

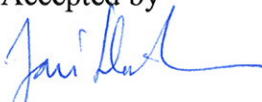




## Digital Automation System Reliability Analysis — Literature survey

Authors: Kim Björkman

Confidentiality: Public

Report's title Digital Automation System Reliability Analysis — Literature survey	
Customer, contact person, address VYR	Order reference 27/2007SAF
Project name Challenges of risk-informed safety management	Project number/Short name 32503-1.5/CHARISMA
Author(s) Kim Björkman	Pages 25
Keywords digital I&C, dynamic reliability methodologies, fault tree	Report identification code VTT-R-08153-09
Summary <p>The new programmable digital logic controllers enable more complicated control tasks than the old analogue systems. This creates new challenges for safety evaluation.</p> <p>The static event tree/ fault tree (ET/FT) approach has traditionally been used in the reliability modelling of digital I&amp;C systems. Because of the complex interactions between the hardware, software and the physical process, new more dynamic approaches are required for dependable reliability analysis.</p> <p>A literature review of dynamic reliability assessment methodologies is presented. These methodologies include e.g. Petri nets, Markov models and Bayesian methodologies. Additionally, Model Checking and Binary Decision Diagrams are discussed. Model checking is computer aided automatic verification technique and Binary Decision Diagrams are data structures used to represent Boolean functions.</p> <p>All of the methods surveyed included unique features that could be suitable for specific reliability analysis applications. Nevertheless, it seems that none of them can be used as such to model all relevant features of digital I&amp;C systems. If these features could be developed further or integrated with others models, the methods resulting from such modifications could lead to approaches that could be able to model digital I&amp;C systems in a quantitative manner with required level of detail.</p>	
Confidentiality	Public
Espoo 17.03.2010	
Written by  Kim Björkman Research Scientist	Reviewed by  Jan-Erik Holmberg Chief Research Scientist
	Accepted by  Jari Hämäläinen Technology Manager
VTT's contact address Vuorimiehentie 3, Espoo P.O.Box 1000, FI-02044 VTT, Finland e-mail: <a href="mailto:name.surname@vtt.fi">name.surname@vtt.fi</a> , <a href="http://www.vtt.fi">www.vtt.fi</a>	
Distribution (customer and VTT) SAFIR TR8, VTT archive	
<p><i>The use of the name of the VTT Technical Research Centre of Finland (VTT) in advertising or publication in part of this report is only permissible with written authorisation from the VTT Technical Research Centre of Finland.</i></p>	

## Contents

Abbreviations .....	3
1 Introduction .....	4
2 Basics of Digital Design and Reliability Theory.....	5
2.1 Definitions .....	5
2.2 Digital Design .....	6
2.3 Basics of Reliability Theory .....	7
3 Fault Tree Analysis.....	8
4 Dynamic Flowgraph Methodology.....	10
5 Petri Nets.....	11
6 Markov Models .....	12
7 Bayesian Methodologies .....	13
8 Model Checking .....	14
9 Binary Decision Diagram.....	15
10 Other Methods .....	16
11 Discussion .....	17
12 Conclusions .....	19
References.....	20
APPENDIX 1. Reliability Analysis Software.....	24

## Abbreviations

BBN	Bayesian Belief Network
BDD	Binary Decision Diagram
CCMT	Cell to Cell Mapping Technique
DFM	Dynamic Flowgraph Methodology
ET	Event tree
FSAP	Formal Safety Analysis Platform
FT	Fault tree
FTA	Fault tree analysis
I&C	Instrumentation and Control
MCS	Minimal Cut Set
PRA	Probabilistic Risk Assessment
PSA	Probabilistic Safety Assessment
RBD	Reliability Block Diagram
ROBDD	Reduced Ordered BDD

# 1 Introduction

Instrumentation and control (I&C) systems play a crucial role in the operation of safety critical processes. An important change is the replacement of the old analogue I&C systems with new digitalised ones. The programmable digital logic controllers enable more complicated control tasks than the old analogue systems. This creates new challenges for safety evaluation.

The static event tree/ fault tree (ET/FT) approach has traditionally been used in the reliability modelling of digital I&C systems especially in the nuclear power plant domain. Because of the complex interactions between the hardware, software and the physical process, new more dynamic approaches are required for dependable reliability analysis. These interactions include issues like:  
[NUREG/CR-6901]

- Digital I&C systems rely on sequential circuits which have memory. Consequently, digital I&C system outputs may be a function of system history, as well as the rate of progress of the tasks.
- Tasks may compete for a digital controller's resources. This competition requires coordination between the tasks leading to problems such as deadlock and starvation.
- The choice of internal/external communication mechanisms for the digital I&C system (such as buses and networks) and the communication protocol affect the rate of data transfer and subsequently the digital I&C system reliability and robustness.
- The ability to coordinate multiple digital controllers directly and explicitly may necessitate a finer degree of communication and coordination between the controllers.
- A digital controller can remain active and not only react to data, but can anticipate the state of the system.
- Tight coupling and less tolerance to variations in operation increases the digital I&C system sensitivity to the dynamics of the controlled physical process and, thus, its representation in the digital I&C system reliability model.

Reliability modelling approaches that can through explicit consideration of the time element in system evolution account for the coupling between the triggered or stochastic logical events can be considered as dynamic methodologies. Dynamic methodologies can provide a much more accurate representation of probabilistic system evolution in time than the ET/FT approach.

The purpose of this report is to be a brief survey of some of the current reliability assessment methodologies for digital I&C systems. Similar reviews have been performed in [NUREG/CR-6901 and NUREG/CR-6962]. This report discusses many of the same items as the other surveys, but additionally a couple of methods are discussed that could be used to support more efficient reliability assessment and also some existing software are listed. The rest of this report is structured as follows. Chapter 2 introduces basics of digital design and reliability theory.

Chapter 3-7 reviews a number of reliability assessment methodologies. Fault tree analysis is discussed in chapter 3, DFM in chapter 4, Petri nets in chapter 5, and Markov models and Bayesian methodologies in chapters 6 and 7. Additionally model checking, an automatic verification technique, is discussed in chapter 8. Chapter 9 presents binary decision diagram, which is a directed acyclic graph for representing Boolean functions. In chapter 10 some additional dynamic reliability modelling approaches are briefly presented. Chapters 11 and 12 conclude this study. Appendix 1 lists a number of software of the presented methodologies. The list is by no means complete. The purpose of the list is to show that there are a high variety of tools already available.

## 2 Basics of Digital Design and Reliability Theory

### 2.1 Definitions

**Boolean function:** Boolean function of  $n$  variables is a function on  $\mathbf{B}^n$  into  $\mathbf{B}$ , where  $\mathbf{B}$  is the set  $\{0, 1\}$ ,  $n$  is a positive integer, and  $\mathbf{B}^n$  denotes the  $n$ -fold Cartesian product of the set  $\mathbf{B}$  with itself.

**Literal:** A binary variable or its complement (e.g.  $x, y, \bar{x}, \bar{y}$ )

**Product term:** A single literal or a logical product of two or more literals (e.g.  $\bar{x}, x \cdot y, x \cdot \bar{y} \cdot z$ )

**Sum of products:** A logical sum of product terms (e.g.  $\bar{x} + (x \cdot y) + (x \cdot \bar{y} \cdot z)$ )

**Sum term:** A single literal or a logical sum of two or more literals (e.g.  $\bar{x}, x + y, x + \bar{y} + z$ )

**Product of sums:** A logical product of sums (e.g.  $\bar{x} \cdot (x + y) \cdot (x + \bar{y} + z)$ )

**Minterm:** An  $n$ -variable minterm is a product term with  $n$  literals. There are  $2^n$  such product terms (e.g. 3-variable minterms,  $x \cdot \bar{y} \cdot z, x \cdot y \cdot \bar{z}$ )

**Maxterm:** An  $n$ -variable maxterm is a sum term with  $n$  literals. There are  $2^n$  such sum terms (e.g. 3-variable maxterms,  $x + \bar{y} + z, x + y + \bar{z}$ )

**Imply:** A logical function  $P(x_1, \dots, x_n)$  implies a Boolean function  $F(x_1, \dots, x_n)$  if for every input combination such that  $P=1$ , then  $F=1$  also

**Prime implicant:** A prime implicant of a Boolean function  $F(x_1, \dots, x_n)$  is a normal product term  $P(x_1, \dots, x_n)$  that implies  $F$ , such that if any variable is removed from  $P$ , then the resulting product term does not imply  $F$ .

**Structure function:** A binary function  $\varphi(a) = \varphi(a_1, a_2, \dots, a_n)$ , where  $a_i$  is 1 if the component is functioning and 0 if the component is not functioning. Similarly  $\varphi(a)$  is 1 if the system is functioning and 0 if it is not functioning.

**Cut vector:** A cut vector for a binary function is any  $a$  such that  $\varphi(a) = 0$ . A cut vector,  $a$ , is a minimal cut vector if for every  $b > a$  (component-wise inequality),  $\varphi(b) = 1$ .

**Minimal cut set:** A minimal cut set is the set of components in a minimal cut vector that are failed.

## 2.2 *Digital Design*

Logic circuits can be classified into two types, combinational circuits and sequential circuits. The output of a combinational logic circuit depends only on its current inputs. The output of the circuit is always the same with the same inputs

The outputs of a sequential logic circuit depend, besides, on its current inputs also on the current state of the circuit. The current state of the circuit depends on the current and previous inputs. The output can alter even with the same inputs.

Truth table is the most basic representation of a Boolean function. A truth table lists the output of a circuit for every possible input combination. Generally, the input combinations are arranged in row in ascending binary counting order. The truth table of an  $n$ -variable Boolean function has  $2^n$  rows, which makes truth tables impractical to write for Boolean functions with more than few variables.

A combinational circuit can be straightforwardly represented by truth tables. In case of sequential logic circuit state tables are used. State tables are the sequential analogue for truth tables. In a state table the current states of the circuit is considered as additional inputs, whereas the next states of the circuit is considered as additional outputs. The construction of a state table is usually less straightforward than the construction of a truth table.

Based on the correspondence between minterms and the truth table an algebraic representation of a Boolean function can be formed. The canonical sum of a Boolean function is a sum of the minterms corresponding to the input combination for which the function produces a 1 output.

Binary Decision Diagrams (BDD) are a graphical representation of a Boolean function. A BDD is an acyclic directed graph. The Binary Decision Diagram is more thoroughly discussed in chapter 9.

It is often uneconomical to realize a logic circuit from the first logic expression that one can think of. Canonical sum expressions are especially expensive, since the number of possible minterms, and hence gates, grows exponentially with the number of variables. Combinational circuits are usually minimized by reducing the number and size of gates that are needed to build it. Most minimization methods are based on a generalization of the theorem:

$$x_1 \cdot F(x_2, x_3, \dots) + \bar{x}_1 \cdot F(x_2, x_3, \dots) = F(x_2, x_3, \dots), \quad (1)$$

where  $x_1$  is a Boolean variable and  $F(\cdot)$  is a Boolean function.

The classical procedures for minimizing Boolean functions are the Karnaugh mapping and the Quine-McCluskey algorithm. A Karnaugh map is a graphical representation of a Boolean function's truth table. The Karnaugh map of an  $n$ -input Boolean function is an array with  $2^n$  cells. One cell for each input combination or minterm. Karnaugh maps are applicable for expressions with 2-6 variables. The map is used for minimization of Boolean expressions. The map is based on drawing the truth table of a coupling function to a format that allows visually apply the generalization of theorem (1).

The Quine-McCluskey algorithm is an approach for logic minimization. The algorithm is functionally similar to Karnaugh mapping, but the tabular form makes the algorithm more efficient for use in computer algorithms. The tabular method makes use of the generalization of theorem (1). The algorithm as two steps: [Wakerly\_2000]

1. finding all prime implicants of the function, and
2. selecting a minimal set of prime implicants that covers the function.

The Quine-McCluskey algorithm has its limitation because the problem the algorithm strives to solve is NP-Complete. That is, in the worst case the runtime of the algorithm grows exponentially with the input size.

## 2.3 **Basics of Reliability Theory**

The starting point of reliability theory is that life time of systems or components can not be predicted reliably because failures are random. A failure is however not completely random, but, rather, follows some statistical laws. Reliability analysis combines various mathematical techniques, drawn mainly from probability, statistics and the theory of stochastic processes, to be used to the measurement and prediction of the reliability of components and systems. Components can be understood to mean e.g. mechanical, electronic or software components as well as human interventions. A system could include anything from a computer to a digital feed water control system.

Reliability is the ability of an item to perform a required function, under given environmental and operational conditions and for a stated period of time [ISO 8402]. The term "item" is here used to mean any component, system or sub-system that can be considered as an entity. There are many ways to measure reliability, for instance, mean time to failure (MTTF), failure rate, survival probability (the probability that the item does not fail in a time interval  $(0,t]$ ), availability at time  $t$  (the probability that the item is able to function at time  $t$ ), and fulfilment probability (the probability that the item fulfils its mission).

In reliability analysis systems are often modelled graphically. This provides a visual representation of the components and how they are configured to form a system. The reliability block diagram (RBD) is one of the most commonly used system representation in reliability analysis. In a reliability block diagram, a system is divided into blocks that represent distinct elements, e.g., components or subsystems. The structure of the RBD defines the logical interaction of failures within a system. The blocks are assembled in series and parallel arrangements between the system input and output nodes. For the system to be functioning at least one path must exist between the system input and output nodes.



A structure function provides another way to represent the relationships of a system's components. A structure function of a system with  $n$  components is a binary function

$$\varphi(a) = \varphi(a_1, a_2, \dots, a_n), \quad (2)$$

where  $a_i$  is 1 (TRUE) if the component is functioning and 0 (FALSE) if the component is not functioning. Similarly  $\varphi(a)$  is 1 if the system is functioning and 0 if it is not functioning. If the functioning of all components is known also the functioning of the system is known. The structure function is generally established from a RBD.

A system is said to be coherent if each of its component is relevant and the structure function is monotone. (A component is irrelevant if it does not matter if the component is working or not.) A system is said to be monotone if fixing a failed component cannot make the system worse. A structure function of a monotone system has the property  $\varphi(a) \leq \varphi(b)$  when  $a \leq b$ , where the latter inequality is understood to be applied component-wise. [Samaniego\_2007]

Besides reliability block diagrams and structure functions minimal path and cut sets are used to represent the structure of a system. A path vector for a system is any  $a$  such that  $\varphi(a) = 1$ . A cut vector for a system is any  $a$  such that  $\varphi(a) = 0$ . A path vector  $a$  is a minimal path vector if for every  $b < a$  (inequality applied component-wise)  $\varphi(b) = 0$ . The minimal path set is the set of components in a minimal path vector that are functioning. A cut vector,  $a$ , is a minimal cut vector if for every  $b > a$  (component-wise inequality),  $\varphi(b) = 1$ . A minimal cut set is the set of components in a minimal cut vector that are failed. That is, a minimal cut set is a minimal set of components such that if they all have failed the system has failed, but if one of them is functioning then the system is functioning.

Reliability analysis can be divided into two main categories: qualitative and quantitative. Qualitative analysis strives to identify the various failure modes and causes that effect the unreliability of a system. Quantitative analysis aims to produce quantitative estimates of system reliability using real failure data together with suitable mathematical models.

There are sophisticated reliability methods for analogue systems. For digital systems more suitable methods are continuously developed.

### 3 Fault Tree Analysis

Fault tree analysis (FTA) has gained wide spread acceptance as an important tool for evaluating reliability and safety in system design, development and operation. Fault tree analysis is extensively and successfully used in safety studies in e.g. nuclear, chemical and aerospace studies. The method is also finding its way into many other fields, such as robotic and transport. FTA is the most common technique for causal analysis in risk and reliability studies [Rausand\_&\_Høyland\_2004].

Fault tree analysis is a top down approach for failure analysis. Fault trees are graphical representation of undesired events that can lead to an undesired state of the system represented by a top event. In the analysis it is determined how the top event can be caused by individual or combined lower level failures or events. The efficiency of fault tree analysis is reached primarily when it is integrated with the event tree analysis as a part of the probabilistic safety assessment. However, in recent years some drawbacks of the FTA have become apparent. Especially, the inability to capture time dependent dynamic behaviours have made the use of fault trees somewhat impractical in assessing the reliability of digital I&C systems.

Generally fault tree analysis methods are based on the minimal cut set approach. In the minimal cut set approach the number of possible cut sets grows exponentially with the size of the fault tree. Therefore, with very large fault trees enumerating all possible cut sets is nearly impossible due to memory requirements and long computing times. The most common methods to determine a manageable sized collection of minimal cut sets are elimination of cut sets whose size is greater than some predefined value, or whose probability is lower than some predefined probability called truncation probability [NUREG-0492].

There are several approaches for generating minimal cut sets for a given fault tree. Generally, the approach for generating minimal cut sets is based on deterministic programming or Monte Carlo simulation [Kara-Zaitri\_IJQRM\_96] and [Lee\_et\_al\_1985].

Deterministic methods are based on the direct reduction or expansion of the expression using Boolean algebra for the top event in terms of the element. There exist a high variety of different programs realizing deterministic methods. Some of these are briefly presented in [Kara-Zaitri\_IJQRM\_96]

In Monte Carlo simulation approach, events are randomly sampled to determine if a sampled combination of events results in the occurrence of the top event. If top event occurs, the combination represents a cut set, and the minimal cut sets are determined from those cut sets.

While simulation approaches can be substantially faster than deterministic approaches, it has limitations, such as:

- All minimal cut sets may not be found
- It is less accurate and may result in the non-occurrence of important minimal cut sets (smaller order or higher probabilities).
- In practice, obtaining minimal cut sets with order four or more can require unreasonably long computing times.

Because the number of MCSs grows exponentially with the size of the fault tree, all MCSs cannot be calculated due to computing resources limitations. Therefore, MCSs are generally truncated in the evaluation of large fault trees. To overcome this problem binary decision diagram based approaches have been developed.

Evaluation of fault trees using the BDD approach has been studied in e.g. [FT\_handbook-Aerospac, Rauzy\_FTA\_93, Tang\_RAMs\_2004, Rauzy\_2001]. The BDD provides advantage from the computational viewpoint. However, failure causality is better represented by fault trees. Thus, generally the advantages

of BDD are utilized by first constructing a fault tree and then converting this to a BDD. There are several methods for fault tree conversion. For instance, in the BDD constructing method developed by Rauzy [Rauzy\_FTA\_93] a set of rules are repeatedly applied to construct a BDD. An alternative approach for BDDs is presented in [Way\_RESS\_2000], where BDD constructs are first formed for each of the gate types and then joined together as specified by the gates in the fault tree. BDD allows calculating of the top-event probability in an exact and efficient way. Additionally BDDs can be used to compute and calculate very large sets of MCS.

A collection of FTA software is presented in Appendix 1.

## 4 Dynamic Flowgraph Methodology

Dynamic Flowgraph Methodology (DFM) [Garrett\_DFM\_1995] is an approach based on directed graphs for modelling and analyzing the behaviour and interaction of software and hardware within an embedded system. The directed graphs of DFM models consist of relations of causality and conditional switching actions represented by arcs that connect network nodes and special operations. Important process variables and parameters are represented by nodes while different types of causal and logical relationships among them are represented by operators. With some limitations a DFM model can provide a complete representation of the way a system of interacting components and parameters is supposed to work and how failures and abnormal conditions and interactions can endanger this working order.

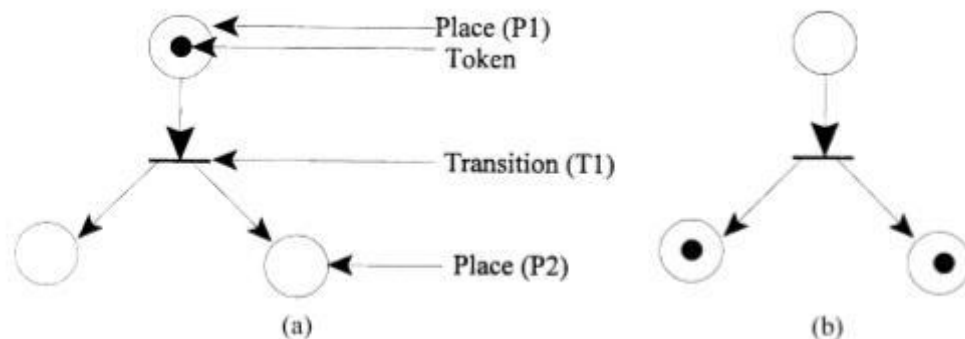
The benefits of DFM are that it has the ability to represent changes in the system logic in discrete points in time. This allows the development of timed fault trees which reflect static relationships between the variables at different points in time. Thus, the results of a DFM analysis are attained in the form of fault trees, which show how the analysed system states may occur. Therefore, the results provided by DFM shares many of the features of fault tree analysis. The prime implicants of the fault trees can be used to identify and eliminate system faults resulting from unanticipated combinations of software logic errors, hardware failures and adverse environmental conditions. A second advantage of DFM, as compared with fault trees, is that the system model and the top event are separated, and thus the same system model can easily be used with different top events.

Currently, it seems that there is only one DFM software (see Appendix 1), called DYMONDA. DYMONDA is in the beta testing stage. DYMONDA contains a graphical editor for DFM models, an analyzer that can analyze a DFM model forward or backward in time, and save the results in pure textual or XML form. When analyzing backward in time, DYMONDA can also compute the probability of the top event, given the probabilities of the individual states of input variables.

For further information see [Karanta\_Maskuniitty\_2009].

## 5 Petri Nets

A Petri net can be described as a graphical modelling language. The use of state transitions, arcs and nodes makes it similar to finite state machines. Petri nets are used in modelling discrete event systems. The Petri net is a modelling scheme introduced by C. A. Petri in the early 1960s. A Petri net is composed of a set of places, a set of transitions and an initial marking representing the number of tokens in each place at time  $t = 0$ . Petri nets are represented with a graph structure, where places are represented by circles, transitions by horizontal lines and tokens by dots in the circles. The execution of a Petri net is performed by firing transitions (movement of tokens around places). A transition is enabled when each of the input places contains at least one token. Figure 1 (a) represents a simple Petri net and Figure 1 (b) the Petri net after the transition has fired. [Labeau et al. 2000]



**Figure 1 (a) A simple Petri net. (b) After firing of T1 [Labeau et al. 2000]**

Petri nets are a promising tool for describing and studying systems that are characterised as being concurrent, asynchronous, distributed, parallel, nondeterministic, and/or stochastic. Petri nets provide a uniform environment for modelling, formal analysis, and design of discrete event systems.

Over the years, the concept of time dependent transitions has been introduced to Petri nets. Originally, they were either deterministic times or times derived from a constant transition rate, which created the mapping between Petri nets and Markov chains. The theory of Petri nets was extended to include a set of transitions that fire at random times. These Petri nets are called Generalized Stochastic Petri Nets (GSPN) and are representatives of semi Markov processes.

Dynamic systems can be qualitatively represented by stochastic Petri nets [Labeau et al. 2000]. Process variable models can be modelled with Petri nets and Monte Carlo simulation can be used to perform quantitative analyses of these Petri Nets. An overview of Petri net theory and Stochastic Petri nets is presented in [Cordier\_RESS\_1997].

Petri nets have been used in automation in different applications. One of the main application areas of Petri nets is deadlock prevention and analysis. [Fanti & Zhou 2004] presents a comprehensive survey of deadlock control methods based on Petri nets to be applied to automated manufacturing systems. Petri nets or its extensions (e.g. fuzzy, coloured, timed and stochastic) have also been studied in the use in detection and treatment of faults (e.g. [Gomes et al. 2004, Zhenjuan et al. 2006]). Other application areas include, e.g., modelling and analysis of

communication protocols, sequence controllers, manufacturing systems, software systems and communication networks [Zurawski & Zhou 1994]. Petri nets can also be used to directly simulate fault trees. [Liu\_&\_Chiou\_1997] describes such a method and presents an algorithm for generating minimal cut sets of the translated trees.

Limitations of Petri nets include that they lead to a combinatorial explosion with the number of states in larger systems. Stochastically interpreted Petri nets cannot be formally proof checked and, thus, the verification process can be difficult and take a long time [Labeau et al. 2000]. Additionally, stochastic Petri nets assume exponential distribution of timed firings that may be unreasonable for some types of systems.

There exists several Petri net software. [Petri\_Net\_World] reviews a large number of these software (see also Appendix 1).

## 6 Markov Models

Traditionally Markov models have been utilized to model statistically dependent failures of hardware systems [NUREG/CR-6901]. Markov methods have generally been accepted as an appropriate method for analyzing fault tolerant digital systems. Markov models are useful for modelling various sequence dependencies, common-cause failures and failure event correlation [Digi\_1997]. Due to the state-space explosion of Markov models, they cannot be used to model large scale systems. To overcome this issue Markov models are usually used for modelling subsystems where needed, e.g. sub-systems of a digital I&C system [NUREG/CR-6942].

A Markov chain is a stochastic process that possesses the Markov property. The states of a system in a Markov chain are defined in advance and the transitions between states occur at discrete points in time. If the time is continuous we have a continuous time Markov chain, also called the Markov process. In this case the transitions are assumed to be exponentially distributed. Markov chains are memoryless, that is, transition probability is dependent only on the current state of the system, and not of the history of the states.

According to [NUREG/CR-6901] one of the most promising Markov approaches for modelling digital systems is the Markov/CCMT (cell-to-cell mapping) technique. The Markov/CCMT approach combines the traditional Markov methodology with cell to cell mapping. This approach allows one to represent possible coupling between failure events, originated from dynamic interactions between the digital I&C system and the controlled process, and among the different components of the I&C system [NUREG/CR-6985]. The CCMT is a systematic procedure to present the dynamics of linear and non-linear systems in discrete time and discretized state space. [Bucci et al\_RESS\_08] gives an overview of the Markov/CCMT methodology used to construct the Markov reliability model for an example dynamic system. The Markov/CCMT can represent: [NUREG/CR-6985]

- memory effects (by adding auxiliary states or auxiliary variables),

- logic loops along with time dependent and system state dependent transitions, epistemic uncertainties,
- non-linear aspects of the system dynamics and stochastic fluctuations in dynamic system operation,
- logic interactions within the digital I&C system components (failover),
- statistically dependent failure probabilities/rates, and,
- failure probabilities/rates that may be affected by the environment (e.g. pressure, temperature)

The limitations of the approach are that significant computational effort may be required to solve the model, construction of the model requires large amount of technical knowledge and because the method produces a large amount of data, post processing of the data is required.

There exists a number Markov analysis software. Quite often they are a part of a larger reliability analysis tool family (see Appendix 1).

## 7 Bayesian Methodologies

A Bayesian Belief Net (BBN) [Pearl\_1988] is a directed graph consisting of a set of nodes connected by a set of edges. With each node events and singular propositions are associated. Uncertainty is expressed by a probability density. The belief of a variable is expressed by the probability density. Beliefs are the probability that a variable will be in a certain state based on the addition of evidence in a current situation. The probability depends conditionally on the status of other nodes at the incoming edges to the node. The edges of the network model the relationship between adjacent nodes. The strength of these relations is represented as conditional probability distributions. The computation of a specific nodes belief is based on the rules for probability calculations using Bayes' formula.

Bayesian belief networks provide an efficient way to model reasoning with uncertainty and they have a solid mathematical background in probability theory. BBNs can be utilized to formally include expert judgement into the modelling process. BBNs have been used in a variety of application areas, including reliability and they can, for example, provide alternative representations of fault trees and reliability block diagrams [Sigurdsson\_2001]. A method for assessing multilevel system reliability is given in [Wilson\_RESS\_2007]. The use of BBNs in reliability have been reviewed, e.g., in [Sigurdsson\_2001, Doguc\_RESS\_2009].

In literature there are several methods defined for estimating system reliability mainly focusing on doing it for specific applications, e.g. nuclear power plants. However, basically none of these studies focuses on the problem of constructing Bayesian networks without the need of a human expert. [Doguc\_RESS\_2009] introduces a methodology to use statistical system reliability data to construct a representative BN model.

BBNs have mainly been used in artificial intelligence research as framework for modelling and reasoning with uncertainty. Bayesian methodologies have been employed in practice in many fields, such as interpretation of live telemetry data,

power generation monitoring, real-time weapons scheduling, medical diagnosis systems, and many other diagnostics applications [Ziv\_ICSM\_1997, Sigurdsson\_2001]. BBN has also been adapted to software safety assessment (for further information see [Gran\_2002, Helminen\_2001, Helminen\_Pulkkinen\_2003]). One of the main difficulties using Bayesian networks for quantitative reliability estimation of safety critical software is that the user had to build a different BN for each software development environment, which is very laborious and time-consuming work. [Eom\_et\_al\_NPIC\_2009] suggests solving this problem by using generalized BBN templates which are not restricted to a specific development environment.

Additionally, in the field of Bayesian methodologies, [Kelly\_RESS\_2009] presents the current state of the art of Bayesian inference in PRA. [Hamada et\_al\_2008] focuses on Bayesian reliability analysis, including topics like modelling, computation, sensitivity analysis, and model checking.

The use of BDDs to enhance the computation of BBN probabilities has been considered. [Minato\_IJCAI\_2007] proposes a method for compiling BBNs into Multi-Linear Functions based on Zero-suppressed BDD, a variant of BDD (see section 9). [Sanner\_IJCAI\_2005] presents a method for computing BBN probabilities using Affine Algebraic DDs (AADDs), another variant of BDDs.

There exists several BBN software. A number of these software are reviewed in [Murphy\_2005] (see also Appendix 1).

## 8 Model Checking

Model checking [Clarke et al 1999] is a computer aided automatic verification technique for formally verifying the correct functioning of a system design model against its formal specification. Generally, to model the system some version of a state machine is used. The specifications are, typically, formalized with temporal logics. The model checking tools examine the behaviour of the system design with all input sequences and compare it to the specification of the system. If none of the specifications is violated by the system behaviour, the (model of the) system is correct. If violation of a specification occurs, the model checking tools creates a counterexample execution of the system model demonstrating why the property is violated.

Model checking tools face the state explosion problem. There are several approaches to cope with this problem, e.g. symbolic model checking and bounded model checking. In symbolic model checking BDDs (see chapter 9) [Bryant\_1986, Clarke et al 1999] are usually used to symbolically represent and explore a system's state space. Bounded model checking is typically based on propositional satisfiability (SAT) (for further information on SAT see [Lynce 2005]).

The use of model checking to verify the correctness of fault trees has been studied. For example, [Koh\_NPIC\_2009] presents an approach that combines a safety assessment methodology (fault tree analysis) and a formal methodology (model checking) to provide formal, automated and qualitative assistance to

informal and quantitative safety assessment. A model checking tool called UPPAAL is used to verify the correctness of the fault tree [UPPAAL].

FSAP/NuSMV-SA is a platform that aims to improve the development cycle of complex systems by providing a uniform environment that can be used both at design time and for safety assessment [Bozzano\_&\_Villafiorita\_2007]. The platform consists of a graphical user interface (FSAP Formal Safety Analysis Platform) and an engine (NuSMV-SA) based on the NuSMV model checker [NuSMV]. The model checking engine provides support for system simulation and standard model checking capabilities. Additionally, the platform allows generating items typical to reliability analysis, such as fault trees. The platform is able to generate fault trees automatically from the definition of the system model and of the possible faults.

Also efficient symbolic techniques for probabilistic model checking have been developed [PRISM]. For example, the PRISM model checking tool uses BDDs and multi-terminal BDDs (MTBDD [MTBDD]) as underlying data structures. However, the tool provides three distinct engines for numeral computation. The first is a pure MTBDD approach, the second is a conventional explicit version using sparse matrixes, and the third is a hybrid approach of the previous two [PRISM].

## 9 Binary Decision Diagram

Binary Decision Diagrams (BDD's) was introduced by Akers [Akers\_1978]. A Binary Decision Diagram is a data structure used to represent a Boolean function. The BDD is based on the repeated application of the classical Shannon expansion formula:

$$f(x_1, x_2, x_3, \dots) = x_1 f(1, x_2, x_3, \dots) \vee \bar{x}_1 f(0, x_2, x_3, \dots) \quad (3)$$

The Boolean function is represented as a rooted, directed acyclic graph that consists of decision nodes with two edges the 1-edge and 0-edge, and terminal nodes called 0-terminal and 1-terminal representing the Boolean functions 0 and 1. A variable assignment for which the represented Boolean function is true is represented by a path from the root node to 1-terminal node. The size of a BDD is related to the number of nodes, which depends on the number of input variables and their ordering.

An ordered binary decision diagram (OBDD) is a BDD with the constraint that the input variables are ordered and every decision node to terminal node path in the OBDD visits the input variables in ascending order. By reducing the OBDD (see details in [Bryant\_1986]) a reduced ordered binary decision diagram (ROBDD) is obtained. Bryant [Bryant\_1986] demonstrated how a BDD could be modified to a ROBDD so that a canonical representation a Boolean function could be created.

For the representation of Boolean functions as BDDs the *ite* (If-Then-Else) connective is usually used. The *ite* connective is defined as follows. Let F, G and H be Boolean functions then



$$ite(F, G, H) = (F \wedge G) \vee (\bar{F} \wedge H) \quad (4)$$

The *ite* operation can be used to implement all two-variable Boolean functions. Because *ite* is the logical function performed at each node of the ROBDD, it can efficiently be used as a building block for many other operations on the ROBDD. For instance, [Brace et al 1990] and [Rauzy\_FTA\_1993] present efficient algorithms to build up a ROBDD utilizing the *ite* operation recursively. [Rauzy\_FTA\_1993] and [Coudert & Madre 1992] have presented algorithms to compute a BDD encoded prime implicants of a function represented by a BDD. Both of the algorithms use the same inductive decomposition principles. The algorithm presented by [Rauzy\_FTA\_1993] works only for monotone functions, but it is more efficient than the other. However, both of the algorithms outperform by orders of magnitude other already presented methods.

Several variants of the BDD have been proposed, for instance, Multi-Terminal BDD (MTBDD) and Zero-suppressed BDD (ZBDD). MTBDD [Clarke et al\_1993, Bahar\_ICCAD\_93] (also referred as algebraic decision diagram (ADD)) extends the BDD by allowing values from an arbitrary finite domain to be attached to the terminal nodes. Zero-suppressed BDD [Minato\_ZBDD\_1993] are based on a different reduction rules than ROBDD. This allows more efficient handling of function in certain cases, for example when the ON-set of the function to be represented is very sparse.

There exists several BDD libraries for BDD manipulation (see appendix 1), e.g. BuDDy [BuDDy] and CUDD [CUDD]. For instance, CUDD package provides functions to manipulate BDDs, ADDs, and ZBDDs, and it has been used in several softwares that utilize BDDs for efficient exhaustive testing of a systems model e.g. NuSMV [NuSMV] and PRISM [PRISM].

Aralia is a BDD engine for quantification of Boolean risk assessment models, such as fault trees, event trees and reliability block diagrams [Aralia]. Aralia provides several different data structures to encode Boolean functions, for example BDDs and ZBDDs that are used to encode minimal cut sets and prime implicants.

Besides the above mentioned application areas of BDDs, BDDs have been used, also in other applications of reliability assessment. For instance, [Tang\_Dugan\_IEEE\_2006, Zang\_et\_al\_IEEE\_1999] presents BDD-based methods for reliability analysis of phased mission systems.

## 10 Other Methods

Other dynamic reliability methodologies include, e.g., Test-Based methodologies, Flow Network model and Black-Box methodologies.

Test-based approaches can be used to approximate the reliability of a digital system. A measure of reliability of a digital system can be generated, by running a number of tests and measuring the number of failures [NUREG/CR-6901].

[Yang\_NPIC\_2009] proposes a method for calculating software reliability. The method uses a graphic flow network to represent the software structure. The reliability of the software is estimated based on the flow network model and test results. A flow network is used to model the software structure by nodes and edges. Software test results determine an edge's failure probability.

Black-Box models consider the software associated with a system or subsystem as a single "black box". The black box is characterized by one overall failure rate. Black box methodologies include, for instance, the Schweidewind model [Schneidewind\_IEEE\_1992] and Musa-Okumoto model [Musa\_Okumoto\_1984].

[Männistö\_VTT\_2006] reviews several dynamic reliability methodologies in the context of long mission time reliability. [Männistö\_VTT\_2006] discusses improvements to and extensions of the ET/FT method, (such as the GO-FLOW) method, explicit state-transition methods and implicit state transition methodologies. Several of these methods have also been reviewed in, e.g., [NUREG/CR-6901], [NUREG/CR-6962] and in [Borysiewicz\_et\_al\_2006].

## 11 Discussion

Event though static fault tree/event tree methodology has been used for reliability modelling of digital I&C systems especially in the nuclear power plant domain, several questions have been raised about the capability of the fault tree/event tree approach to properly account for dynamic interactions. Generally, dynamic methodologies provide a much more accurate representation of probabilistic system evolution in time than the fault tree/event tree approach. However, usually it is a difficult task to integrate dynamic models to existing PRAs of which almost all are based on the fault tree/event tree approach.

For a dynamic reliability methodology to be acceptable for assessment of digital I&C at least following requirements should be met [NUREG/CR-6901]

1. The methodology should account for possible dynamic interactions between: a) the digital system and controlled/supervised plant physical processes, and, b) the components of the digital system itself.
2. The model must be able to predict future failures well and cannot be purely based on previous experience.
3. The model must make valid and plausible assumptions and the consequences of violating these assumptions need to be identified.
4. The data used in the quantification process must be credible to a significant portion of the technical community.
5. The model must be able to differentiate between a state that fails one safety check and those that fail multiple ones.
6. The model must be able to differentiate between faults that cause function failures and intermittent failures.
7. The model must have the ability to provide uncertainties associated with the results.

None of the presented methods fulfil all the requirements. Since the fulfilment of these requirements is presented in [NUREG/CR-6901], the rest of this chapter concentrates on discussing some of the main benefits and limitations of the methodologies.

[NUREG/CR-6962] indicates that the traditional methodologies ET/FT and Markov modelling appear to be useful approaches for PRA of digital I&C systems, but they contain some limitations. As already previously discussed, ET/FT does not explicitly treat the timing of events in accident sequences and interactions with plant processes are implicitly and approximately considered. Markov models, on the other hand, the construction of the model can be a laborious, time-consuming manual process, and the resulting transition matrix can be extremely large (it can be so large that it may not be practical to build it).

[NUREG/CR-6901] ranked as the two top dynamic reliability modelling approaches with the most positive features and least negative features DFM and Markov approach coupled with cell to cell mapping techniques.

Dynamic flowgraphs can predict future failures and are able to integrate hardware and software components. The modelling paradigm used by DFM can be adjusted to logic forms that are equivalent to forms used by other binary logic model environments, such as RBDs, ETs, and FT. This makes it compatible with several traditional reliability model and PRA software. However, creation of a DFM model requires extensive technical knowledge of the behaviour of the system being analyzed. Continuous variables have to be discretized in such a way as to balance the model accuracy versus complexity and analysis time. The number of time steps that can be analyzed in deductive mode is limited by computational constraints.

Markov/CCMT model accounts for transitions between all system states defined by the user. Construction of a full Markov/CCMT model may not be computationally feasible if the analyzed system contains a large number of states (more than several thousand). In these cases, Markov/CCMT can more efficiently be used in the inductive mode where only a limited range of initial conditions are considered rather than all possible conditions. However, the construction of a Markov/CCMT model for any system requires a substantially larger amount of technical knowledge compared to that needed for a traditional ET/FT analysis. The Markov/CCMT approach may require significant computational effort in order to solve the model and to generate event sequences. Additionally some post processing of the results is required.

Markov/CCMT and the DFM approaches are not compensatory methods but rather they should be used in a complementary fashion. [NUREG/CR-6985] proposes that DFM should be used in the deductive mode (that is, backward in time) to identify possible failure sequences or initiating events that lead to specified event. Markov/CCMT should, on the other hand, be used in the inductive mode (forward in time) to guarantee completeness and verification of the quantification of the failure sequences that may require more detailed modelling.

Even though Petri net approaches are able to model well digital system, a combinatorial explosion with the number of states in larger systems can affect the solvability of the model in reasonable time. Additionally stochastically interpreted Petri nets cannot be formally proof checked.

The advantages of Bayesian methodologies concern mainly the assessment of software. For instance, Bayesian networks can be considered as a promising methodology for quantitative reliability estimation of safety critical software. One of the serious difficulties is that the user has to build a different BBN for each software development environment, which is time-consuming and laborious work. However, the abstraction level of the models is relatively high compared to the other methodologies making the models somewhat less sensitive to changes in the software. Additionally, integrating these results into a PRA may not be a trivial task

Model checking and Binary Decision Diagrams differ substantially from the other presented methods. Model checking is a method for system validation by exhaustively testing a model of a system. BDDs are data structures used to represent Boolean functions.

Model checking is a promising method for verifying safety I&C systems that enables exhaustive verification of systems with large state spaces [Björkman\_et\_al\_2009, Valkonen\_et\_al\_2007] compared to other methodologies. Model checking is not directly applicable for reliability assessment of digitalized I&C systems. However, model checking could be utilized in reliability assessment to formally verify the correctness of the constructed reliability models.

Binary decision diagram as such is not an approach for safety evaluation but rather are a way for representing Boolean functions. The BDD provides several advantages from the computational viewpoint. The use of BDDs as the data structure of reliability assessment methodologies have been studied in literature, and there already exist efficient software based on BDDs. However, most of the applications are related to static methodologies such as fault tree and event tree methodologies. The use of BDDs, for instance, in model checking indicates that BDDs can manage relatively large dynamic systems with reasonable computational effort. Therefore, BDDs could probably also be used as the underlying data structure of dynamic reliability analysis tools for evaluation of digital I&C systems to enhance computational efficiency.

## 12 Conclusions

Besides considering hardware and software, the reliability modelling of digital I&C systems should account for the dynamic interaction between the I&C system components and between the system and the controlled process. The limitation of the traditional fault tree methodology to capture time dependent dynamic behaviours have made the use of fault trees somewhat impractical in assessing the reliability of digitalized I&C systems. There exists a number of dynamic methodologies, but it seems that none of them can be used as such to model all relevant features of digital I&C systems.

All of the methods surveyed included unique features that could make them suitable for specific applications. If these features could be developed further or integrated with others models, the methods resulting from such modifications could lead to approaches that were able to model digital I&C systems in a quantitative manner with required level of detail. Whichever modelling technique is used, is should be able to at least fulfil the discussed requirements and include

all relevant system features and interactions required by current regulatory guidance.

It is reasonable to assume that the future PRA-models will still largely depend on the ET/FT-methodology. However, it is a necessity to develop PRA compatible dynamic reliability methodologies to account for dynamic features the static ET/FT-methodology is unable to capture.

Also the modelling interface needs to be considered. The modelling formalism should enable transparent verification of the reliability model that it represents the actual modelled target. Additionally, a graphical representation of the models that allowed, for instance, the illustration of time dependencies and feedback loops would be a useful feature.

Thus far, the scalability of dynamic reliability methodologies is somewhat inadequate. BDD algorithms have proven to be efficient in Boolean variable manipulation and they are utilizable in the computation of dynamic reliability models. Thus, the advancement of BDD computation algorithms could enable the calculation of increasingly complex dynamic reliability models.

## References

- [Akers\_1978] Akers, S.B., *Binary Decision Diagrams*, IEEE Trans. on Computers, Vol. C-27, 1978.
- [Aralia] Aralia User Manual, Arboost Technologies 2006 <http://www.arboost.com/aralia-page.htm>
- [Björkman\_et\_al\_2009] K. Björkman, J. Frits, J. Valkonen, K. Heljanko, I. Niemelä, Model-Based Analysis of a Stepwise Shutdown Logic - MODSAFE 2008 Work Report, VTT Working Papers 115, VTT Technical Research Centre of Finland, Espoo, Finland, March 2009, 42 p.
- Borysiewicz, M.J., Garanty, I., Kozubal, A. Part 5 – Assessment and Management of Risk, “Quantitative Risk Assessment (QRA)”, Monography Models and techniques for health and environmental hazard assessment and management”, Warsaw 2006.
- [Bozzano\_&\_Villafiorita\_2007] Bozzano, M., Villafiorita, A., The FSAP/NuSMV-SA Safety Analysis Platform; International Journal on Software Tools for Technology Transfer, Volume 9, Number 1, February 2007, pp. 5-24(20)
- [Brace et al 1990] Brace, K.L., Rudell, R.L., Bryant, R.E., Efficient Implementation of a BDD Package; 27th ACM/IEEE Design Automation Conference, 1990, Paper 3.1, 40-45; IEEE; 1990
- [Bryant\_1986] Bryant, R. E., Graph-Based Algorithms for Boolean Function Manipulation; IEEE Transactions on Computers C-35, 6(Aug), pp. 677-691; ; 1986
- [Bucci et al\_RESS\_08] Bucci, P., Kirschenbaum, J., Mangan, L. A., Aldemir, T., Smith, C., Wood, T., Construction of event-tree/fault-tree models from a Markov approach to dynamic system reliability; Reliability Engineering and System Safety 93 (2008) 1616–1627; ; 2008
- [BuDDy] BuDDy - A Binary Decision Diagram Package version 2.4 <http://sourceforge.net/projects/buddy/> 2004
- [Clarke et al 1999] E. M. Clarke, Jr., O. Grumberg, and D. A. Peled, *Model Checking*, The MIT Press, (1999)
- [Clarke et al\_1993] Clarke, E., Fujita, M., McGeer, P., McMillan, M., Yang, J., Zhao, X., Multi-Terminal Binary Decision Diagrams: An Efficient Data Structure for Matrix Representation; Proceedings of the International Workshop on Logic Synthesis, p. 1-15 1993; ; 1993
- [Cordier\_RESS\_1997] Cordier, C., Fayot, M., Leroy, A., Petit, A., Integration of process simulations in availability studies; Reliability Engineering and System Safety, Vol. 55, pp. 105-116; ; 1997
- [Coudert & Madre 1992] O. Coudert and J.C. Madre, Implicit and incremental computation of primes and essential primes of Boolean functions, *Proceedings of the 29th ACM/IEEE design automation conference, DAC'92* (1992).

- [CUDD] F. Somenzi. CUDD: CU Decision Diagram package. Public software, Colorado University, Boulder, 1997.
- [Digi\_1997] Digital Instrumentation and Control Systems in Nuclear Power Plants: Safety and Reliability Issues, Final Report, by Committee on Application of Digital Instrumentation and Control Systems to Nuclear Power Plant Operations and Safety, and National Research Council, National Academy Press Washington, D.C. 1997
- [Doguc\_RESS\_2009] O. Doguc, J. E. Ramirez-Marquez, A generic method for estimating system reliability using Bayesian networks, Reliability Engineering & System Safety, Volume 94, Issue 2, February 2009, Pages 542-550
- [Eom\_et\_al\_NPIC\_2009] Eom, H.-S., Park, G.-Y., Kang, H.-G., Jang, S.-C., Reliability Assessment Of A Safety-Critical Software By Using Generalized Bayesian Nets. Sixth American Nuclear Society International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies, NPIC&HMIT 2009, Knoxville, Tennessee, April 5-9, 2009, on CD-ROM, American Nuclear Society, LaGrange Park, IL 2009.
- [Fanti & Zhou 2004] Fanti, M.P., Zhou, M., Deadlock control methods in automated manufacturing systems, IEEE Transactions on Systems, Man and Cybernetics, Part A, vol. 34, issue 1, Jan. 2004, page(s): 522.
- [FT\_handbook-Aerospac] Vesely, W., Stamatelatos, M., Dugan, J., Fragola, J., Minarick III, J., Railsback, J., Fault Tree Handbook with Aerospace Applications; NASA; 2002
- [Garrett\_DFM\_1995] Garrett, C.J, Guarro, S.B., Apostolakis, G.E., The Dynamic Flowgraph Methodology for Assessing the Dependability of Embedded Software Systems; IEEE Trans. on Systems, Man and Cybernetics Vol. 25. No. 5, 824-840; IEEE; 1995
- [Gomes et al. 2004] Gomes, L., Barros, J.P., Lino, R., Addition of fault detection capabilities in automation applications using Petri nets; IEEE International Symposium on Industrial Electronics, 2004, vol. 1, 47 May 2004, page(s):645 – 650.
- [Gran\_2002] Gran B. The use of Bayesian Belief Networks for combining disparate sources of information in the safety assessment of software based systems. Department of Mathematical Sciences Norwegian University of Science and Technology, Dr.Ing. Thesis 2002.
- [Hamada et\_al\_2008 ] Hamada, M.S., Wilson, A., Reese, C.S., Martz, H.F., Bayesian Reliability, Springer Series in Statistics, Springer New York 2008, XVI, 436 p
- [Helminen\_2001] Helminen A. Reliability estimation of safety-critical software-based systems using Bayesian networks. STUK-YTO-TR 178. STUK, Helsinki 2001. <http://www.stuk.fi/julkaisut/tr/stuk-yto-tr178.pdf>
- [Helminen\_Pulkkinen\_2003] Helminen A, Pulkkinen U. Reliability assessment using Bayesian network. Case study on quantitative estimation of a software-based motor protection relay. STUK-YTO-TR 198. STUK, Helsinki 2003. <http://www.stuk.fi/julkaisut/tr/stuk-yto-tr198.pdf>
- [ISO8402] ISO 8402 Quality management and quality assurance – Vocabulary, International Organization for Standardization, Geneva. 1994.
- [Kara-Zaitri\_IJQRM\_96] Kara-Zaitri, C., An improved minimal cut set algorithm; International Journal of Quality & Reliability Management 1996, Vol13, Issue 2, p. 114 - 132; ; 1996
- [Karanta\_Maskuniitty\_2009] Karanta, I., Maskuniitty, M. Reliability of digital control systems in nuclear power plants — Modelling the feedwater system, VTT research report VTT-R-01749-08, VTT, Espoo, 2009.
- [Kelly\_RESS\_2009] D. L. Kelly, C. L. Smith Bayesian inference in probabilistic risk assessment—The current state of the art, Reliability Engineering & System Safety, Volume 94, Issue 2, February 2009, Pages 628-643
- [Koh\_NPIC\_2009] Koh, K. Y., Seong, P. H., SACS<sup>2</sup>: A Dynamic and Formal Approach to Safety Analysis for Complex Safety Critical Systems, Sixth American Nuclear Society International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies, NPIC&HMIT 2009, Knoxville, Tennessee, April 5–9, 2009, on CD-ROM, American Nuclear Society, LaGrange Park, IL 2009.
- [Labeau et al. 2000] Labeau, P.E., Smidts, C., Swaminathan, S., Dynamic reliability: towards an integrated platform for probabilistic risk assessment, Reliability Engineering and System Safety 68 (219254), Elsevier Science Limited 2000, 36p.
- [Lee\_et\_al\_1985] Lee, W.S., Grosh, D. L., Tillman, F.A., Lie, C.H. Fault tree analysis, methods, and applications —a review, IEEE Trans Reliab 34 (1985), p. 194-203.
- [Liu\_&\_Chiou\_1997] Liu, T.S., Chiou, S. B., The application of Petri nets to failure analysis; Reliability Engineering and System Safety, Vol 57, February 1997, Pp. 129-142; ; 1997

- [Lynce\_2005] Lynce, I., Propositional satisfiability: Techniques, algorithms and applications. Ph.D. thesis, IST, Technical University of Lisbon. 2005.
- [Minato\_IJCAI\_2007] Minato, S., Satoh, K., Sato, T., "Compiling Bayesian Networks by Symbolic Probability Calculation Based on Zero-suppressed BDDs," In Proc. of 20th International Joint Conference of Artificial Intelligence (IJCAI-2007), pp. 2550-2555. Jan. 2007.
- [Minato\_ZBDD\_1993] Minato, S., Zero-Suppressed BDDs for Set Manipulation in Combinatorial Problems; Proceedings of the 30th ACM/IEEE Design Automation Conference, pp. 272-277, June 1993.; ; 1993
- [Murphy\_2005] Murphy, K. Software Packages for Graphical Models / Bayesian Networks <http://people.cs.ubc.ca/~murphyk/Bayes/bnsoft.html> (last updated 2005) 2009
- [Musa\_Okumoto\_1984] Musa, J.D., Okumoto, K., "A Logarithmic Poisson Execution Time Model for Software Reliability Measurement," Proceedings of Seventh International Conference on Software Engineering, 230-238, Orlando, FL, 1984.
- [Männistö\_VTT\_2006] Männistö, I., Long Mission Time Reliability: pre-study, VTT Research Report, VTT-R-00862-06, 2006, 32p
- [NUREG/CR-6901] Aldemir, T., Miller, D.W., Stovsky, M.P., Kirschenbaum, J., Bucci, P. A.W. Fentiman I , L.T. Mangan I, Current State of Reliability Modeling Methodologies for Digital Systems and Their Acceptance Criteria for Nuclear Power Plant Assessments; NUREG/CR-6901;
- [NUREG-0492] Vesely, W.E., Goldberg, F.F., Roberts, N.H., Haasl, D.F., Fault tree handbook; NUREG-0492; U.S.NRC; 1981
- [NUREG/CR-6942] Aldemir, T., Stovsky, M.P., Kirschenbaum, J., Mandelli, D., Bucci, P., Mangan, L.A., Miller, D.W., Sun, X., Ekici, E., Guarro, S., Yau, M., Johnson, B., Elks, C., Arndt, S.A., Dynamic Reliability Modeling of Digital Instrumentation and Control Systems for Nuclear Reactor Probabilistic Risk Assessments; NUREG/CR-6942; U.S.NRC;
- [NUREG/CR-6962] Chu, T.L., Martinez-Guridi, G., Yue, M., Lehner, J. and Samanta, P. Traditional Probabilistic Risk Assessment Methods for Digital Systems; NUREG/CR-6962, BNL-NUREG-80141-2008; U.S.NRC; 2008
- [NUREG/CR-6985] Aldemir, T., Guarro, S., Kirschenbaum, J., Mandelli, D., Mangan, L.A., Bucci, P., Yau, M., Johnson, B., Elks, C., Ekici, E., Stovsky, M.P. Miller, D.W., Sun, X., Arndt, S.A., Nguyen, Q. Dion, J., A Benchmark Implementation of Two Dynamic Methodologies for the Reliability Modeling of Digital Instrumentation and Control Systems; NUREG/CR-6985; U.S.NRC; 2009
- [NuSMV] NuSMV Model Checker v.2.4.3. <http://nusmv.irst.itc.it/> (2009).
- [Pearl\_1988] Pearl, J. *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. Morgan Kaufmann Publishers, San Mateo, CA, 1988.
- [Petri\_Net\_World] Petri Nets World: Complete Overview of Petri Nets Tools Database [http://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/complete\\_db.html](http://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/complete_db.html) 2009
- [PRISM] PRISM Probabilistic Model Checker <http://www.prismmodelchecker.org/> 2009
- [Rausand\_&\_Høyland\_2004] Rausand, M., Høyland, A., System reliability theory: models and statistical methods, John Wiley & Sons, Hoboken, NJ. 2004. - xix, 636 p
- [Rauzy\_2001] Rauzy, A., Mathematical Foundation of Minimal Cutsets; IEEE Transactions on Reliability, 50(4), 389-396, december 2001; IEEE; 2001
- [Rauzy\_FTA\_1993] Rauzy, A., New Algorithms for Fault Trees Analysis; Reliability Engineering and System Safety 40 (1993) 203-211; Elsevier; 1993.
- [Samaniego\_2007] Samaniego, F. J., System Signatures and their Applications in Engineering Reliability, International Series in Operations Research & Management Science, Vol. 110, 154p. Springer, 2007.
- [Sanner\_IJCAI\_2005] Sanner, S., McAllester, D., "Affine algebraic decision diagrams (aadds) and their application to structured probabilistic inference," In *Proceedings of the 19th International Joint Conference on AI (IJCAI-05) 2005*.
- [Schneidewind\_IEEE\_1992] Schneidewind, N.F., Keller, T.W.; "Applying Reliability Models to the Space Shuttle," IEEE Software, 28-33, July 1992.
- [Sigurdsson\_2001] Sigurdsson J.H., Walls L., Quigley J. Bayesian Belief Nets for managing expert judgement and modelling reliability, Quality and Reliability International 17 p. 181-190. (2001)

- [Tang\_RAMIS\_2004] Tang, Z., Dugan, J. B., Minimal Cut Set/Sequence Generation for Dynamic Fault Trees; Reliability and Maintainability, 2004 Annual Symposium - RAMS , vol., no., pp. 207-213, 26-29 Jan. 2004;
- [Tang\_Dugan\_IEEE\_2006] Tang, Z., Dugan, J.B., "BDD-based reliability analysis of phased-mission systems with multimode failures," *Reliability, IEEE Transactions on* , vol.55, no.2, pp. 350-360, June 2006
- [Uppaal] Uppaal integrated tool environment v. 4.0.6, <http://www.uppaal.com/> (2009).
- [Valkonen\_et\_al\_2007] J. Valkonen, V. Pettersson, K. Björkman, J.-E. Holmberg, M. Koskimies, K. Heljanko, and I. Niemelä, Model-Based Analysis of an Arc Protection and an Emergency Cooling System - MODSAFE 2007 Work Report. VTT Working Papers 93, VTT Technical Research Centre of Finland, Espoo, Finland, February 2008, 51 p.
- [Wakerly\_2000] Wakerly, J. F., Digital design : principles and practices; Upper Saddle River, NJ : Prentice Hall, 2000;
- [Way\_RESS\_2000] Way, Y.-S., Hsia, D-Y., A simple component-connection method for building binary decision diagrams encoding a fault tree; *Reliability Engineering and System Safety* 70 (2000) 59–70; ; 2000
- [Wilson\_RESS\_2007] Wilson, A., Huzurbazar, A. Bayesian networks for multilevel system reliability'. *Reliability Engineering & System Safety* **92**(10):1413-1420. (2007).
- [Yang\_NPIC\_2009] Yang, Y., A Flow Network Model For Software Reliability Assessment. Sixth American Nuclear Society International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies, NPIC&HMIT 2009, Knoxville, Tennessee, April 5-9, 2009, on CD-ROM, American Nuclear Society, LaGrange Park, IL 2009.
- [Zang\_et\_al\_IEEE\_1999] Zang, X., Sun, H., Trivedi, K. S., "A BDD-based algorithm for reliability evaluation of phased mission systems," *IEEE Transactions on Reliability* vol. 48, pp. 50–60, 1999.
- [Zhenjuan et al. 2006] Zhenjuan, L., Bo, Pan., Hongguan, L., Batch Process Fault Diagnosis Based on Fuzzy Petri Nets, ICICIC '06 First International Conference on Innovative Computing, Information and Control, 2006, vol. 2, 3001 Aug. page(s): 474-477.
- [Ziv\_ICSM\_1997] Ziv, H.; Richardson, D.J., "Constructing Bayesian-network models of software testing and maintenance uncertainties," *Proceedings., International Conference on Software Maintenance, 1997.*, vol., no., pp.100-109, 1-3 Oct 1997
- [Zurawski & Zhou 1994] Zurawski, R., Zhou, M., Petri nets and industrial applications: A tutorial, *IEEE Transactions on Industrial Electronics*, vol. 41, iss.6, Dec 1994, pages: 567-583.



## APPENDIX 1. Reliability Analysis Software

Software	Methodology	Properties	www
Risk Spectrum	FTA	Event tree-fault tree calculation. Includes also a BDD tool.	<a href="http://www.scandpower.com/en/risk/">http://www.scandpower.com/en/risk/</a>
ARALIA	FTA	A solver for for Boolean reliability models (e.g. fault trees, block diagrams, event trees). Uses BDDs (ZDD) as the data structure. Used by some reliability calculation software, e.g. item software	<a href="http://www.arboost.com/aralia-page.htm">http://www.arboost.com/aralia-page.htm</a>
STUK PSA	FTA	Dynamic event tree calculation tool based on ET-FT calculation and Monte Carlo simulation. So far, available for DOS environment. In the windows version (FinPSA) only ET-FT calculation. The minimal cut set algorithm is based on solution of path net, and it contains features like logic optimization, predictive minimization, complete and partial modularization, dependent sub-trees, and optimal substitution order	e.g. <a href="http://www.stuk.fi/ydinturvallisuus/ydinvoimalaitokset/en_GB/finpsa/_print/">http://www.stuk.fi/ydinturvallisuus/ydinvoimalaitokset/en_GB/finpsa/_print/</a>
DFM Software Toolset	DFM	Tool for DFM calculation. A beta-version of the tool is available for a trial period	<a href="http://www.ascainc.com/">http://www.ascainc.com/</a>
NuSMV	Model Checking	NuSMV is a symbolic model checker. The state space is represented symbolically and explored by using BDDs. In addition, SAT (propositional satisfiability) based bounded model checking is supported.	<a href="http://nusmv.irst.itc.it/">http://nusmv.irst.itc.it/</a>
CUDD	BDD	CUDD is a package for the manipulation of Binary Decision Diagrams (BDDs), Algebraic Decision Diagrams (ADDs) and Zero-suppressed Binary Decision Diagrams developed. CUDD package is in use by, e.g., NuSMV	<a href="http://vlsi.colorado.edu/~fabio/CUDD/">http://vlsi.colorado.edu/~fabio/CUDD/</a>
SAPHIRE	FTA	SAPHIRE (Systems Analysis Programs for Hands-on Integrated Reliability Evaluations) is a probabilistic risk and reliability assessment software tool.	<a href="http://saphire.inl.gov">http://saphire.inl.gov</a> , <a href="http://saphiresoftware.com">http://saphiresoftware.com</a>
Reliasoft	FTA	Reliasoft provides tools for e.g. Reliability Block Diagram and Fault Tree analysis	<a href="http://www.reliasoft.com/index.html">http://www.reliasoft.com/index.html</a>
Relex	FTA	The Relex Software product line contains e.g. the following reliability analysis tools: Reliability Block Diagram, FMEA/FMECA, Fault Tree/Event Tree, Human Factor Risk Analysis, and Markov. Have some activities around BDDs	<a href="http://www.relex.com/index.asp">http://www.relex.com/index.asp</a>
HUGIN	BBN	The Hugin tool is based on Bayesian Networks. Not primarily used for reliability analysis	<a href="http://www.hugin.com/">http://www.hugin.com/</a>
AgenaRisk	BBN	A Bayesian network and simulation software for Risk Analysis and Decision support	<a href="http://www.agenarisk.com/">http://www.agenarisk.com/</a>
Bnet.Builder	BBN	Not primarily used for reliability analysis	<a href="http://www.cra.com/commercial-solutions/belief-network-modeling.asp">http://www.cra.com/commercial-solutions/belief-network-modeling.asp</a>

Software	Methodology	Properties	www
Itemsoftware	FTA	Itemsoftware provides a variety of different reliability analysis tools, such as, reliability block diagram, fault tree, event tree and markov. BDDs utilised in fault tree analysis	<a href="http://www.itemsoft.com/">http://www.itemsoft.com/</a>
Isograph	FTA	Itemsoftware provides a variety of different reliability analysis tools, such as, reliability block diagram, fault tree, event tree and markov. Utilises BDDs in fault tree analysis	<a href="http://www.isograph-software.com/index.htm">http://www.isograph-software.com/index.htm</a>
RiskMan	FTA	Event tree fault tree analysis. BDDs used in fault tree analysis. Includes e.g. Monte Carlo simulation and Bayesian analysis	<a href="http://www.absconsulting.com/riskmansoftware/index.html">http://www.absconsulting.com/riskmansoftware/index.html</a>
SPIN	Model Checking	A software tool for formal verification of distributed software systems.	<a href="http://spinroot.com/spin/whatispin.html">http://spinroot.com/spin/whatispin.html</a>
UPPAAL	Model Checking	Uppaal is an integrated tool environment for modeling, validation and verification of real-time systems modeled as networks of timed automata	<a href="http://www.uppaal.com/">http://www.uppaal.com/</a>
CAFTA	FTA	Fault Tree and event tree analysis	<a href="http://my.epri.com/portal/server.pt?">http://my.epri.com/portal/server.pt?</a>
Collection of software	BBN	Software Packages for Graphical Models / Bayesian Networks	<a href="http://people.cs.ubc.ca/~murphyk/Bayes/bnsoft.html">http://people.cs.ubc.ca/~murphyk/Bayes/bnsoft.html</a>
Collection of software	BBN	Bayesian software	<a href="http://www.mas.ncl.ac.uk/~ndjw1/bk2site/Stats/Software-Statistical_computing/Bayesian_software/index.html">http://www.mas.ncl.ac.uk/~ndjw1/bk2site/Stats/Software-Statistical_computing/Bayesian_software/index.html</a>
Collection of software	Petri Net	Overview of Petri Nets Tool	<a href="http://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/complete_db.html">http://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/complete_db.html</a>