted by Ω_t, and the probability density function (PDF) of t, with support Ω_t, is denoted by f_t. The idea of discretisation is to approximate f_t by (1) Partition Ω_t into a set of interval $\Psi_t = \{w_j\}$, and (2) Define a locally constant function $\tilde{f}_t$ on the partitioning intervals.

As in [22], we estimate the relative entropy error induced by the discretised function using an upper bound of the Kullback-Leibler (KL) metric between two density functions f and g: $D(f \| g) = \int f(x)\log\left(f(x)/g(x)\right)dx$. Under the KL metric, the optimal value for the discretised function $\tilde{f}_t$ is given by the mean of the function f_t in each of the intervals of the discretised domain. The discretisation task reduces then to finding an optimal partition set $\hat{\Psi}_t$.

Our approach to dynamic discretisation searches Ω_t for the most accurate specification of the high-density regions given the model and the evidence, calculating a sequence of discretisation intervals in Ω_t iteratively. At each stage in the iterative process, a candidate discretisation, $\Psi_t = \{w_j\}$, is tested to determine whether the relative entropy error of the resulting discretised probability density $\tilde{f}_t$ is below a given threshold, defined according to the trade off between the acceptable degree of precision and computation time.

The dynamic discretisation approach allows more accuracy in the regions that matter and incur less storage space over static discretisations. Moreover, we can adjust the discretisation any time in response to new evidence to achieve greater accuracy. By efficiently integrating our iterative approximation scheme within existing robust propagation algorithms on BN architectures, such as Junction Tree (JT) [20], we are able to perform robust inference analysis on complex systems. A detailed description of the dynamic discretisation algorithm is given in [34].

3.2. Estimating the CPD for DFT constructs

Once we have determined the marginal time-to failure distributions for the root nodes, the conditional probability distributions (CPDs) for the constructs need to be computed. In this the case, the conditional distributions involve a deterministic function of the input random variables. In general, calculating the probability distribution of variables that are a deterministic function of its parents represents a major challenge for most BN software. For standard continuous and discrete density functions this does not represent a problem but for more complex conditional distributions approximation techniques need to be used.

A simple method for generating the local conditional probability table $p(X \mid pa\{X\})$ commonly used under the static discretisation approach proceeds by first sampling values from each parent interval in $\Omega_{pa\{X\}}$ for all parents of X and calculating the result $X = f(pa\{X\})$, then counting the frequencies with which the results fall within the static bins predefined for X, and finally normalising the NPT.

Although simple this procedure is flawed. On the one hand, there is no guarantee that every bin in Ω_X will contain a probability density if the parents' node values are under sampled. The implication of this is that some regions of Ω_X might be void; they should have probability mass but *do not*. Any subsequent inference in the BN then will return an inconsistency when it encounters either a valid observation in a zero mass interval in X or attempts inference involving X. The only way to counter this under static discretisation is to generate a large number of samples, which is expensive and made more difficult by the fact that the sampling configuration settings in tools that use the static approach are inaccessible.

On the other hand, samples from each parent interval in $\Omega_{pa\{X\}}$ are usually taken uniformly such that at least two samples are taken for each interval in $\Omega_{pa\{X\}}$. As the number of parent nodes increases, and the states in Ω_X and $\Omega_{pa\{X\}}$ increases, the number of cells in the NPT, $p(X \mid pa\{X\})$, increases exponentially.

We resolve all deterministic functions by modelling them as an approximate mixture of Uniform distributions and use the dynamic discretisation algorithm to fit a histogram composed of Uniform distributions.

3.3. BN modelling of Boolean constructors

We now define the basic BN constructs (OR, AND, Voting OR) used in static FT analysis. Let us denote by t_i, $i = 1,..,n$, the time-to-failure of the i-th input component of the construct.

- *The AND gate*. In order for the output of an AND gate to fail, all input components of the gate must fail. So, if t_{AND} represents the time-to-failure of the output event, then, the probability of failure of the output of the AND gate in the time interval [0, t] is given by

$$\begin{aligned} P(t_{AND} \le t) &= P(t_1 \le t, \ldots, t_n \le t) \\ &= P\left(\max_i \{t_i\} \le t\right) \end{aligned} \qquad \textbf{Eq. 1}$$

  That is, the time-to-failure of the AND gate is a random variable defined as a function of its parents by $t_{AND} = \max_i \{t_i\}$.

- *The OR gate*. The output of the OR gate will fail if at least one of the input components of the gate fail, so

$$\begin{aligned} P(t_{OR} \le t) &= 1 - P(t_1 > t, \ldots, t_n > t) \\ &= 1 - P\left(\min_i t_i > t\right) \end{aligned} \qquad \textbf{Eq. 2}$$

  We then define the time-to-failure of the OR gate by $t_{OR} = \min_i \{t_i\}$.

- *The Voting OR*. If any k-out-of-n input components of a Voting OR fail the output will fail. Consider a Voting OR system consisting of three components such that *2*-out-of-*3* components are required for the system to operate. The output of the Voting OR gate will fail in the time interval [0, t] if

  (i) $\{t_1 \le t, t_2 \le t, t_3 > t\}$ or

(ii) $\{t_1 \leq t, t_2 > t, t_3 \leq t\}$ or

(iii) $\{t_1 > t, t_2 \leq t, t_3 \leq t\}$ or

(iv) $\{t_1 \leq t, t_2 \leq t, t_3 \leq t\}$

So, the time-to-failure of the Voting OR gate can be written as a combination of min and max of the components time to failure as

$$t_{VotingOR} = \min\left(\max\{t_1,t_2\},\max\{t_1,t_3\},\max\{t_2,t_3\},\max\{t_1,t_2,t_3\}\right) \qquad \textbf{Eq. 3}$$

We have defined the time-to-failure of the basic FT constructs as deterministic functions of its input components. The next step is to define the marginal time-to failure distributions for the basic components. In our framework, any parametric failure distribution can be used. The CPDs for the FT constructs are then estimated using approximation approach explained in Section 3.2.. Once this has been achieved, we can compute the reliability of the system for any mission duration, and other metrics of interest can also be automatically derived. These include the expected value of the failure density function, often designated as the Mean Time To Failure (MTTF), and the quantile of order a (0<a <1), representing a warranty period or time by which reliability will be equal to (1-a)%.

Example 1

Consider a system consisting of two components, C_1 and C_2, arranged in series (parallel). The corresponding FT consists of two basic inputs connected by an OR (respect. AND) gate. If we denote by t_C the continuous random variable representing the time-to-failure of component C_i, i = 1,2, then, in the BN framework, t_C and t_C are represented by root nodes connected to the basic gate node $t_{OR} = \min_i\{t_i\}$ (respect. $t_{AND} = \max_i\{t_i\}$). The BN models are depicted in Figure 2, with marginal and conditional distributions superimposed on the graph. In this example we assumed that C_1 is exponentially distributed with inverse scale (rate) parameter $l = 1/10.000$, and C_2 follows a Weibull distribution with shape (s) = 6 and inverse scale (b) = 1/10000. We also included in the model a binary random variable C = OR (AND), with an incoming arc from t_C, representing the state of the system at a mission time t = 7000 hours. The NPT for the discrete node C give use an estimate of the reliability of the system at a given time. This is computed from the component time-to-failure distribution by $P_t(C = \text{on}) = P(t_C > t)$. In our example, the reliability of the system at time t is given by

$$P_t(\text{OR\_gate} = \text{on}) = P(t_{OR} > t) = e^{-lt} \times e^{-(bt)^s}$$

$$P_t(\text{AND\_gate} = \text{on}) = P(t_{AND} > t) = 1 - (1 - e^{-lt}) \times (1 - e^{-(bt)^s})$$

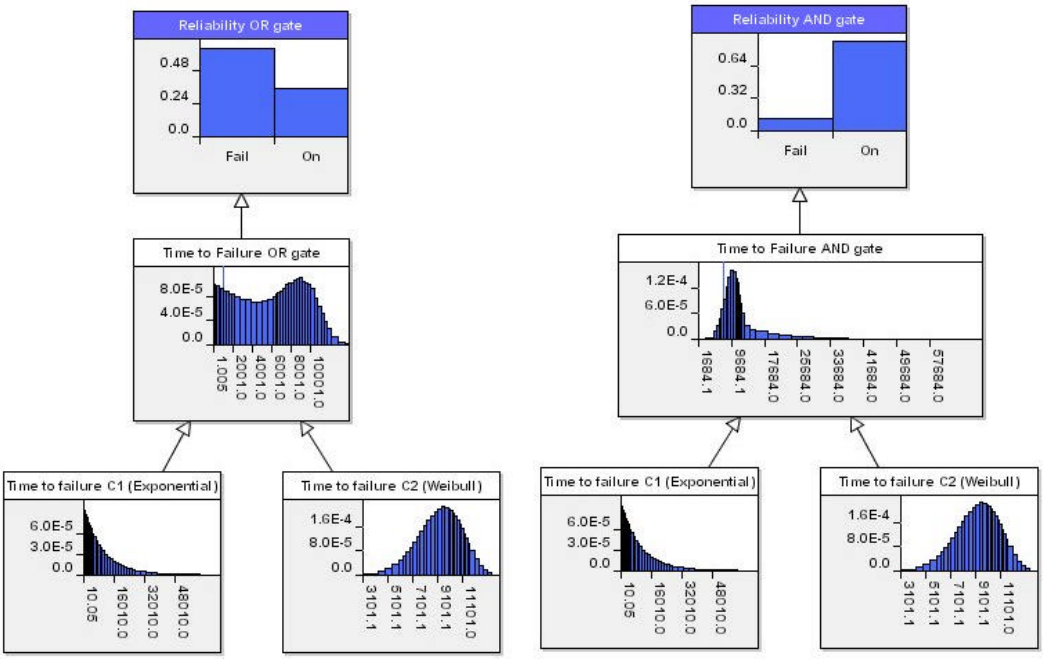

Figure 2 – BNs for OR (left) and AND (right) gates

Running the model for 40 iterations results in the summary values given in Example 1 of Table 1. This are compared with the analytical results for the reliability and estimates using numerical integration for the other metrics. The results are very close, and in the case of both the MTTF and the 90% warranty period of the AND gate, our approach produces predictions that are not possible analytically.

Example 2

Consider now a component (or system) that has already accumulated T_1 hours of operation. For general (non-exponential) failure density functions, the probability of failure in the next mission of duration T is dependent upon the prior operating time. This can be computed by

$$P\left(t_S \le T_1 + T \middle| P\left(t_S > T_1\right)\right) = \frac{\int_{T_1}^{T_1+T} f_{t_s}(t)\,dt}{\int_{T_1}^{\infty} f_{t_s}(t)\,dt} \qquad \textbf{Eq. 4}$$

We can modify the BN model to update the marginal failure density functions of a root component that has already accumulated some operation time. This is achieved by simply adding a binary child node to the aged components time-to-failure, with state values "yes" and "no", representing the statement that the component has or not already operated for a given period of time. Once we enter "yes" as evidence to the conditioning node, the backward (diagnostic) analysis capabilities of the BN allows revising the probabilities in the light of the new evidence. The resulting posterior failure density function represents the conditional time-to-failure density function for the component, given that it has already operated during a given period of time. Figure 3 shows the resulting posterior failure densities for a component C_2 with Weibull distribution with shape (s) = 6 and inverse scale (b) = 1/10000, that has already operated during 5,000 hours, together with the reliability of the component at a new mission time of T = 3,000 hours. Results together with the analytical values are shown in Table 1, Example 2.

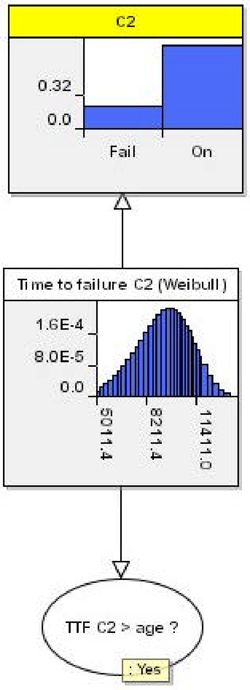

Figure 3 Conditional reliability

3.4. BN modelling of dynamic constructs

In the DFT formalism, new special purpose constructs (called dynamic gates) have been added to the traditional Boolean (AND, OR, M-out-of N) gates, in order to cope with many complex sequence dependencies among failure events or component states. In particular, the Spare gates model dynamic replacement of a failed main component from a pool of (independent) spares. The Sequence Enforcing gates (SEQ) model failures that occur only if others occur in certain order. SEQ gates are a special case of the Cold spare gate, so the same BN model can be use for both types. The Functional and Probability Dependency gates (FDEP and PDEP) model dependencies that propagate failure in one component to others. Finally, the Priority And gate (PAND) models situations where failures can occur only in a predefined order.

We now show the BN models for the Spare and PAND constructs. The SEQ gate is a special case of the Cold spare gate, and, for non-repairable systems with perfect coverage [10], the FDEP and PDEP gates can be modelled using OR gates, therefore these gates will not be considered in the following.

The SPARE gate. In a Spare or standby redundancy configuration, the spare components have two operation modes: a standby mode and an active mode. Each operation mode is represented by its failure distribution. A standby spare component becomes active (is called into service) when the current active component fails. A system in a spare configuration fails if all, main and spare, components fail. According to the standby mode failure distribution of the spare component, Spare gates are classified as: Hot, Warm, and Cold spares.

- *Hot spare (HSP) gate*. Both standby and active failure distributions of the spare are the same as the failure distribution of the main component. This is equivalent to a static AND gate.

- *Cold spare (CSP) gate*. The spare components never fail (the hazard rate is zero) when in standby mode. If t_{main} represents the time-to-failure of the main component and t_i is the time-to-failure of the i^{th} spare component when in active mode, then the probability that a system in a warm standby configuration fails in the time interval [0, *t*] is given by

$$\begin{gathered} P\left(t_{CSP} \le t\right) = P\left(t_{main} + t_1 + \cdots t_n \le t\right) \\ F_{t_{main}} * F_{t_1} * \cdots * F_{t_n}\left(t\right) \end{gathered} \qquad \textbf{Eq. 5}$$

That is, the failure distribution of the CSP gate is given by the convolution of the failure distributions of the main and active spare components. Calculating such a distribution represents a major challenge for most BN software. Traditional methods to obtain this function include Fast Fourier Transform (FFT) or Monte Carlo simulation. Here we use the estimation procedure explained in Section 3.2. using the mixture of Uniform and dynamic discretisation.

Example 3

Consider a cold spare configuration with two components. Both the main and active spare failure distributions are Weibull with shape (s) = 1.5 and inverse scale (b) = 1/1000. The BN model is depicted in Figure 4, with marginal and conditional distributions superimposed on the graph. The reliability of the system at a mission time *t* = 1000 hours is also shown in the graph. In this example, the reliability of the system at time *t* is given by (See Appendix)

$$P_t\left(\text{CSP\_gate} = \text{on}\right) = P\left(t_{CSP} > t\right) = \int_t^{\infty} e^{-(b\,u)^s} e^{-\left(b\,(t-u)\right)^s}\, du$$

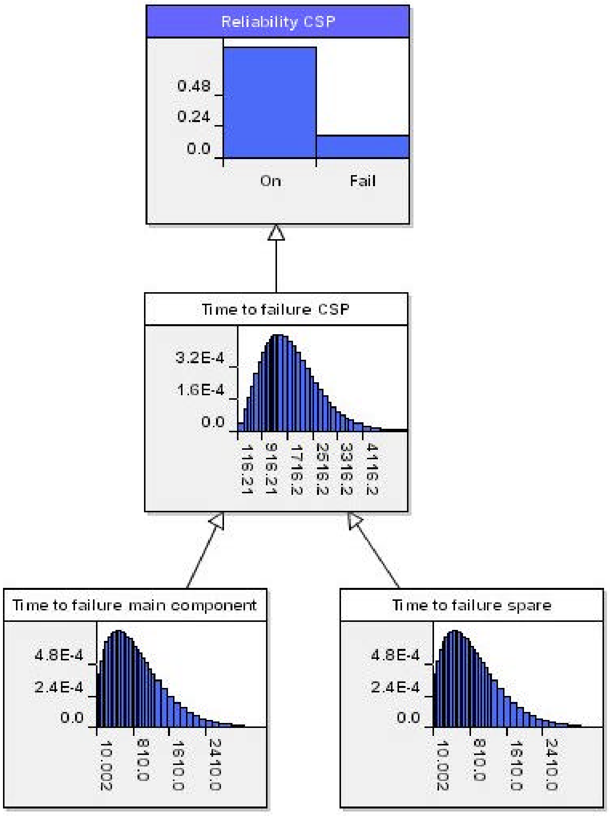

Figure 4 – BNs for Cold spare gate

Running the model for 40 iterations results in the summary values given in Table 1, example 3, and are compared with the estimates using numerical integration.

- *Warm spare (WSP) gate*. The hazard rate of the spare components is less in standby mode than in active mode. Consider a warm standby system consisting in one main component and one spare. Let us denote by t_{main} he time-to-failure of the main component, t_{standby}^{sb} the time-to-failure of the spare component when in standby mode, and t_{standby}^{act} is the time-to-failure of the spare component when in active mode. Then, the mutually exclusive events leading to the warm standby system fail in the time interval [0, *t*] are

  i) The spare component fails (when in standby mode) before the main component and the main component fails at time $t_1 < t$, or

  ii) The main component fails at time $t_1 < t$, the spare component is not failed at time t_1 (when in standby mode), and the spare component fails in the active mode during the time $t - t_1$

For spare components with constant hazard rate (exponential failure distribution), the above statements can be directly written in terms of the components time-to-failure by,

i) $\left\{t_{\text{standby}}^{sb} < t_{\text{main}}\right\} \& \left\{t_{\text{main}} < t\right\}$

ii) $\left\{t_{\text{standby}}^{sb} > t_{\text{main}}\right\} \& \left\{t_{\text{main}} < t\right\} \& \left\{t_{\text{standby}}^{act} < t - t_{\text{main}}\right\}$

From the above expression, we can derive the failure distribution of the WSP gate with exponential failure distribution by

$$t_{WSP} = \begin{cases} t_{\text{main}} & \text{if } t^{sb}_{\text{standby}} < t_{\text{main}} \\ t_{\text{main}} + t^{act}_{\text{standby}} & \text{if } t^{sb}_{\text{standby}} > t_{\text{main}} \end{cases} \qquad \textbf{Eq. 6}$$

Example 4

Consider a system with two components in a warm standby configuration. The main component has an exponential failure distribution with rate $l_1 = 0.001$. The spare component also has exponential failure distribution with rate $l_2 = 0.0005$ and $l_3 = 0.001$ when is in standby and active mode respectively. The BN model is depicted in Figure 5, with marginal and conditional distributions superimposed on the graph. The reliability of the system at a mission time *t* = 800 hours is also shown in the graph. In this example, the reliability of the system by time *t* is given by (See Appendix)

$$\begin{aligned} P\left(t_{WSP} > t\right) &= \int_t^{\infty} e^{-l_1 u}\,du + \int_0^t l_1 e^{-l_1 u} e^{-l_2 u} e^{-l_3(t-u)}\,du \\ &= e^{-l_1 t} + \frac{l_1}{l_1 + l_2 - l_3}\left[e^{-l_3 t} - e^{-(l_1+l_2)t}\right] \end{aligned}$$

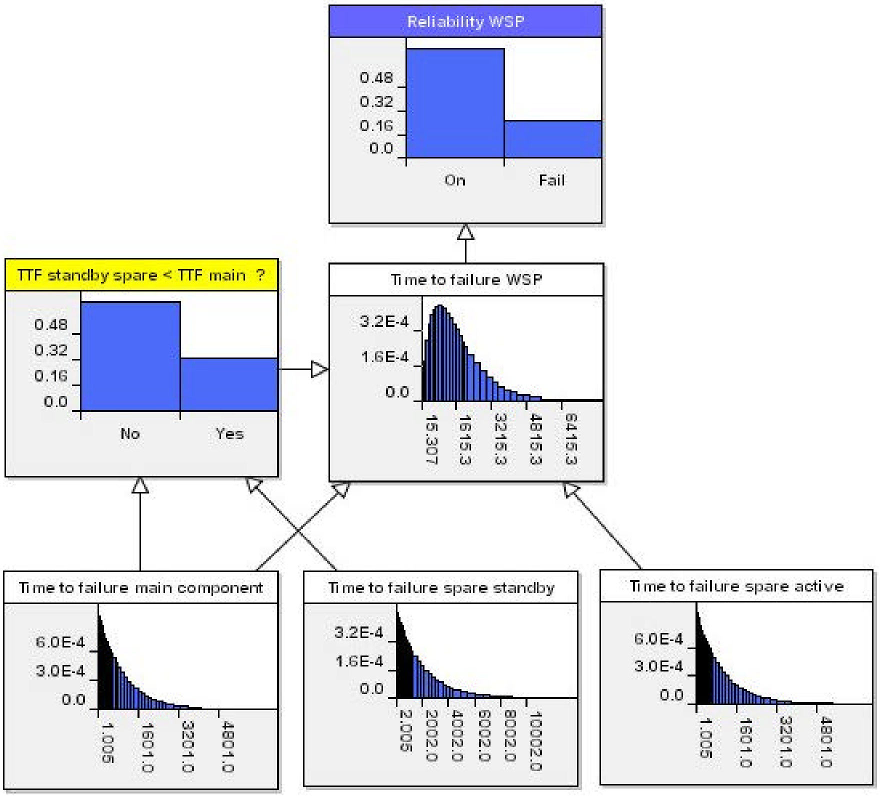

Figure 5 – BNs for Warm spare gate (exponential case)

Running the model for 40 iterations results in the summary values given in Table 1, Example 4 and are compared with the analytical solutions.

The Equation 5 for the failure distribution of the WSP gate is no longer valid if the failure distribution of the spare component is not exponential. As stated previously, for components that have already accumulated some operation time in its wearout region (non-exponential failure distribution), the probability of failure during the next mission time depends upon the prior operating time. In order to compute the probability that the spare component fails in the active mode during the time $t - t_1$, in a system with non-exponential warm standby configuration, we need to compute the equivalent operating time t_e for the spare component, had it be operating in its active mode since the start of the mission. This is because, when operating in its wearout region, $P\left(0 < t_{\text{standby}}^{act} < t\right) < P\left(t_e < t_{\text{standby}}^{act} < t_e + t\right)$.

If we modify the Equation 5 to include the accumulated operation time t_e of the spare component when it becomes active, had it been operating in the active mode since the start of the mission, we obtain the following expression for failure distribution of the WSP gate

$$t_{WSP} = \begin{cases} t_{\text{main}} & \text{if } t_{\text{standby}}^{sb} < t_{\text{main}} \\ \left(t_{\text{main}} + t_{\text{standby}}^{act} - t_e \middle| t_{\text{standby}}^{act} > t_e\right) & \text{if } t_{\text{standby}}^{sb} > t_{\text{main}} \end{cases} \qquad \textbf{Eq. 7}$$

Example 5

Consider a system with two components in a warm standby configuration. The main component has Weibull failure distribution with shape (s_1) = 1.5 and inverse scale (b_1) = 1/1500. When operating in the active mode, the spare component also has Weibull failure distribution with shape (s_2) = 1.5 and inverse scale (b_2) = 1/1500. When operating in the standby mode, the spare failure distribution is Weibull with shape (s_3) = 1.5 and inverse scale (b_3) = 1/2000.

We can modify the BN model shown in Figure 5 to account for the accumulated operation time of the spare component when it becomes active. As before, this is achieved by simply adding a binary child node to spare time-to-failure, with state values “yes” and “no”, representing the statement that the component has or not already operated for a given period of time, and entering “yes” as evidence. The resulting posterior failure density function represents the conditional active time-to-failure distribution for the spare component, given that it has already operated during a period of time equivalent to the period of operation in standby mode. The resulting BN is depicted in Figure 6, with marginal and conditional distributions superimposed on the graph. The reliability of the system at a mission time t = 1000 hours is also shown in the graph. In this example, the reliability of the system by time t is given by (See Appendix)

$$P\left(t_{WSP} > t\right) = e^{-\left(b_1 t\right)^{s_1}} + b_1^{s_1} \int_0^t u^{s_1 - 1} e^{-\left(b_1 u\right)^{s_1}} e^{-\left(b_2 u\right)^{s_2}} \frac{e^{-\left(b_3 \left(t_e + t - u\right)\right)^{s_3}}}{e^{-\left(b_3 t_e\right)^{s_3}}} du$$

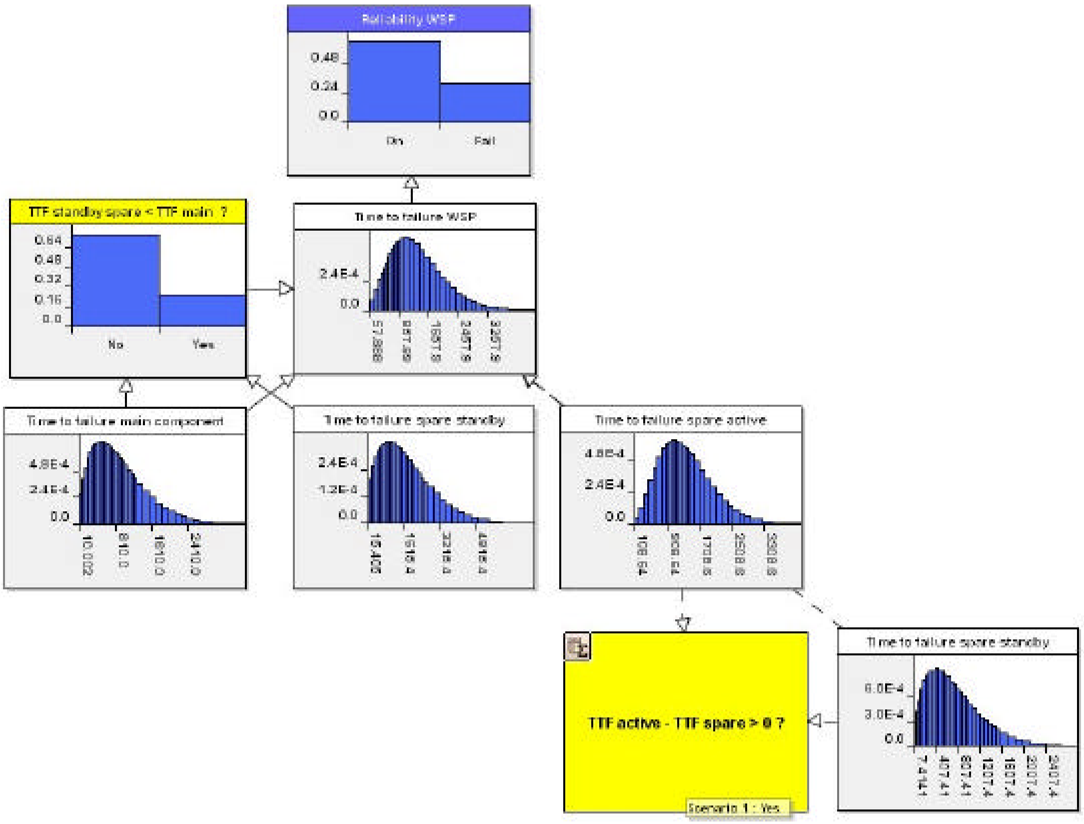

Figure 6 – BNs for Warm spare gate (non-exponential case)

Running the model for 40 iterations results in the summary values given in Table 1, Example 5 and are compared with the estimates using numerical integration.

- *The PAND gate*. The output of the PAND gate will fail if all of its input components fail in a predefined order (left to right). Consider a PAND system consisting in two components, and denote by t_i the time-to-failure of the i^{th} component, $i = 1,2$. If t_{AND} represents the time-to-failure of the output event, then, the probability of failure of the output of the PAND gate in the time interval [0, t] is given by

$$\begin{aligned} P(t_{PAND} \le t) &= P(t_1 \le t_2 \le t) \\ &= P(t_2 \le t | t_2 \ge t_1) P(t_2 \ge t_1) \end{aligned} \qquad \textbf{Eq. 8}$$

Example 6

Consider a PAND system consisting of two components, C_1 and C_2. The BN models are depicted in Figure 7, with marginal and conditional distributions superimposed on the graph. In this example we assumed that both components are exponentially distributed with rate parameters $l_1 = 0.002$ and $l_2 = 0.001$ respectively. The NPT for the discrete node gives an estimate of the reliability of the system at a given time. In our example, the reliability of the system at a mission time t = 1000 hours is given by

$$\begin{aligned} P(t_{WSP} > t) &= 1 - \int_0^t l_1 e^{-l_1 u} \left[e^{-l_2 u} - e^{-l_2 t} \right] du \\ &= 1 - \frac{l_1}{l_1 + l_2} \left[1 - e^{-(l_1 + l_2)t} \right] + e^{-l_2 t} \left[1 - e^{-l_1 t} \right] \end{aligned}$$

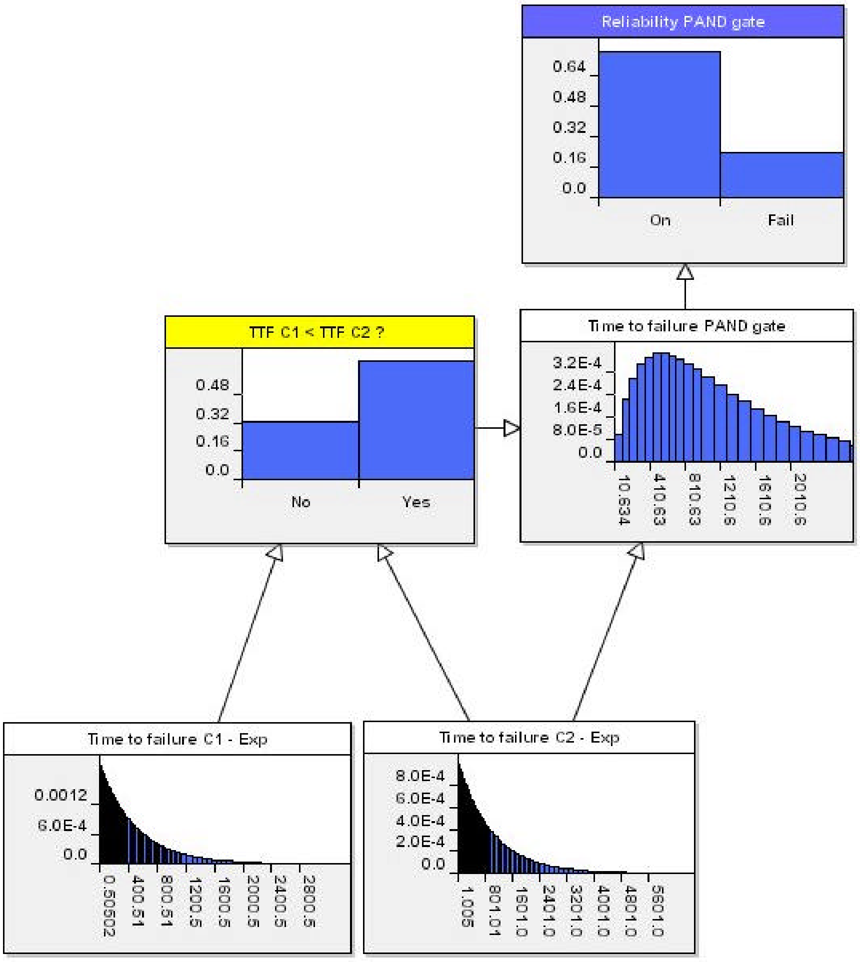

Figure 7 – BNs for PAND gate

Running the model for 40 iterations results in the summary values given in Table 1, Example 6 and are compared with the analytical results for the reliability and estimates using numerical integration for the other metrics.

4. Application example

We now show how our BN formalism can be used to perform DFT-like analysis of a real-world system. The example provided in this section is the CPU module of the Hypothetical Cardiac Assist System (HCAS), designed to treat mechanical and electrical failures of the heart. A detailed description of the system is given in [5], [6]. Figure 8 shows the corresponding BN model.

The BN model for the CPU module of the HCAS consists of three sub modules: a trigger (T), a WSP gate (CPU), and a FDEP gate (CPUT). The trigger consists of a crossbar switch (CS) and a system-supervision (SS). The CPU unit is a warm standby configuration with main component P and spare B. The CPU unit is also functionally dependent on the trigger component: the failure of either CS or SS causes the failure of the CPU unit. Thus, the CPUT node in the BN models the time to failure of the FDEP gate with trigger T and dependent event CPU. The failure distribution for all the components is exponential: CS, SS, and the standby mode spare B have failure rate $l_1 = 10^{-6}/\text{h}$, and P and the active mode spare B have failure rate $l_2 = 2\times10^{-6}/\text{h}$.

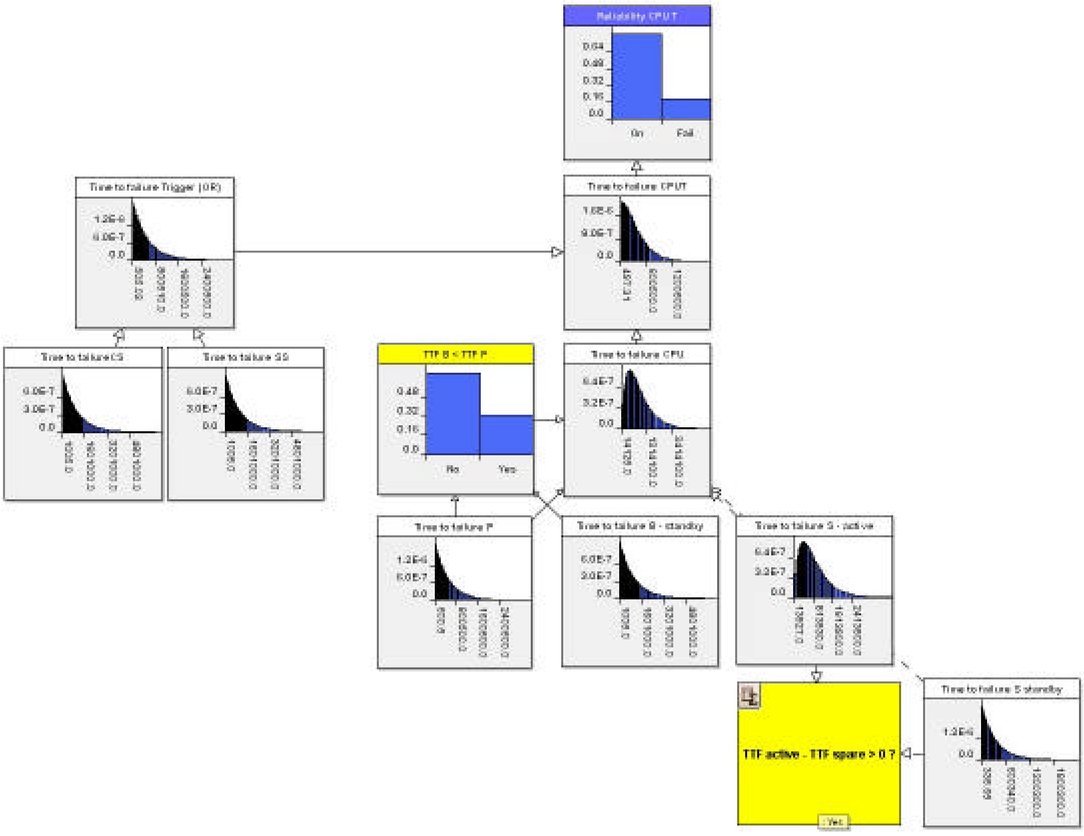

Figure 8 – BNs for CPU module of HCAS system

Once we have defined the marginal time-to failure distributions for the basic components, the CPDs for the DFT constructs are automatically estimated using the approximation approach and dynamic discretisation algorithm explained in Section 2 <AGAIN!!!>. No analytical calculation needs to be performed and no tables need to be filled. From the estimated failure distributions of the DFT constructs, we also obtain estimates for the reliability of the system for any mission time and other metrics of interest. Running the model for 40 iterations results in the summary values given in Table 1 Example 7 and are compared with the analytical results given in [6].

5. Conclusions

We have presented a new, effective and flexible event-based BN framework for system reliability modelling. By combining the modelling capabilities of BNs with our dynamic discretisation inference algorithm we offer a unified technique for reliability analysis of large, safety critical dynamic systems. Our BN framework is mathematically sound and at the same time simple enough to allow the interaction with domain experts and decision makers.

The modelling power of BNs provides a versatile high-level modelling tool to express complex components dependencies and different behavioural modes. It also overcomes most of the problems inherent in state-space based reliability models, like DFTs. In particular, it avoids the state-space explosion problem of the Markov Chain based approaches, and the limitation on the modelling of dynamic gates with general failure distributions. The modelling covers:

- OR, AND, Voting OR gates
- Warm standby
- Hot standby
- Cold standby
- Sequential failure gate
- Functional dependency gate
- Common cause failures
-
- Shared spares
- Priority AND gate

Within our framework, approximated solutions for both Boolean and dynamic constructs are obtained simultaneously. No modularisation method is necessary. Furthermore, by modelling the failure distributions of the DFT constructs as an approximate mixture of Uniform distributions and using the dynamic discretisation algorithm, our BN framework is able to solve any configuration of static and dynamic gates with general time-to-failure distributions, without using numerical integration techniques or simulation methods.

Another advantage of the combined effect of the event-based BN and dynamic discretisation algorithm is that the diagnostic analysis capabilities of the BN can be used to obtain estimates of the parameterised marginal failure distribution (for the root nodes), either using some available raw failure data or as prior information according to expert knowledge. No exact expression for the marginal is needed and no tables need to be filled.

In a range of examples we have demonstrated that our approach achieves results which are almost as good as analytical results, but with much less effort. Moreover, in many cases we were able to obtain results that cannot be computed analytically.

The approach offers a powerful framework for analysts and decision makers to successfully perform robust reliability assessment. Sensitivity, uncertainty, diagnosis analysis, common cause failures, and warranty analysis can also be easily performed within this framework.

Table 1: Summary Comparative Values (Analytical results in brackets)

Example 1

	Reliability at *t* = 7000	**MTTF**	**90% Warranty period**
OR gate	0.44149(0.4415)	5980.8(5978.9*)	t = 1054.9(1053.6*)
AND gate	0.94312(0.9441)	13319(13298.2*)	t = 7677.7(7687*)

Example 2

	Reliability at *t* = 3000
New component	0.9991(0.9993)
Aged component (5000 hours)	0.7910(0.7815)

Example 3

	Reliability at *t* = 1000	**MTTF**	**90% Warranty period**
Cold spare	0.8355(0.8212*)	1808.1(1805.2*)	t = 782.6(784.2*)

Example 4

	Reliability at *t* = 800	**MTTF**	**90% Warranty period**
Warm spare	0.7434(0.7456)	1681.2(1666.1)	t = 441.2(435.8*)

Example 5

	Reliability at *t* = 1000	**MTTF**	**90% Warranty period**
Warm spare	0.6807(0.7057*)	1378.1(1436.4*)	t = 543.2(621.5*)

Example 6

	Reliability at *t* = 800
PAND gate	0.7599 (0.7524)

Example 7

	Reliability at *t* = 10^5 hours	**MTTF**
HCAS_CPU	0.805(0.797)	351.470(350.000) hours

* Approximate results obtained by numerical methods using ReliaSoft

Appendix. Computing the reliability of standby redundant systems

An analytical expression for the reliability of standby redundant systems can be derived by adding the reliability associated with the mutually exclusive events leading to system success. For simplicity, let us consider a warm standby system consisting in one main component and one spare. Let us denote by t_{main} the time-to-failure of the main component, t_{standby}^{sb} the time-to-failure of the spare component when in standby mode, and t_{standby}^{act} is the time-to-failure of the spare component when in active mode.

For spare components with constant hazard rate (exponential failure distribution), the mutually exclusive events leading to the warm standby system success in the time interval [0, *t*] are

i) $\{t_{\text{main}} > t\}$ or

ii) $\{t_{\text{main}} = t_1 < t\} \, \& \left\{t^{sb}_{\text{standby}} > t_1\right\} \& \left\{t^{act}_{\text{standby}} > t - t_1\right\}$

Then, if R_1 and R_2 denote the reliabilities associated with events i) and ii) respectively, we obtain

$$\begin{aligned} R(t) &= R_1(t) + R_2(t) \\ &= \int_t^\infty f_{t_{\text{main}}}(y)\,dy + \int_0^t f_{t_{\text{main}}}(y) \int_y^\infty f_{t^{sb}_{\text{spare}}}(x) \int_{t-y}^\infty f_{t^{act}_{\text{spare}}}(z)\,dzdxdy \\ &= R_{\text{main}}(t) + \int_0^t f_{t_{\text{main}}}(y) R^{sb}_{\text{spare}}(y) R^{act}_{\text{spare}}(t-y)\,dy \end{aligned} \qquad \textbf{Eq. 9}$$

where, R_{main} denotes the reliability of the main component, R^{sb}_{spare} is the reliability of the spare when in standby mode, and , R^{act}_{spare} is the reliability of the spare when active.

If the failure distribution of the spare component is not exponential, we need to modify the Equation 9 to include the accumulated operation time t_e of the spare component when it becomes active, had it been operating in the active mode since the start of the mission. This is because, when operating in its wear-out region, $P\left(t^{act}_{\text{standby}} > t - t_1\right) > P\left(t^{act}_{\text{standby}} > t - t_1 + t_e \middle| t^{act}_{\text{standby}} > t_e\right)$. Therefore, for general (non-exponential) failure distribution, the reliability of the WSP gate is given by the following expression

$$R(t) = R_{\text{main}}(t) + \int_0^t f_{t_{\text{main}}}(y) R^{sb}_{\text{spare}}(y) \frac{R^{act}_{\text{spare}}(t_e + t - y)}{R^{act}_{\text{spare}}(t_e)}\,dy \qquad \textbf{Eq. 10}$$

Note that, in the case of hot spares, $R^{sb}_{\text{spare}} \equiv R^{act}_{\text{spare}}$ and $t_e = y$ for all $y \in [0, t]$, so Equation 10 reduces to

$$\begin{aligned} R(t) &= R_{\text{main}}(t) + R^{act}_{\text{spare}}(t) \int_0^t f_{t_{\text{main}}}(y)\,dy \\ &= P(t_{\text{main}} > t) + P\left(t_{\text{spare}} > t\right) - P(t_{\text{main}} > t) P\left(t_{\text{spare}} > t\right) \\ &= 1 - \left(1 - R_{\text{main}}(t)\right)\left(1 - R_{\text{spare}}(t)\right) \end{aligned} \qquad \textbf{Eq. 11}$$

which coincides with the reliability for a parallel configuration (or AND gate). Similarly, in the case of cold spares, $R^{sb}_{\text{spare}} \equiv 1$, which means that the equivalent operating time for the spare unit if it had been operating at the active mode is $t_e = 0$. So, Equation 10 simplifies to

$$R(t) = R_{\text{main}}(t) + \int_0^t f_{t_{\text{main}}}(y) R_{\text{spare}}^{act}(t-y)\,dy \qquad \textbf{Eq. 12}$$

References

[1] Agena Ltd. 2005. AgenaRisk Software Package, www.agenarisk.com.

[2] Allen J. F., "An interval based representation of temporal knowledge. In IJCAI-81, 221–226, 1981.

[3] Amari S, Dill G, Howald E. "A new approach to solve dynamic fault trees", Reliab Maintainability Symp, 27–30 Jan: 374–9. 2003

[4] Bobbio A, Portinale L, Minichino M, and Ciancamerla E, "Improving the analysis of dependable systems by mapping fault trees into Bayesian networks", Reliability Engineering and System Safety 71 (3), 249–260, 2001.

[5] Boudali H, Dugan J. B, "A discrete-time Bayesian network reliability modeling and analysis framework," Reliability Engineering and System Safety, vol. 87, no. 3, pp. 337–349, 2005.

[6] Boudali H, Dugan J. B, "A Continuous-Time Bayesian Network Reliability Modeling and Analysis framework," IEEE Transactions on Reliability, vol 55, pp 86-97, 2006.

[7] Bryant R. E, "Symbolic Boolean manipulation with Ordered Binary Decision Diagrams", ACM Computing Surveys, 24:293-318, 1992.

[8] Cobb B., and Shenoy P. "Inference in Hybrid Bayesian Networks with Mixtures of Truncated Exponentials", University of Kansas School of Business, working paper 294, 2005a.

[9] Doyle S. A. and Dugan J. B., "Dependability Assessment Using Binary Decision Diagrams (BDDs). In FTCS '95: Proceedings of the Twenty-Fifth International Symposium on Fault-Tolerant Computing, page 249, Washington, DC, USA, IEEE Computer Society 1995.

[10] Dugan J.B., Bavuso S. J. and Boyd M.A., "Dynamic Fault Tree models for Fault Tolerant Computer Systems", IEEE Trans. Reliability, vol 41, pp 363-377, Sept. 1992.

[11] Dugan J.B., Bavuso S. J. and Boyd M.A., "Fault-trees and Markov models for reliability analysis of fault-tolerant digital systems", Reliability Engineering and System Safety 39, 291–307, 1993.

[12] Dugan J.B., Sullivan K. and Coppit D, "Developing a low-cost high-quality software tool for dynamic fault-tree analysis", IEEE Trans Reliability, vol 49 (1), pp 49–59, 2000.

[13] Fenton N, Krause P, Neil M, "A Probabilistic Model for Software Defect Prediction", submitted to IEEE Transactions of Software Engineering, 2001.

[14] Fenton NE, Krause P, Neil M, "Probabilistic Modelling for Software Quality Control", Journal of Applied Non-Classical Logics 12(2), 173-188, 2002

[15] Fenton N, Littlewood B, Neil M, Strigini L, Sutcliffe A, Wright D, "Assessing Dependability of Safety Critical Systems using Diverse Evidence", IEE Proceedings Software Engineering, 145(1), 35-39, 1998.

[16] Fenton N and Neil M, "A Critique of Software Defect Prediction Models", 25(5) IEEE Transactions on Software Engineering, 675-689,1999.

[17] Ghahramani Z., "Learning Dynamic Bayesian Networks", In C.L. Giles and M. Gori (eds.), Adaptive Processing of Sequences and Data Structures. Lecture Notes in Artificial Intelligence, 168-197. Berlin: Springer-Verlag, 1998.

[18] Gulati R. and Dugan J. B., “A modular approach for analyzing static and dynamic fault trees,” in Proc. Ann. Reliability and Maintainability Symp., pp. 57–63, 1997.

[19] Jensen F. Bayesian Networks and Decision Graph, Springer, 2001.
[20] Jensen F, Lauritzen S.L, Olesen K. Bayesian updating in recursive graphical models by local computations, Computational Statistics Quarterly, 4, p. 260-282, 1990.
[21] Koller D., Lerner U., and Angelov D. “A general algorithm for approximate inference and its applications to Hybrid Bayes Nets”, in K.B Laskey and H. Prade (eds.), Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence, pp. 324–333, 1999.
[22] Kozlov A.V, Koller D. Nonuniform dynamic discretization in hybrid networks, in D. Geiger and P.P. Shenoy (eds.), Uncertainty in Artificial Intelligence, 13, p. 314-325, 1997.
[23] Langseth H, “Bayesian Networks with Applications in Reliability Analysis”, Ph.D. thesis, Dept. of Mathematical Sciences, Norwegian University of Science and Technology, 2002.
[24] Langseth, H., and Portinale, L. “Bayesian networks in reliability”, Reliability Engineering and System Safety. (To appear), 2006. (Also available as Technical Report No. TR-INF-2005-04-01-UNIPMN, Dept. of Computer Science, University of Eastern Piedmont “Amedeo Avogadro”, Alessandria, Italy)
[25] Lauritzen S.L. Graphical Models, Oxford, 1996.
[26] Lauritzen S.L., and Speigelhalter D.J. “Local Computations with Probabilities on Graphical Structures and their Application to Expert Systems (with discussion)”, Journal of the Royal Statistical Society Series B, Vol. 50, No 2, pp.157-224, 1988.
[27] Montani S, Portinale L, Bobbio A. “Dynamic Bayesian networks for modeling advanced fault tree features in dependability analysis.In: Proceedings of the sixteenth European conference on safety and reliability. Leiden, The Netherlands: A.A. Balkema, p. 1415–22, 2005.
[28] Moral S., Rumi R., and Salmeron A. “Mixtures of truncated exponentials in hybrid Bayesian networks”, in P. Besnard and S. Benferhart (eds.), Symbolic and Quantitative Approaches to Reasoning under Uncertainty, Lecture Notes in Artificial Intelligence, 2143, 156–167, 2001.
[29] Murphy K, “Dynamic Bayesian Networks: Representation, Inference and Learning. PhD thesis, Dept. Computer Science, UC Berkeley, 2002.
[30] Murphy, K. “A variational approximation for Bayesian networks with discrete and continuous latent variables, in K.B. Laskey and H. Prade (eds.), Uncertainty in Artificial Intelligence, 15, 467-475, 1999.
[31] Neil M, Fenton N, Forey S, Harris R. Using Bayesian Belief Networks to Predict the Reliability of Military Vehicles, IEE Computing and Control Engineering, 12(1), p. 11-20, 2001.
[32] Neil M., Krause P., and Fenton N, “Software Quality Prediction Using Bayesian Networks in Software Engineering with Computational Intelligence”, edited by Khoshgoftaar T. M). The Kluwer International Series in Engineering and Computer Science, Volume 73, 2003b.
[33] Neil M, Malcolm B, Shaw R. Modelling an Air Traffic Control Environment Using Bayesian Belief Networks, 21st International System Safety Conference, Ottawa, Ontario, Canada. 2003a.
[34] Neil M., Tailor M. and Marquez D., “Inference in Bayesian Networks using dynamic discretisation”. Accepted for publication in the Journal of Statistics and Computing, 2006.
[35] Neil M, Tailor M, Marquez D, Fenton N. and Hearty P, "Modeling Dependable Systems using Hybrid Bayesian Networks". Proceedings of the first international conference on Availability, Reliability and Security, 20-22, p. 817-821, 2006.

[36] Pearl J. “Graphical models, causality, and intervention”, Statistical Science, vol 8, no. 3, pp. 266-273, 1993.

[37] Portinale L, Bobbio A, “Bayesian networks for dependability analysis: an application to digital control reliability. In: Proceedings of the fifteenth conference on uncertainty in artificial intelligence. San Francisco, CA: Morgan Kaufmann Publishers; 1999.

[38] Rausand, M. Hoyland, A. System Reliability Theory: Models and Statistical Methods, John Wiley & Sons Inc, 2004.

[39] Shacter R., and Peot M. “Simulation approaches to general probabilistic inference on belief networks”, In Proceedings of the 5th Annual Conference on Uncertainty in AI (UAI), pages 221-230, 1989.

[40] Schneeweiss W. G. The Fault Tree Method. LiLoLe Verlag, 1999.

[41] Shenoy P., and Shafer G. “Axioms for probability and belief-function propagation, Readings in uncertain reasoning, Morgan Kaufmann Publishers Inc, p.p. 575 – 610, 1990.

[42] Solano-Soto J, Sucar, LE. “A methodology for reliable system design”, In: Proceedings of the 4th international conference on industrial and engineering applications of artificial intelligence and expert systems. Lecture notes in artificial intelligence, vol. 2070. Berlin, Germany: Springer, p. 734–45, 2001.

[43] Speigelhalter D.J., Lauritzen S.L. “Sequential updating of conditional probabilities on directed graphical structures”, Networks, 20, pp. 579-605, 1990.

[44] Torres-Toledano JG, Sucar LE. “Bayesian networks for reliability analysis of complex systems”, In: Proceedings of the 6th Ibero-American conference on AI (IBERAMIA 98), Lecture notes in artificial intelligence, vol. 1484. Berlin, Germany: Springer, p. 195–206, 1998.

[45] Watson H. A. and Bell Telephone Laboratories, “Launch Control Safety Study,” Bell Telephone Laboratories, Murray Hill, NJ USA, 1961

[46] Weber P, Jouffe L., “Reliability modeling with Dynamic Bayesian Networks”, 5th IFAC Sysmposium on Fault Detection, Supervision and Safety of Technical Processes, Washington, D.C., USA, 2003