Department of Mathematics and Systems Analysis

Theoretical and methodological extensions to dynamic reliability analysis

Tero Tyrväinen

Aalto University

DOCTORAL DISSERTATIONS

Aalto University publication series
DOCTORAL DISSERTATIONS 154/2017
VTT SCIENCE 161

Theoretical and methodological extensions to dynamic reliability analysis

Tero Tyrväinen

A doctoral dissertation completed for the degree of Doctor of Science (Technology) to be defended, with the permission of the Aalto University School of Science, at a public examination held at the lecture hall H304 of the school (Otakaari 1, 02150 Espoo, Finland) on 13 October 2017 at 12 noon.

Aalto University
School of Science
Department of Mathematics and Systems Analysis
Systems Analysis Laboratory

Supervising professor
Professor Ahti Salo, Aalto University School of Science, Finland

Thesis advisors
Dr. Jan-Erik Holmberg, Risk Pilot AB, Finland
Professor Ahti Salo, Aalto University School of Science, Finland

Preliminary examiners
Professor Tunc Aldemir, Ohio State University, USA
Professor Lixuan Lu, University of Ontario, Institute of Technology, Canada

Opponent
Professor Tim Bedford, University of Strathclyde, United Kingdom

Aalto University publication series
DOCTORAL DISSERTATIONS 154/2017

VTT SCIENCE 161

ISBN 978-952-60-7571-6 (printed)
ISBN 978-952-60-7570-9 (pdf)
ISSN-L 1799-4934
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)
http://urn.fi/URN:ISBN:978-952-60-7570-9

ISBN 978-951-38-8565-6 (printed)
ISBN 978-951-38-8564-9 (pdf)
ISSN-L 2242-119X
ISSN 2242-119X (printed)
ISSN 2242-1203 (pdf)
http://urn.fi/URN:ISBN:978-951-38-8564-9

Unigrafia Oy
Helsinki 2017

Finland

441 697
Printed matter

Aalto University

Abstract

Aalto University, P.O. Box 11000, FI-00076 Aalto www.aalto.fi

Author
Tero Tyrväinen

Name of the doctoral dissertation
Theoretical and methodological extensions to dynamic reliability analysis

Publisher School of Science

Unit Department of Mathematics and Systems Analysis

Series Aalto University publication series DOCTORAL DISSERTATIONS 154/2017

Field of research Systems and Operations Research

Manuscript submitted 9 June 2017 **Date of the defence** 13 October 2017

Permission to publish granted (date) 16 August 2017 **Language** English

☐ **Monograph** ☒ **Article dissertation** ☐ **Essay dissertation**

Abstract

Rigorous analysis of the reliability of a dynamic system calls for modelling of the dynamic behaviour of the system and its interactions. However, traditional and the most frequently used reliability analysis methods, such as fault tree analysis, are static and have only limited capability to represent dynamic systems. Therefore, dynamic reliability analysis methods have been studied since 1990s.

Dynamic flowgraph methodology (DFM) is a method for the reliability analysis of dynamic systems containing feedback loops. A DFM model is a dynamic graph representation of the analysed system. DFM has been most often applied to different digital control systems. One reason for this is that a DFM model can represent the interactions between a control system and the controlled process.

The main goal of DFM analysis is to identify prime implicants, which are minimal combinations of events and conditions that cause the analysed top event, for example, system failure. This dissertation strengthens the mathematical foundation of DFM by developing an improved definition of a prime implicant.

Risk importance measures can be used to identify components and basic events that are most important for the reliability of the system. This dissertation develops new dynamic risk importance measures as generalisations of two traditional risk importance measures for the needs of DFM. Unlike any other importance measure, the dynamic risk importance measures utilise all the information available in prime implicants of DFM. They primarily measure the importances of different states of components and variables of a DFM model. The computation of the dynamic risk importance measures for failure states of components provides significant additional information compared to other importance values.

This dissertation also examines common cause failures (CCFs) in dynamic reliability analysis. Taking CCFs into account is important when modelling systems with redundancies. The dissertation extends the DFM by presenting CCF models that take failure times of components into account.

Keywords Reliability analysis; dynamic system; risk importance measure; common cause failure; prime implicant; digital control system

ISBN (printed) 978-952-60-7571-6 **ISBN (pdf)** 978-952-60-7570-9

ISSN-L 1799-4934 **ISSN (printed)** 1799-4934 **ISSN (pdf)** 1799-4942

Location of publisher Helsinki **Location of printing** Helsinki **Year** 2017

Pages 111 **urn** http://urn.fi/URN:ISBN:978-952-60-7570-9

A! Aalto-yliopisto

Tiivistelmä

Aalto-yliopisto, PL 11000, 00076 Aalto www.aalto.fi

Tekijä
Tero Tyrväinen

Väitöskirjan nimi
Teoreettisia ja menetelmällisiä laajennuksia dynaamiseen luotettavuusanalyysiin

Julkaisija Perustieteiden korkeakoulu

Yksikkö Matematiikan ja systeemianalyysin laitos

Sarja Aalto University publication series DOCTORAL DISSERTATIONS 154/2017

Tutkimusala Systeemi- ja operaatiotutkimus

Käsikirjoituksen pvm 09.06.2017 **Väitöspäivä** 13.10.2017

Julkaisuluvan myöntämispäivä 16.08.2017 **Kieli** Englanti

☐ **Monografia** ☒ **Artikkeliväitöskirja** ☐ **Esseeväitöskirja**

Tiivistelmä

Dynaamisen järjestelmän luotettavuuden tarkka analyysi vaatii järjestelmän dynaamisen käyttäytymisen ja vuorovaikutusten mallintamista. Kuitenkin perinteiset ja useimmin käytetyt luotettavuusanalyysimenetelmät, kuten vikapuuanalyysi, ovat staattisia ja niiden sopivuus dynaamisten järjestelmien kuvaamiseen on rajallinen. Siksi dynaamisen luotettavuusanalyysin menetelmiä on tutkittu 1990-luvulta lähtien.

Dynaaminen vuokaaviomallintaminen on menetelmä takaisinkytkentöjä sisältävien dynaamisten järjestelmien luotettavuusanalyysiin. Dynaaminen vuokaaviomalli on dynaaminen verkkoesitys analysoidusta järjestelmästä. Dynaamista vuokaaviomallinnusta on useimmin sovellettu erilaisiin digitaalisiin ohjausjärjestelmiin. Yksi syy tälle on, että dynaaminen vuokaaviomalli pystyy kuvaamaan ohjausjärjestelmän ja ohjattavan prosessin väliset vuorovaikutukset.

Päätavoite dynaamisessa vuokaaviomallinnuksessa on tunnistaa minimitermit (prime implicants), jotka ovat tapahtumien ja tilojen minimaalisia yhdistelmiä, jotka aiheittavat tarkasteltavan huipputapahtuman, esimerkiksi järjestelmän vikaantumisen. Tämä väitöskirja vahvistaa dynaamisen vuokaaviomallintamisen matemaattista perustaa kehittämällä paremman määritelmän minimitermille.

Riskitärkeysmittoja voidaan käyttää järjestelmän luotettavuuden kannalta tärkeimpien komponenttien ja perustapahtumien tunnistamiseen. Tämä väitöskirja kehittää uudet dynaamiset riskitärkeysmitat yleistyksinä kahdesta perinteisestä riskitärkeysmitasta dynaamisen vuokaaviomallinnuksen tarpeisiin. Toisin kuin mikään muu tärkeysmitta, dynaamiset riskitärkeysmitat hyödyntävät kaiken dynaamisen vuokaaviomallinnuksen minimitermeihin sisältyvän tiedon. Ne mittaavat ensisijaisesti dynaamisen vuokaaviomallin komponenttien ja muuttujien eri tilojen tärkeyksiä. Dynaamisten riskitärkeysmittojen laskenta komponenttien vikatiloille antaa merkittävää lisätietoa verrattuna muihin tärkeysarvoihin.

Tämä väitöskirja tutkii myös yhteisvikoja dynaamisessa luotettavuusanalyysissä. Yhteisvikojen huomioiminen on tärkeää redundansseja sisältävien järjestelmien mallinnuksessa. Väitöskirja laajentaa dynaamista vuokaaviomallinnusta esittämällä yhteisvikamalleja, jotka huomioivat komponenttien vikaantumisten ajankohdat.

Avainsanat Luotettavuusanalyysi; dynaaminen järjestelmä; riskitärkeysmitta; yhteisvika; minimitermi; digitaalinen ohjausjärjestelmä

ISBN (painettu) 978-952-60-7571-6 **ISBN (pdf)** 978-952-60-7570-9

ISSN-L 1799-4934 **ISSN (painettu)** 1799-4934 **ISSN (pdf)** 1799-4942

Julkaisupaikka Helsinki **Painopaikka** Helsinki **Vuosi** 2017

Sivumäärä 111 **urn** http://urn.fi/URN:ISBN:978-952-60-7570-9

Preface

This dissertation would not have been possible without my instructor Jan-Erik Holmberg. I would like to thank him for the interesting topic and all the guidance during these six years. I would also like to thank professor Ahti Salo for supervising and guiding the work. My co-worker and Master's thesis instructor Kim Björkman also gave me remarkable support, especially at the early phase of the work and concerning software implementations. All in all, our co-operation has been very fruitful throughout the years. In addition, I would like to thank those who have provided useful comments to my manuscripts, notably Antti Toppila and Ilkka Karanta. I would also like to thank my opponent professor Tim Bedford and preliminary examiners professor Tunc Aldemir and professor Lixuan Lu.

Large part of the work of this dissertation was conducted in The Finnish Research Programme on Nuclear Power Plant Safety 2011-2014 (SAFIR-2014) and some of the writing was also performed in SAFIR2018. Most of the work was performed in the SARANA project led by Janne Valkonen. I would finally like to thank SAFIR programme director Jari Hämäläinen and former director Kaisa Simola, because they are actually the persons, along with Jan-Erik Holmberg, who originally gave me this job opportunity at VTT.

Espoo, August 24, 2017,

Tero Tyrväinen

Contents

List of Publications

This thesis consists of an overview and of the following publications which are referred to in the text by their Roman numerals.

I Tyrväinen, T. Prime implicants in dynamic reliability analysis. *Reliability Engineering and System Safety*, Vol. 146, pp. 39-46, doi: 10.1016/j.ress.2015.10.007, February 2016.

II Tyrväinen, T. Risk importance measures in the dynamic flowgraph methodology. *Reliability Engineering and System Safety*, Vol. 118, pp. 35-50, doi: 10.1016/j.ress.2013.04.013, October 2013.

III Tyrväinen, T. Common cause failures in the dynamic flowgraph methodology. *Manuscript*, 19+17 pages, 2017.

Author's Contribution

Tyrväinen is the sole author in all the papers contained in the Dissertation.

1. Introduction

1.1 Background

Comprehensive risk analysis [1] of a complex system calls for a systematic identification of all significant accident scenarios, and assessment of the likelihood and consequences of each accident scenario. Probabilistic risk analysis (PRA) [2] is an approach that has been widely used in the risk analysis of complex facilities, especially nuclear power plants. Compared to qualitative risk analysis methods [3], PRA is more precise and more effective when a large number of complex accident sequences needs to be analysed. Usually, the main purpose of PRA is to support risk-informed decision making and to help fulfil regulatory requirements. PRA can point out the weaknesses of the analysed system, and the system's reliability can be improved effectively on the basis of the results of PRA.

PRA modelling is generally performed using fault trees [4, 5] and event trees [2]. An event tree models how an accident can progress from an initiating event to different consequences depending on a set of nodal questions e.g. about which safety systems fail. The probabilities of different event tree branches can be calculated using fault trees. A fault tree typically represents the failure logic of the analysed system and thus helps to determine which component failure and event combinations can cause the system to fail. The probability of the system's failure can be calculated from the fault tree if the probabilities of its basic events (e.g. component failures) are known. This computation is usually based on minimal cut sets [4], which are minimal combinations of basic events that can cause the top event, such as failure of the system. The probabilities can be estimated on the basis of operational data or determined by expert judgement.

Fault tree analysis is currently the leading method for risk and reliability analysis of complex systems. However, fault trees are static and have only limited capability to represent dynamic systems such as digital control systems. Dynamic interactions between software and hardware or interactions between the control system and the controlled process cannot be modelled properly using fault trees. In addition, fault trees do not support non-binary logic or modelling of the system's evolution in time. This has motivated the extensive development of dynamic reliability analysis methods since the 1990s [6].

Dynamic flowgraph methodology (DFM) is a method for the reliability analysis of dynamic systems containing feedback loops [7, 8, 9]. As in fault tree analysis, the aim of DFM is to identify which conditions can cause a top event, which can be e.g. system failure. A DFM model is a graph representation of the analysed system. The components of DFM models are analysed at discrete time points, and they can have multiple states. DFM has most often been applied to different digital control systems that include both hardware and software components. One reason for this is that a DFM model can represent the interactions between a control system and the controlled process.

The main alternative to DFM is the methodology that combines Markov modelling and cell-to-cell mapping (CTCM) technique [10, 11, 12]. Markov models can represent the dynamic and multi-state logic of a system to a degree of accuracy that is comparable to DFM. The main difference is that every state transition is associated with a probability in Markov models. Other dynamic reliability and risk analysis methods include dynamic event trees [13, 14, 15], Petri nets [16], event sequence diagrams [17], GO-FLOW methodology [18] and dynamic fault trees [19]. In addition, there are some Monte Carlo simulation based methods for the reliability analysis of digital instrumentation and control systems [20, 21, 22].

Broadly viewed, there are two main approaches to analyse the reliability of a dynamic system:

1. to simulate the system (inductive analysis),
2. to identify different ways in which the system can fail, e.g. minimal cut sets, and to determine the system's failure probability based on this kind of logical analysis (deductive analysis).

Some dynamic models, such as DFM and Markov models, can be solved in both ways. Markov models have been used more for inductive analysis,

whereas DFM has been considered more suitable for deductive analysis.

Although risk importance measures [23] and common cause failures [24] are important areas of reliability theory, they have not been studied much in the context of DFM. Risk importance measures can be used to identify components and basic events that are most important with regard to the system's reliability. The importance of a component depends both on the reliability of the component and the consequences that its failure would have on the system's reliability. Risk importance values help to determine how the system's reliability can best be improved.

A common cause failure (CCF) means that multiple components fail due to a common cause [25]. A CCF can occur between components that share some failure mechanism which can cause them to fail simultaneously or during a relatively short time window, for instance during the PRA mission time which is usually 24 hours. Modelling CCFs is an important part of the reliability analysis of complex systems including redundant components. If CCFs are not taken into account, the risk of the system's failure can be underestimated.

1.2 Objectives

This dissertation develops new risk importance measures for DFM. Traditional risk importance measures have been developed for binary and static logic, meaning that they cannot directly be applied in DFM. Some risk importance measures have previously been developed for DFM [26, 27], but they have limitations. For example, they cannot fully measure the importances of different failure modes of components, because they are not formulated for the states of nodes of DFM. Neither do they consider information about the timings of events and conditions properly. This motivates the development of new risk importance measures for measuring the importances of states of components and for taking the time aspect of DFM into account. The new importance measures also need to support the interpretation of results.

Another objective of this dissertation is to study CCFs in the DFM context, which has not been addressed in the earlier literature. Model in [28] included CCFs, but they were not really discussed in the paper. Approaches for the modelling of CCFs as well as the computation of CCF probabilities are developed. Compared to static analysis, DFM introduces a new dimension in that it considers the failure times of components. It

needs to be considered how the failure times should be taken into account in the CCF modelling and the calculation of probabilities.

One important basis for risk importance measure calculation and CCF modelling is the interpretation of DFM results. The primary result of DFM analysis is a set of prime implicants [29], which are minimal combinations of events and conditions that are sufficient to cause the top event. The interpretation of prime implicants is not always completely clear and unambiguous, for example when non-repairable components are considered as identified in the author's MSc thesis [30]. Therefore, the definition and interpretation of prime implicants are also studied in this dissertation.

2. Methodological background

The most often used reliability analysis method, fault tree analysis, is summarised in Section 2.1. Section 2.2 presents binary decision diagrams, because they are a commonly used method to solve a reliability model and they have been applied to DFM analysis. Markov modelling is described briefly in Section 2.3 because it is the most often used dynamic reliability analysis method and the main alternative to DFM. Sections 2.4-2.6 present DFM, risk importance measures and CCFs as the main background for the results of the dissertation, presented in Chapter 3.

2.1 Fault tree analysis

Fault tree analysis [4, 5] is a widely used method to estimate the failure probability of a system. A fault tree represents the ways in which the system can fail. It is a graphical tree structure in which basic events (component failures and other events that can cause the system to fail) are connected using logical gates, such as OR and AND; there are equivalent Boolean operations [31] (+ and $\cdot$).

A fault tree is typically used to identify minimal cut sets [4]. A cut set is a set of basic events that causes the top event which represents the system's failure. A minimal cut set is a cut set that contains the minimal number of basic events. Thus if one of the basic events is removed from the minimal cut set, it is not a cut set any more and the system does not fail.

The probability of the top event can be calculated on the basis of minimal cut sets and the probabilities of basic events [32]. It is also possible to calculate the top event probability directly from the fault tree without the identification of minimal cut sets. On the other hand, the analysis can also focus only on the identification of minimal cut sets if the probability

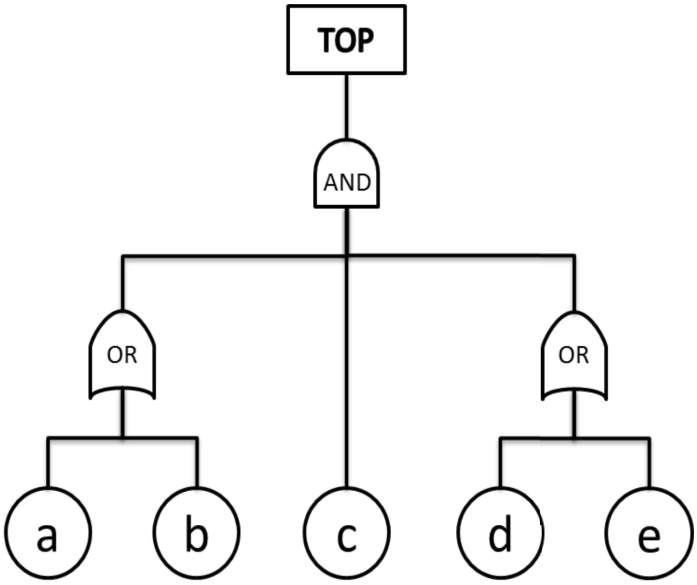

Figure 2.1. A fault tree with five basic events.

estimates of basic events are not available. Minimal cut sets themselves are useful qualitative information.

An example of a fault tree is presented in Figure 2.1. Boolean formula representation of the fault tree is $F = (a+b)\cdot c\cdot(d+e) = acd+ace+bcd+bce$. The minimal cut sets of the fault tree are acd, ace, bcd and bce.

2.2 Binary decision diagrams

A binary decision diagram (BDD) [33, 34] is an efficient data structure for symbolic Boolean manipulation. It is a directed acyclic graph that consists of decision nodes, two kinds of edges, 0-edges and 1-edges, and terminal nodes, 1-terminal and 0-terminal. In a BDD, each decision node, representing a Boolean variable, has a 0-edge and a 1-edge. When a BDD represents a Boolean formula, each path from the root node to the 0-terminal or the 1-terminal represents a Boolean assignment.

BDDs are based on repeated application of the Shannon expansion formula [34]

$$F = x \cdot F|_{x=1} + \overline{x} \cdot F|_{x=0}, \tag{2.1}$$

where F is a Boolean formula, x is a Boolean variable, $\overline{x}$ is the negation of variable x, $F|_{x=1}$ is Boolean formula F with condition that $x = 1$, and $F|_{x=0}$ is Boolean formula F with condition that $x = 0$. For example, if

$F = abc + ad$, F can be presented in form

$$\begin{aligned} F =& a \cdot (bc + d) \\ =& a \cdot (b \cdot (c + d) + \overline{b} \cdot d) \\ =& a \cdot (b \cdot (c + \overline{c} \cdot d) + \overline{b} \cdot d). \end{aligned}$$

Reliability analysis has been one application area of BDDs [35, 36, 37]. For example, a fault tree can be transformed into a BDD. Minimal cut sets can be generated from a BDD representing a fault tree, or the top event probability can directly be calculated from the BDD. In a BDD, each path ending in the 1-terminal also corresponds to a cut set, and these cut sets are mutually exclusive, which is an advantage compared to fault trees.

2.3 Markov modelling

Markov modelling is described briefly in this section because it is the most often used dynamic reliability analysis method and the main alternative to DFM. A Markov model consists of a set of system states, and transition rates between the states [38]. A Markov model is usually analysed in discrete time steps. The probabilities of different system states at a time step can be calculated on the basis of the probabilities of the states at the previous time step and the state transition rates. Typically, initial probabilities are defined for the system states, whereafter probabilities for how the states evolve in time are calculated.

The methodology of Markov/CTCM [10, 11, 12] has been applied to the reliability analysis of dynamic systems. In CTCM, the variables of the analysed system are discretised to a finite number of states. The variables can, for example, be physical variables such as water level or represent states of components such as a valve. States of different variables are combined to form state combinations called *cells*. These cells are then used as system states in a Markov model. The transition rates between the cells are determined e.g. on the basis of physical equations, system design and estimated failure rates of components. From the model, it is possible to analyse the ways in which some postulated top event can occur [12, 39] and how probable this event is or, alternatively, how the system evolves on the basis of some initial conditions [8, 11].

2.4 Dynamic flowgraph methodology

A DFM model [7, 8, 9, 40, 41, 42, 43, 44] is a graph representation of the analysed system. Nodes in the model represent the components and variables of the system, and edges connecting the nodes represent causal and other dependencies between the nodes. These dependencies can involve time delays, and nodes can have two or more states. If a node does not depend on any other node, it is a stochastic node, the state of which is determined by a discrete probability distribution at each time step. The state of a deterministic node is determined on the basis of the states of input nodes. Each deterministic node has a decision table which specifies the output state for each state combination of the input nodes. Decision tables can be constructed on the basis of empirical knowledge about the system, physical equations, simulations, expert judgement, software design or software code.

Figure 2.2 shows a simple DFM model of a tank system with a valve that is controlled on the basis of water level measurement, and Table 2.1 gives the decision table of node V as an example. In the model, node V represents the functional state of a valve (state 0 for closed and 1 for open), L represents the water level and M represents the water level measurement value. Nodes M and L have three states -1, 0 and 1 indicating water levels low, medium and high. Nodes S and F are stochastic nodes determining whether the water level measurement and the valve have failed. A row in the decision table specifies a combination of states of the input nodes, and the corresponding state of the output node. Delays in the dependencies are shown in the time lag row. Table 2.1 can be interpreted so that the valve is stuck in its previous state if it has failed (F is 1). Otherwise, the valve is opened if the water level measurement has a high value and closed if the water level measurement has a low value.

The primary target of DFM is usually to identify prime implicants of the top event [29]. An implicant is a combination of conditions that causes the top event, and a prime implicant is a minimal combination of conditions that is sufficient to cause the top event. In DFM, these conditions are represented by literals. In this context, a literal is a triplet consisting of a variable V, state s and time point $-t$, and denoted as $V_s(-t)$. A literal can, for example, represent a value of a physical variable or a state of a component, or indicate the occurrence of some event at a particular time step. Prime implicants of DFM can be interpreted as multi-state and

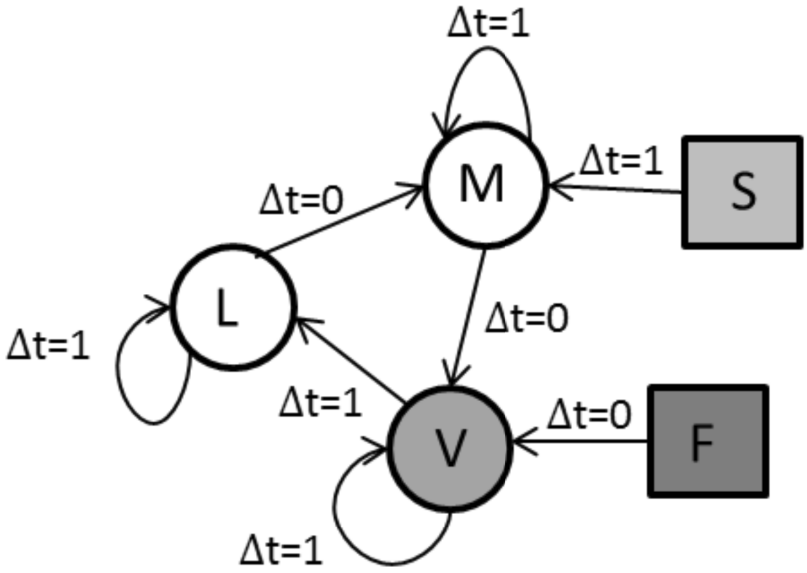

Figure 2.2. A DFM model.

Table 2.1. The decision table of node V.

	Output	Inputs		
Node	V	F	M	V
Time lag		0	0	1
	0	0	−1	0
	0	1	−1	0
	0	0	0	0
	0	1	0	0
	1	0	1	0
	0	1	1	0
	0	0	−1	1
	1	1	−1	1
	1	0	0	1
	1	1	0	1
	1	0	1	1
	1	1	1	1

timed minimal cut sets. Generally, prime implicants are an extension of minimal cut sets for non-coherent logic [35]. Paper I of this dissertation concerns the mathematical definition of a prime implicant.

The top event is also defined as a set of literals. The analyst can freely

choose any top event. Therefore, it is possible to analyse several top events in parallel, and both success and failure scenarios can be analysed.

A DFM model is typically analysed by tracing event sequences backwards from effects to causes [7]. Deductive analysis starts from the top event. The model is traced backwards in the cause-and-effect flow to identify what states of variables produce the top event. The process ends when the initial time step is reached. Prime implicants of a top event can contain initial states of deterministic nodes and states of stochastic nodes at any time step.

DFM analysis does not always have an unambiguous solution, because there are different ways to interpret some literals, prime implicants and constraints related to literals, and to handle the initial time step. Issues related to the interpretation of prime implicants, literals and literal constraints are studied in Paper I of this dissertation.

DFM models can also be analysed inductively by simulating the model with particular initial conditions [45]. All the possible consequences of the system's initial or boundary conditions are generated. The initial or boundary conditions can either be desired or undesired states. If these conditions are desired states, inductive analysis can be used to verify system requirements with the aim of ensuring that operation under normal conditions does not lead to undesired states. If these conditions are undesired states, inductive analysis can be used to verify the system's safety behaviour. Inductive analysis can be used, for example, to analyse the prime implicants identified in deductive analysis in more detail, and to examine the effects of mitigation actions.

DFM involves key concepts such as process node, condition node, causality edge, condition edge, transfer box and transition box (see e.g. [40]). The difference between process nodes and condition nodes is largely based on the modelling philosophy. From a technical point of view, there is no difference. Transfer boxes correspond to decision tables without time lags, and transition boxes correspond to decision tables with time lags. Causality edges connect process nodes, and condition edges connect condition nodes to process nodes via transfer or transition boxes.

The two most frequently cited DFM software tools are Dymonda [46] and Yadrat [41]. Dymonda has been developed by the original developers of DFM. It solves the graph model by transferring it to a set of timed fault trees representing different time steps, or alternatively combining the decision tables of the model into one critical transition table [29]. Dy-

monda solves an initial set of prime implicants directly from the timed fault trees or the critical transition table, and then applies the method of generalized consensus [47, 48] to solve the complete set of prime implicants. Yadrat has been developed by VTT. It transforms the DFM model into a BDD from which the prime implicants are solved. The prime implicant solving methods of the tools have not been compared properly with regard to computation times. A benefit of a BDD is that the non-coherent logic of the model is naturally present in the BDD structure. Because of this, the BDD approach requires less prime implicant processing after initial identification, whereas the Dymonda's approach of using the method of generalized consensus relies heavily on the comparisons between initial prime implicants. On the other hand, multi-state nodes have to be converted into binary variables when a BDD is used, and the prime implicants of the BDD have to be converted back to represent the multi-state logic.

Dymonda and Yadrat use slightly different specifications and terminology. Dymonda follows the official DFM specifications [49]. Yadrat can be considered as an alternative interpretation of the methodology. Despite their differences, the same deductive analyses can be performed using both tools. Yadrat does not provide support for inductive analysis.

In the computation of the top event probability in DFM, the basic idea is similar to the computation of the top event probability in fault tree analysis [32]. In DFM, the top event probability is calculated on the basis of the prime implicants and the probabilities of the literals. Determination of the probabilities of literals has not been addressed much in the literature. Probabilities have been presented mainly considering one time step, whereas time-dependent probability models have not been presented. However, DFM specifications [49] do mention that Dymonda contains time-dependent probability models. In this dissertation (see Paper III), an exponential model with a constant failure rate is used for the computation of failure probabilities. For computation of the top event probability, the usual upper bound algorithms [32] used in fault tree analysis can also be applied in DFM. More accurate top event probability algorithms have also been developed, such as the algorithm presented in [50] and the algorithm cited in [49].

The application areas of DFM have included digital control and safety systems in nuclear power plants [8, 45, 51], space systems [28, 52, 53], hydrogen production plants [40, 54], human performance [55], networked

control systems [9] and field programmable gate arrays (FPGAs) [44, 56]. The reliability analysis of digital systems is considered to be one of the greatest challenges in modern nuclear power plant PRA. Traditional static methods, such as fault trees, cannot capture the dynamic interactions of digital systems very well. NUREG/CR-6901 [57] has identified DFM as one of the promising methods for the reliability analysis of digital I&C systems. DFM has been considered effective in modelling dynamic interactions, such as delays, memories, logic loops and system states [51]. Interactions can, for example, lead to the coupling of events, such as opening of a valve and starting of a pump, and therefore, have a significant effect on the reliability of the system. Multi-state logic is advantageous, because the behaviour of software controlled systems is usually non-binary.

Most DFM models reported in the literature are rather small. To the author's knowledge, the largest model found in the literature represents the FPGA-based reactor trip logic loop in a detailed manner and contains 396 nodes [44]. The complexity of DFM analysis depends on the number of nodes and states of nodes, the complexity of decision tables, and the number of time steps used in the analysis. The computation times are rather sensitive to increase in any of these factors. The model in [44] was traced backwards only one time step, because the computation with multiple time steps would have lasted too long.

Aldemir et al. [8] compared DFM to Markov/CTCM methodology in modelling a digital feedwater control system. The results of the methodologies were consistent. An approach utilising both DFM and Markov analysis was proposed. The authors suggested that DFM could first be used to identify prime implicants. Thereafter, inductive Markov analysis could be performed to validate the prime implicants and to examine their sensitivity to variations of initial conditions.

2.5 Risk importance measures

Risk importance measures [23, 58] are used to analyse which components contribute most to a system's failure probability. This information helps to determine how the system's reliability can be improved effectively, e.g. where to add redundancy, which components to upgrade and how to allocate testing activities. The importance of a component depends on the reliability of the component itself, its position in the system's structure, and the need for the component in the system. The failure probability

of the component is a significant factor, but if the failure of component does not jeopardise the functioning of the system, it is not very important. At least two different importance measures should be used in the importance analysis, because one measure is usually limited to describing the component's influence over the system's reliability from one point of view only.

In the reliability analysis of nuclear power plants, the Fussell-Vesely measure of importance [59, 60] and the risk increase factor [61, 62] (also known as the risk achievement worth) are frequently used risk importance measures. Fussell-Vesely takes into account both the failure probability of the component and the system's capability to survive without the component. Therefore, Fussell-Vesely is typically used as the primary risk importance measure. The risk increase factor measures how much the failure of the component increases the probability of the system's failure. It is a good complement to Fussell-Vesely and it is useful, e.g. when the repairing order of failed components must be decided.

Although the previous paragraphs discussed component failures, risk importance measures can be calculated for any basic event. The Fussell-Vesely measure of importance $I^{FV}(i)$ for basic event i is defined as the probability that at least one minimal cut set containing basic event i has been realised assuming that the system has failed.

Definition 1 *Fussell-Vesely:*

$$I^{FV}(i) := \frac{Q^i_{TOP}}{Q_{TOP}}, \tag{2.2}$$

where Q_{TOP} is the probability that the system fails and Q^i_{TOP} is the probability that a minimal cut set including basic event i causes the system to fail.

The risk increase factor $I^I(i)$ for basic event i is defined as the system's failure probability with the condition that basic event i has occurred divided by the system's failure probability (without any conditions).

Definition 2 *The risk increase factor:*

$$I^I(i) := \frac{Q_{TOP}(i=1)}{Q_{TOP}}, \tag{2.3}$$

where $Q_{TOP}(i=1)$ is the failure probability of the system, given that basic event i has occurred.

2.6 Common cause failures

CCFs [24, 63] are an important part of the reliability analysis of complex systems including redundant components. If CCFs are not taken into account, the risk of the system's failure is likely to be underestimated. In [25], a CCF is defined using the following criteria:

"1. Two or more individual components fail, are degraded (including failures during demand or in-service testing), or have deficiencies that would result in component failures if a demand signal had been received.

2. Components fail within a selected period of time such that success of the probabilistic risk assessment (PRA) mission would be uncertain.

3. Components fail because of a single shared cause and coupling mechanism.

4. Components fail within the established component boundary."

There are different ways to model CCFs. In nuclear power plant PRA, parametric models [64] are used most often. In parametric models, the CCF probability is calculated by multiplying individual component failure probability with some CCF parameters. Another option is to estimate the CCF probability independently without considering the failure probabilities of individual components.

Two parametric models, β- and α-factor models, have been used in this dissertation. They are introduced in the following.

Consider a group of m identical components with a common failure mechanism. When CCFs are modelled using the β-factor model, it is assumed that a component can either fail independently or in a CCF of all m components. If a component fails, the failure is a CCF with probability β. Hence, if the component fails with probability Q, the probability of independent failure is $Q^1 = (1-\beta)\cdot Q$, and the probability of a CCF of all m components is $Q^m = \beta \cdot Q$.

The α-factor model considers the possibility that a subset of m components can fail due to a common cause, i.e. CCFs between different component combinations are possible. The formulas for the α-factor model are

$$Q^k = \frac{k}{\binom{m-1}{k-1}} \frac{\alpha_k}{\alpha_{tot}} Q, \tag{2.4}$$

$$\alpha_{tot} = \sum_{k=1}^{m} k\alpha_k, \tag{2.5}$$

where the factors α_1,..., α_m are determined by the analyst.

If an α-factor group includes four components, the failure probability of

a component is 0.001, $\alpha_1 = 0.935$, $\alpha_2 = 0.05$, $\alpha_3 = 0.01$ and $\alpha_4 = 0.005$, it follows that

$$\alpha_{tot} = 0.935 + 2 \cdot 0.05 + 3 \cdot 0.01 + 4 \cdot 0.005 = 1.085,$$

$$Q^1 = \frac{1}{\binom{3}{0}} \frac{0.935}{1.085} \cdot 0.001 \approx 8.62 \cdot 10^{-4},$$

$$Q^2 = \frac{2}{\binom{3}{1}} \frac{0.05}{1.085} \cdot 0.001 \approx 3.07 \cdot 10^{-5},$$

$$Q^3 = \frac{3}{\binom{3}{2}} \frac{0.01}{1.085} \cdot 0.001 \approx 9.22 \cdot 10^{-6}$$

and

$$Q^4 = \frac{4}{\binom{3}{3}} \frac{0.005}{1.085} \cdot 0.001 \approx 1.84 \cdot 10^{-5}.$$

3. Results

3.1 Prime implicants

Paper I presents a new definition of a prime implicant that is applicable in time-dependent dynamic reliability analysis. The basis for the definition is a reliability model that consists of a top function and a set of additional constraints. The analysis of non-repairable components in the DFM was the case that revealed the need for the new definition, because the results contained prime implicants that implied some other prime implicants. For example, according to the traditional definition [35],

$$\{V_1(-4), VF_0(-3), WL_0(-4), WLM_{-1}(-4), MF_1(-1)\}$$

and

$$\{V_1(-4), VF_0(-3), WL_0(-4), WLM_{-1}(-4), MF_1(-2)\}$$

are prime implicants of the example model presented in Paper I. However, since MF represents the failure of a non-repairable component, implicant

$$\{V_1(-4), VF_0(-3), WL_0(-4), WLM_{-1}(-4), MF_1(-2)\}$$

implies

$$\{V_1(-4), VF_0(-3), WL_0(-4), WLM_{-1}(-4), MF_1(-1)\}.$$

The new definition was developed with the idea that an implicant that implies some other length-minimal implicant is not a prime implicant, because it is not a minimal condition for causing the top event.

The new definition provides solid mathematical foundation for DFM. It takes time-related minimality into account. For example, assume that a top event occurs if a non-repairable component fails during a particular time frame. The component can fail at different time points to cause the

top event, but it does not need to fail until a specific time point. Therefore, the condition that the component is failed at the latest possible time point is minimal, and the condition that the component fails earlier is non-minimal. The new definition also supports the calculation of the top event probability better than the traditional definition [35], and conveniently a smaller number of prime implicants can represent the root causes of the top event. These claims are demonstrated by simple examples in Paper I. The definition and interpretation of prime implicants affect the modelling of the component's time-dependent behaviour and dependent events, and the computation of failure probabilities, the top event probability and risk importance measures. Hence, the new definition is an important basis for further research.

3.2 Risk importance measures

Paper II develops new dynamic risk importance measures as generalisations of traditional risk importance measures: the dynamic Fussell-Vesely (DFV) and the dynamic risk increase factor (DRIF). These risk importance measures map information from prime implicants to values that represent the significances of different events and conditions. The dynamic risk importance measures are calculated for the states of nodes. It is logical to separate different states of a node in the analysis because they represent completely different conditions. Furthermore, the information about time steps is taken into account in the computation of the dynamic risk importance measures.

Fussell-Vesely measures the portion of the top event probability coming from the minimal cut sets that include the analysed basic event. Correspondingly, the DFV measures the portion of the top event probability coming from the prime implicants that include a particular node in a particular state before or at a particular time step. If the analysed system is coherent with regard to the analysed state of the node, the DFV can be interpreted as the relative decrease in the top event probability caused by the condition that the node is not in the considered state until a particular time step. The DFV is presented in its basic form in Definition 3. In addition to the basic form, the DFV is formulated for failure states of components that are modelled with two nodes: one that determines whether the component has failed or not, and one that represents the functional state of a component. Another form of the DFV is also developed to mea-

sure the incoherency of a component.

Definition 3 *The dynamic Fussell-Vesely measure of state s of node i at time step $-t$ is*

$$I^{DFV}(i_s(-t)) := \frac{Q_{TOP}^{i_s(-t)}}{Q_{TOP}}, \tag{3.1}$$

where Q_{TOP} is the top event probability and $Q_{TOP}^{i_s(-t)}$ is the probability that a prime implicant including node i in state s before or at time step $-t$ causes the top event.

The DRIF measures how much the top event probability would relatively increase if the analysed node was in the considered state at all time steps of the DFM analysis time frame. The DRIF is presented in Definition 4. It can also be calculated for failure states of components.

Definition 4 *The dynamic risk increase factor of state s of node i is*

$$I^{DI}(i_s) := \frac{Q_{TOP}(i_s(-t) = 1,\ \forall\ t \in \{0, 1, ..., l-1, l\})}{Q_{TOP}}, \tag{3.2}$$

where $Q_{TOP}(i_s(-t) = 1,\ \forall\ t \in \{0, 1, ..., l-1, l\})$ is the probability that the top event occurs, assuming that node i is in state s at every time step starting from $-l$ which is the earliest possible time step for node i to be in state s considering the initial conditions. The last time step of the analysis is assumed to be 0 in this formula.

Paper II also presents how failure states of components can be tracked, because the information on failure states (as defined in Paper II) does not directly appear in prime implicants. The failure states provide useful information even without risk importance measures, because the failure state is an important factor when analysing the causes of a top event. Moreover, more information is obtained if it is known that a component fails to a particular state than if it is only known that the component fails somehow.

3.3 Common cause failures

One special characteristic of DFM is that components can fail at different time points but still contribute to the same top event. Even though a CCF event is often interpreted as a simultaneous failure of similar components, NUREG/CR-6268 [25] defines that components need to fail only during the PRA mission time, which is typically 24 hours. This definition is used

both in data collection and PRA analysis. In data collection, if multiple failures occur within 24 hours, they are interpreted as a CCF. In addition, 50% of such events where the time between failures is 24-48 hours are counted as CCFs, i.e. a timing factor of 0.5 is used [25]. For traditional fault tree analysis, it is irrelevant whether the components fail simultaneously or not during the mission time, but in DFM non-simultaneous CCFs can be considered, because DFM divides the mission time into smaller time intervals. Paper III takes the possibility of such CCFs into account.

Two parametric CCF models, β- and α-factor models, are used in CCF probability computation. The CCF probability is calculated on the basis of the average of the probabilities of individual component failures in (3.3) for the β-factor model.

$$P_\pi(C_1(-t_1, -t_2, ..., -t_m)) = \beta \cdot \frac{1}{m} \sum_{i=1}^{m} P_\pi(F_1^i(-t_i)), \tag{3.3}$$

where $-t_1, -t_2, ..., -t_m$ are the failure times of components, π is the prime implicant that contains the CCF, and $P_\pi(F_1^i(-t_i))$ is the probability that the i:th component is failed at time step $-t_i$ in the prime implicant π. The probabilities can depend on other literals included in the prime implicant as presented in Paper III.

The method is simple and, in most cases, conservative, because simultaneous CCFs are more likely. If non-simultaneous CCFs are ignored in the analysis, some CCF probabilities are underestimated assuming that non-simultaneous CCFs are possible, and some prime implicants are also left out. It is advantageous that the same β and α parameters can be used as in the traditional case so that ordinary CCF data [65] can be used in DFM analysis.

Paper III also presents how CCFs can be incorporated into DFM results. CCFs do not need to be accounted for when the prime implicants are first solved. All the prime implicants with CCFs can be created on the basis of the original prime implicants that contain individual failures. This approach was chosen so that the graph model would not become excessively complex, which would increase the computational demands significantly and make the analysis time-consuming.

4. Discussion and conclusions

This Dissertation has extended dynamic reliability analysis theory in the following ways. First, the mathematical foundation of DFM was strengthened by developing an improved definition of a prime implicant in Paper I (Section 3.1 in the Dissertation). Second, the DFM analysis was improved by defining and analysing new risk importance measures in Paper II (Section 3.2 in the Dissertation). Third, CCF modelling was developed in the DFM context in Paper III (Section 3.3 in the Dissertation).

Unlike any other importance measure, the dynamic risk importance measures utilise all the information available in prime implicants of DFM. They measure primarily the importances of different states of components and variables. The computation of the dynamic risk importance measures for failure states of components provides significant additional information compared to other importance values. On the basis of DFV results, it is possible to judge at which time points particular failures and conditions contribute to the top event.

The dynamic risk importance measures were developed for the needs of DFM, but their applicability to other dynamic risk analysis methodologies could also be studied. The DRIF could quite easily be applied in some different methodologies because it only measures the change in the top event probability caused by the analysed event, but the DFV relies heavily on prime implicants. In methods that do not solve prime implicants, some alternative way of calculating the DFV should be found or some other importance measure could be used instead. For example, dynamic simulation based methods are often applied in level 2 PRA [15, 66]. Timings of events are important in severe reactor accidents. Hence, the application of the dynamic risk importance measures could be studied in that area. Actually, some dynamic importance measures have already been developed for level 2 PRA [67].

Usually, the essential task in DFM analysis is to identify the prime implicants of the top event. Prime implicant identification should be studied considering the definition presented in Paper I. Previously, e.g. non-repairable components have been handled using case-specific treatments in the identification process. A goal for future research could be to develop a prime implicant identification algorithm that would take additional constraints of any type into account. The decomposition theorem from [68] could possibly be generalised to take the multi-state logic and additional constraints into account.

Computational efficiency is a key issue in the development of DFM for practical reliability analysis. There is a need both to identify prime implicants and to perform quantification in a reasonable time. If a new prime implicant identification algorithm is developed as discussed in the previous paragraph, the implementation of the algorithm should also be efficient enough to be used in practice. Correspondingly, the computation of the top event probability and the risk importance measures should be reasonably fast, but also accurate enough. Approximations can be calculated rapidly, but the computation of accurate values can be demanding if the analysed model is not small. Approximate values are usually sufficient in most reliability analyses as long as they are accurate enough. The quantification of DFM could be studied in greater depth so that an optimal balance between accuracy and computation times could be achieved in the computation of the top event probability and the risk importance measures.

Despite its importance, quantification of individual literals and prime implicants has not been much addressed in the literature. Paper III presented how to calculate the failure probabilities of non-repairable components and CCF probabilities, but different component reliability models, the determination of the probabilities of the initial states of variables, and quantification of cascading failures and other dependent events could also be studied. One factor that adds complexity is that the same literal can have different probabilities in different prime implicants if its probability depends on other literals. Non-repairable components are a simple example of this, as presented in Paper III. Probabilistic analysis depends on the modelling decisions, and the modelling of components and variables should therefore also be considered from the quantification point of view. An interesting and challenging topic for future research is to define good practices in the DFM modelling and quantification.

The CCF probability computation could be made more accurate by developing time-dependent CCF models. The assumption of simultaneous failures is not realistic in many cases. Timing related CCF parameters could be estimated, for example the probability that the difference between failure times lies within a specific interval. Data about failure times is actually already collected [25], but is only utilised in the classification of events. Data analyses would be needed to study to which failure modes time-dependent models should be applied.

DFM has been considered to be too complex to be applied to large systems, and most applications found in the literature are rather small. However, more efficient DFM tools and prime implicant solving technologies, such as BDDs, are being developed, and computers are becoming more and more powerful. Recent DFM models have been larger [28, 44], and this development will probably continue in the future. With larger models, the ability to analyse results efficiently becomes even more important. Therefore, suitable risk importance measures are needed. CCFs are also more important when larger systems with redundancies are analysed.

Bibliography

[1] Häring, I. Risk analysis and management: Engineering resilience. Singapore: Springer, 2015. 397 p. ISBN: 978-981-10-0015-7.

[2] Bedford, T. & Cooke, R. Probabilistic risk analysis: Foundation and methods. Cambridge: Cambridge University Press, 2001. 393 p. ISBN: 0521773202, 9780521773201.

[3] Emblemsvåg, J. & Kjølstad, L.E. Qualitative risk analysis: Some problems and remedies. Management Decision, Vol. 44 (2006) 3, pp. 395-408.

[4] Vesely, W.E., Goldberg, F.F., Roberts, N.H. & Haasl, D.F. Fault tree handbook. Washingon D.C.: U.S. Nuclear Regulatory Commission, 1981. 202 p. NUREG-0492.

[5] Clements, P.L. Fault tree analysis, 4th edition. Massachusetts: Sverdrup Technology, Inc., 1993. 96 p.

[6] Labeau, P.E., Smidts, C. & Swaminathan, S. Dynamic reliability: towards an integrated platform for probabilistic risk assessment. Reliability Engineering and System Safety, Vol. 68 (2000) 3, pp. 219-254.

[7] Garrett, C.J., Guarro, S.B. & Apostolakis, G.E. The dynamic flowgraph methodology for assessing the dependability of embedded software systems. Systems, Man and Cybernetics, Vol. 25 (1995) 5, pp. 824-840.

[8] Aldemir, T., Guarro, S., Mandelli, D., Kirschenbaum, J., Mangan, L.A., Bucci, P., Yau, M., Ekici, E., Miller, D.W., Sun, X. & Arndt, S.A. Probabilistic risk assessment modeling of digital instrumentation and control systems using two dynamic methodologies. Reliability Engineering and System Safety, Vol. 95 (2010), pp. 1011-1039.

[9] Al-Dabbagh, A.W. & Lu, L. Reliability modeling of networked control systems using dynamic flowgraph methodology. Reliability Engineering and System Safety, Vol. 95 (2010), pp. 1202-1209.

[10] 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, Vol. 93 (2008), pp. 1616-1627.

[11] Gomes, I.B., Saldanha, P.L.C. & Frutuoso e Melo, P.F.F. A cell-to-cell Markovian model for the reliability of a digital control system of a steam generator. Proceedings of the 2013 International Nuclear Atlantic Conference - INAC 2013, Recife, Brazil, 24-29 November 2013. ISBN: 978-85-99141-05-2.

[12] Yang, J. & Aldemir, T. An algorithm for the computationally efficient deductive implementation of the Markov/cell-to-cell-mapping technique for risk significant scenario identification. Reliability Engineering and System Safety, Vol. 145 (2016), pp. 1-8.

[13] Acosta, C.G. & Siu, N.O. Dynamic event tree analysis method (DETAM) for accident sequence analysis. Cambridge (USA): Massachusetts Institute of Technology Nuclear Engineering Department, 1991. 138+23 p. MITNE-295.

[14] Karanki, D.R., Kim, T.-W. & Dang, V.N. A dynamic event tree informed approach to probabilistic accident sequence modeling: Dynamics and variabilities in medium LOCA. Reliability Engineering and System Safety, Vol. 142 (2015), pp. 78-91.

[15] Tyrväinen, T., Silvonen, T. & Mätäsniemi, T. Computing source terms with dynamic containment event trees. Proceedings of the 13th International Probabilistic Safety Assessment and Management Conference; 2016 Oct 2-7; Seoul, Korea.

[16] Sadou, N. & Demmou, H. Reliability analysis of discrete event dynamic systems with Petri nets. Reliability Engineering and System Safety, Vol. 94 (2009) 11, pp. 1848-1861.

[17] Swaminathan, S. & Smidts, C. The mathematical formulation for the event sequence diagram framework. Reliability Engineering and System Safety, Vol. 65 (1999) 2, pp. 103-118.

[18] Zhao, J., Liu, T. Zhao, Y., Liu, D., Yang, X., Lin, J., Lin, Z. & Lei, Y. Reliability evaluation of NPP's power supply system based on improved GO-FLOW method. Science and Technology of Nuclear Installations, Vol. 2016 (2016), 10 p.

[19] Cepin, M. & Mavko, B. A dynamic fault tree. Reliability Engineering and System Safety, Vol. 75 (2002) 1, pp. 83-91.

[20] Huang, H.W., Shih, C., Yih, S. & Chen, M.H. Integrated software safety analysis method for digital I&C systems. Annals of Nuclear Energy, Vol. 35 (2008) 8, pp. 1471-1483.

[21] Wang, W., Di Maio, F. & Zio, E. Component- and system-level degradation modeling of digital instrumentation and control systems based on a multi-state physics modeling approach. Annals of Nuclear Energy, Vol. 95 (2016), pp. 135-147.

[22] Zio, E. & Di Maio, F. Processing dynamic scenarios from a reliability analysis of a nuclear power plant digital instrumentation and control system. Annals of Nuclear Energy, Vol. 36 (2009) 9, pp. 1386-1399.

[23] Van Der Borst, M. & Schoonakker, H. An overview of PSA importance measures. Reliability Engineering and System Safety, Vol. 72 (2001) 3, pp. 241-245.

[24] Mosleh, A., Fleming, K.N., Parry, G.W., Paula, H.M., Worledge, D.H. & Rasmuson, D.M. Procedures for treating common cause failures in safety and reliability studies: Procedular framework and examples. Washingon D.C.: U.S. Nuclear Regulatory Commission, Division of Reactor and Plant Systems, 1988. 202 p. NUREG/CR-4780 EPRI NP-5613 Vol. 1.

[25] Wierman, T.E., Rasmuson, D.M. & Mosleh, A. Common-cause failure database and analysis system: Event data collection, classification, and coding. Washingon D.C.: U.S. Nuclear Regulatory Commission, Division of Risk Assessment and Special Projects, 2007. NUREG/CR-6268, Rev. 1 INL/EXT-07-12969.

[26] Karanta, I. Importance measures for the dynamic flowgraph methodology. Espoo (Finland): VTT Technical Research Centre of Finland, Systems Research, 2011. VTT-R-00525-11.

[27] Houtermans, M.J.M. A method for dynamic process hazard analysis and integrated process safety management [doctoral thesis].

Eindhoven (Netherlands): Technische Universiteit Eindhoven, 2001. http://alexandria.tue.nl/extra2/200111699.pdf.

[28] Yau, M., Dixon, S. & Guarro, S. Application of the dynamic flowgraph methodology to the space propulsion system benchmark problem. Proceedings of the 12th International Probabilistic Safety Assessment and Management Conference; 2014 Jun 22-27; Sheraton Waikiki, Honolulu, Hawaii, USA.

[29] Yau, M., Apostolakis, G. & Guarro, S. The use of prime implicants in dependability analysis of software controlled systems. Reliability Engineering and Systems Safety, Vol. 62 (1998), pp. 23-32.

[30] Tyrväinen, T. Risk importance measures and common cause failures in dynamic flowgraph methodology [master's thesis]. Espoo (Finland): Aalto University, School of Science, 2011.

[31] Halmos, P. & Givant, S. Introduction to Boolean algebras. New York: Springer, Undergraduate Texts in Mathematics, 2009. 446 p. ISBN: 978-0-387-40293-2.

[32] Jung, W.S. A method to improve cutset probability calculation in probabilistic safety assessment of nuclear power plants. Reliability Engineering and System Safety, Vol. 134 (2015), pp. 134-142.

[33] Bryant, R.E. Graph-based algorithms for Boolean function manipulation. Computers, Vol. C-35 (1986) 8, pp. 677-691.

[34] Bryant, R.E. Symbolic Boolean manipulation with ordered binary-decision diagrams. ACM Computing Surveys, Vol. 24 (1992) 3, pp. 293-318.

[35] Rauzy, A. Mathematical foundation of minimal cutsets. IEEE transactions on Reliability, Vol. 50 (2001) 4, pp. 389-396.

[36] Contini, S., Cojazzi, G.G.M. & Renda, G. On the use of non-coherent fault trees in safety and security studies. Reliability Engineering and System Safety, Vol. 93 (2008) 12, pp. 1886-1895.

[37] Rauzy, A. Binary decision diagrams for reliability studies. In: Misra, K.B. Handbook of performance engineering. London: Springer London, 2008. pp. 381-396.

[38] Ching, W.-K., Huang, X., Ng, M.K. & Siu, T.-K. Markov chains: Models, algorithms and applications. New York: Springer US, 2013. 258 p. ISBN: 978-1-4614-6312-2.

[39] Hejase, M., Kurt, A., Aldemir, T., Ozguner, U., Guarro, S.B., Yau, M.K. & Knudson, M.D. A quantitative and risk based framework for UAS control system assurance. AIAA Information Systems-AIAA Infotech @ Aerospace; 2017 Jan 9-13; Grapevine, Texas, USA. AIAA 2017-0882.

[40] Al-Dabbagh, A.W. & Lu, L. Dynamic flowgraph modeling of process and control systems of a nuclear-based hydrogen production plant. International Journal of Hydrogen Energy, Vol. 35 (2010), pp. 9569-9580.

[41] Björkman, K. Solving dynamic flowgraph methodology models using binary decision diagrams. Reliability Engineering and System Safety, Vol. 111 (2013), pp. 206-216.

[42] Guarro, S., Yau, M. & Dixon, S. Applications of the dynamic flowgraph methodology to dynamic modeling and analysis. Proceedings of the 11th International Probabilistic Safety Assessment and Management Conference & The Annual European Safety and Reliability Conference; 2012 Jun 25-29; Helsinki, Finland.

[43] Guarro, S., Yau, M. & Dixon, S. Advanced risk modeling and risk-informed testing of digital instrumentation and control systems. Proceedings of the Probabilistic Safety Assessment Conference (PSA-11); 2011 Mar 13-17; Wilmington, NC.

[44] McNelles P., Zeng, Z.C., Renganathan, G., Lamarre, G., Akl, Y. & Lu, L. A comparison of fault trees and the dynamic flowgraph methodology for the analysis of FPGA-based safety systems part 1: Reactor trip logic loop reliability analysis. Reliability Engineering and System Safety, Vol. 153 (2016), pp. 135-150.

[45] Houtermans, M., Apostolakis, G., Brombacher, A. & Karydas, D. The dynamic flowgraph methodology as a safety analysis tool: Programmable electronic system design and verification. Safety Science, Vol. 40 (2002), pp. 813-833.

[46] ASCA Inc. Dymonda. 2010. www.ascainc.com/dymonda/dymonda.html [Referred 17.8.2017].

[47] Quine, W.V. A way to simplify truth functions. American Mathematical Monthly, Vol. 62 (1955), pp. 627-631.

[48] Cepek, O., Kucera, P. & Kurik, S. Boolean functions with long prime implicants. Information Processing Letters, Vol. 113 (2013), 698-703.

[49] ASCA Inc. DFM specifications. 2010. www.ascainc.com/dfm/dfm_specs.html [Referred 17.8.2017].

[50] Karanta, I. Implementing dynamic flowgraph methodology models with logic programs. Journal of Risk and Reliability, Vol. 227 (2013), pp. 302-314.

[51] Pinto, J.M.O., Frutuoso e Melo, P.F. & Saldanha, P.L.C. A dynamic failure evaluation of a simplified digital control system of a nuclear power plant pressurizer. Proceedings of the 13th Brazilian Congress of Thermal Sciences and Engineering; 2010 Dec 5-10; Uberlandia, MG, Brazil. Rio de Janeiro: ABCM; 2010.

[52] Yau, M., Guarro, S. & Apostolakis, G. Demonstration of the dynamic flowgraph methodology using the Titan II space launch vehicle digital flight control system. Reliability Engineering and System Safety, Vol. 49 (1995), pp. 335-353.

[53] Shi, J., Wang, G. & Tong, T. The integrated health monitoring design using the dynamic flowgraph methodology for thermal control systems of payloads. Chemical Engineering Transactions, Vol. 33 (2013), pp. 211-216.

[54] Ahmed, F. Probabilistic risk assessment using dynamic flowgraph methodology for copper chloride CANDU-SCWR hydrogen production. Procedia Computer Science, Vol. 19 (2013), pp. 777-785.

[55] Milici, A., Mulvihill, R. & Guarro, S. Extending the dynamic flowgraph methodology (DFM) to model human performance and team effects. Washingon D.C.: U.S. Nuclear Regulatory Commission, Division of System Analysis and Regulatory Effectiveness, 2001. NUREG/CR-6710.

[56] McNelles, P. & Lu, L. Field programmable gate array reliability analysis using the dynamic flowgraph methodology. Nuclear Engineering and Technology, Vol. 48 (2016) 5, pp. 1192-1205.

[57] Aldemir, T., Miller, D.W., Stovsky, M.P., Kirschenbaum, J., Bucci, P., Fentiman, A.W. & Mangan L.T. Current state of reliability modeling methodologies for digital systems and their acceptance criteria for nuclear power plant assessments. Washingon D.C.: U.S. Nuclear Regulatory Commission, Division of Fuel, Engineering, and Radiological Research, 2006. NUREG/CR-6901.

[58] Zio, E. Risk importance measures. In: Pham, H. Safety and risk modeling and its applications. London: Springer-Verlag, 2011. pp. 151-195.

[59] Meng, F.C. Relationships of Fussell-Vesely and Birnbaum importance to structural importance in coherent systems. Reliability Engineering and System Safety, Vol. 67 (2000) 1, pp. 55-60.

[60] Laitonen, J. & Niemelä, I. Analyzing system changes with importance measure pairs: Risk increase factor and Fussell-Vesely compared to Birnbaum and failure probability. Proceedings of the 12th International Probabilistic Safety Assessment and Management Conference; 2014 Jun 22-27; Sheraton Waikiki, Honolulu, Hawaii, USA.

[61] Bäckström, O., Krcal, P. & Wang, W. Two interpretations of the risk increase factor definition. In: Walls, S., Revie, M. & Bedford, T. Risk, reliability and safety: Innovating theory and practice. London: Taylor & Francis Group, 2017. pp. 2816-2822. ISBN: 978-1-138-02997-2.

[62] Martorell, S., Marton, I., Martorell, P., Carlos, S. & Sanchez, A.I. RAM based metrics for safety assessment of safety systems with application to ageing management. In: Podofilini, L., Sudret, B., Stojadinovic, B., Zio, E. & Kröger, W. Safety and reliability of complex engineered systems. London: Taylor & Francis Group, 2015. pp. 1645-1650. ISBN: 978-1-138-02879-1.

[63] Høyland, A. & Rausand, M. Dependent failures. In: System reliability theory: Models and statistical methods. New York: Wiley Series in Probability and mathematical statistics: Applied probability and statistics section, 1994. pp. 325-354. ISBN: 0-471-59397-4.

[64] Chebila, M. & Innal, F. Unification of common cause failures' parametric models using a generic Markovian model. Journal of Failure Analysis and Prevention, Vol. 14 (2014), pp. 426-434.

[65] Mosleh, A., Rasmuson, D.M. & Marshall, F.M. Guidelines on modelling common-cause failures in probabilistic risk assessment. Washington D.C.: U.S. Nuclear Regulatory Commission, Safety Programs Division, 1998. NUREG/CR-5485, INEEL/EXT-97-01327.

[66] Guigueno, Y., Raimond, E., Duflot, N., Tanchoux, V., Rahni, N., Laurent, B. & Kioseyian, G. Severe accident risk assessment for NPPs - Software tools and methodologies for level 2 PSA development available at IRSN. Proceedings of the 13th International Probabilistic Safety Assessment and Management Conference; 2016 Oct 2-7; Seoul, Korea.

[67] Jankovsky, Z.K., Denman, M.R. & Aldemir, T. Dynamic importance measures in the ADAPT framework. Transactions of the American Nuclear Society, Vol. 115 (2016), pp. 799-802.

[68] Rauzy A & Dutuit Y. Exact and truncated computation of prime implicants of coherent and non-coherent fault trees within Aralia. Reliability Engineering and System Safety, Vol. 58 (1997), pp. 127-144.

Publication I

Tyrväinen, T. Prime implicants in dynamic reliability analysis. *Reliability Engineering and System Safety*, Vol. 146, pp. 39-46, doi: 10.1016/j.ress.2015.10.007, February 2016.

Reliability Engineering and System Safety 146 (2016) 39–46

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

Prime implicants in dynamic reliability analysis

Tero Tyrväinen

VTT Technical Research Centre of Finland, Lifetime Management, P.O. Box 1000, FI-02044 Espoo, Finland

ARTICLE INFO

Article history:
Received 30 May 2014
Received in revised form
3 July 2015
Accepted 5 October 2015
Available online 22 October 2015

Keywords:
Prime implicant
Dynamic reliability analysis
Dynamic flowgraph methodology

ABSTRACT

This paper develops an improved definition of a prime implicant for the needs of dynamic reliability analysis. Reliability analyses often aim to identify minimal cut sets or prime implicants, which are minimal conditions that cause an undesired top event, such as a system's failure. Dynamic reliability analysis methods take the time-dependent behaviour of a system into account. This means that the state of a component can change in the analysed time frame and prime implicants can include the failure of a component at different time points. There can also be dynamic constraints on a component's behaviour. For example, a component can be non-repairable in the given time frame. If a non-repairable component needs to be failed at a certain time point to cause the top event, we consider that the condition that it is failed at the latest possible time point is minimal, and the condition in which it fails earlier non-minimal. The traditional definition of a prime implicant does not account for this type of time-related minimality. In this paper, a new definition is introduced and illustrated using a dynamic flowgraph methodology model.

1. Introduction

1.1. Boolean algebra in reliability analysis

Reliability analyses are often used for identifying the possible root causes of an undesired top event, such as a system's failure [1]. These root causes can be combinations of basic events such as component failures, harmful environmental conditions and human errors. A minimal combination of basic events that is sufficient to cause the top event is called a minimal cut set [2]. Here, the minimality means that if one of the basic events is removed from a minimal cut set, the remaining combination of basic events is no longer sufficient to cause the top event. Minimal cut sets are usually the basic result of a reliability analysis. They are often used as the basis for probabilistic calculations, such as the computation of total probability [1,3] and risk importance measures [1,3,4], uncertainty analysis [3] and sensitivity analysis [3].

The theory of minimal cut sets and prime implicants is based on Boolean algebra. Boolean algebra defines algebraic operations for variables that can have two values: 0 ('false') and 1 ('true'). Boolean variables form Boolean formulas when they are connected using logical connectives, such as $+$ ('OR') and $\cdot$ ('AND'). For example, $F_T = a \cdot b \cdot c + a \cdot d$ is a Boolean formula if a, b, c and d are Boolean variables. A Boolean product is a set of Boolean variables connected by $\cdot$. For instance, $a \cdot b \cdot c$ and $a \cdot d$ are products. The expression can be shortened: $F_T = abc + ad$. The axioms of Boolean algebra are presented in Appendix A.

Let G and H be Boolean formulas. Formula G implies H, if from $G=1$, it follows that $H=1$. Formula F_T has value 1 if and only if a, b and c have value 1, or if a and d have value 1. Hence, products abc and ad imply F_T.

In reliability analysis, a top event can be represented by a Boolean formula of variables that represent basic events and minimal cut sets can be represented by Boolean products that imply the Boolean formula representing the top event. In traditional reliability analysis, basic events are assumed to be independent. In this paper, basic events are assumed to be independent unless dependencies between them are presented. In what follows, the Boolean formula that represents the top event is called a top function. For example, if $F_T = abc + ad$ is a top function and a, b, c and d represent basic events, the top event has two minimal cut sets: abc and ad.

The definition of a minimal cut set is adequate only for coherent reliability models. A reliability model is coherent only if the top function is monotonically increasing with regard to its arguments and all basic events are relevant. In an incoherent reliability model [5], however, failure of a component may actually prevent the top event from occurring, and the act of repairing it could cause the top event. For incoherent reliability models, the concept of a *prime implicant* is used instead of a minimal cut set to represent a minimal combination of conditions that causes the top event [6–8].

E-mail address: tero.tyrvainen@vtt.fi

http://dx.doi.org/10.1016/j.ress.2015.10.007
0951-8320/

A literal is either a Boolean variable a or its negation $\overline{a}$, also called a negative literal. For opposite literals a and $\overline{a}$, it holds that $a \cdot \overline{a} = 0$ and $a + \overline{a} = 1$. It also holds that

$$\overline{a+b} = \overline{a} \cdot \overline{b} \tag{1}$$

and

$$\overline{a \cdot b} = \overline{a} + \overline{b}. \tag{2}$$

A so-called negative basic event is a complement of a regular basic event (e.g. component not failed). In incoherent reliability analysis, negative basic events represented by negative literals can appear in the top function and prime implicants. The definition of a prime implicant is presented in Definition 1 [6].

Definition 1. Let F_T be a top function and π be a product. The product π is an implicant of F_T if π implies F_T.

An implicant π is a prime implicant, if there is no other implicant ρ of F_T such that $\rho \subset \pi$.

To illustrate Definition 1, prime implicants of formula $G = ab + c\overline{d}$ are ab and $c\overline{d}$. For formula $F_T = \overline{a}bc + b\overline{c}d + c\overline{d}f + ce\overline{f}$, the identification of prime implicants is more challenging. It is easy to see that $\overline{a}bc$, $b\overline{c}d$, $c\overline{d}f$ and $ce\overline{f}$ are prime implicants, but $\overline{a}bd$ is also a prime implicant, because, if $c=1$ and $\overline{a}bd = 1$, then $\overline{a}bc = 1$, and if $c=0$ and $\overline{a}bd = 1$, then $b\overline{c}d = 1$. Also, $c\overline{d}e$ is a prime implicant, because if $f=1$ and $c\overline{d}e = 1$, then $c\overline{d}f = 1$, and if $f=0$ and $c\overline{d}e = 1$, then $ce\overline{f} = 1$.

1.2. Dynamic reliability analysis

In dynamic reliability analysis [9,10], there can be causal dependencies between events represented by literals [11]. For example, the failure of a non-repairable component at time point t_1 implies that the component continues to be failed at later time point t_2. If literal f_{t_i} indicates that the component is failed at time step t_i, then f_{t_1} implies f_{t_2}. Fig. 1 presents a fault tree whose prime implicants are af_{t_1}, bf_{t_1} and bf_{t_2} according to Definition 1. However, when analysing implicants bf_{t_1} and bf_{t_2}, it should be noticed that, if $b=1$, then the failure condition represented by f has to start only at time step t_2 to cause the top event. The failure can occur already at time step t_1, but it does not need to. Literal f_{t_1} represents a more restrictive condition than f_{t_2} and on the other hand, if bf_{t_2} implies the top event, then bf_{t_1} also implies the top event.

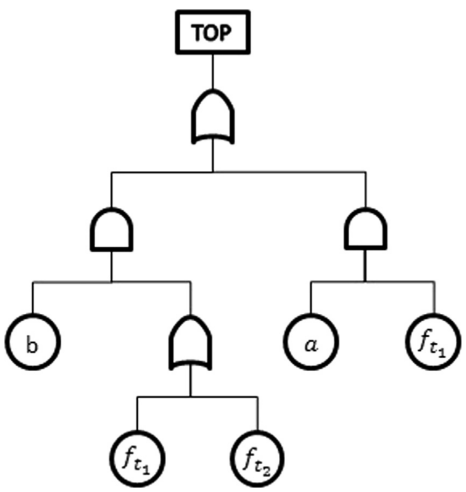

Fig. 1. A fault tree with dependent basic events.

Therefore, bf_{t_1} is not a minimal condition for the top event to occur and is not a prime implicant.

The conclusion of the previous example is that Definition 1 is not adequate when the reliability model contains dynamic dependencies between its variables. A new definition for prime implicants is introduced in Section 2. Section 3 shows how the definition is applicable to multi-state reliability analysis. A dynamic reliability analysis method called dynamic flowgraph methodology (DFM) [10,12–14] is used as an example of a methodology where the new definition is useful. DFM is presented in Section 4. Prime implicants of an example DFM model are identified in Section 5. It is shown that the new definition is logical, supports the computation of the top event probability better and allows the root causes of the top event to be represented by a smaller number of prime implicants. As the prime implicants are the basic result of DFM analysis, their definition and interpretation also affects other areas of the analysis, such as probabilistic reliability models, the computation of risk importance measures [15] and the modelling of common cause failures [16].

2. Definition of a prime implicant

The basis for the development of the new definition is a reliability model that can be represented as a top function and additional constraints. These additional constraints can, in principle, be any Boolean equations between the literals of the model.

The main motivation for the new definition is that it is needed in dynamic flowgraph methodology. A DFM model includes a graph model and constraints for the behaviour of the graph's nodes. This model is converted to a Boolean top function to solve prime implicants. The prime implicants that are solved from the top function need to correspond to the graph model. Definition 1 can easily be applied to literals of DFM, but it does not account such minimality as described in the example of Fig. 1. Minimality that is related to a physical constraint, such as non-repairability, has to be taken into account, and therefore, it is practical to include additional constraints to the reliability model along with the top function.

A more simple approach to account constraints would be to build them directly into the top function so that “the reliability model” would contain only the top function. In that case, the traditional definition could be used as it is, and minimality related to non-repairability of a component could be taken into account, in theory at least. However, prime implicants are a property of a DFM model and they can be identified directly from the graph in simple cases. It has to be possible to apply the prime implicant definition in DFM. Even if correct prime implicants were solved by taking non-repairability constraints into account in the conversion of the DFM model to top function, Definition 1 would not be adequate when identifying prime implicants directly from the DFM model.

Before introducing the new definition, the concept of a minterm needs to be defined. If V is the set of all the Boolean variables in the model, then a minterm is a product consisting of each variable in V or its negation. For example, $ab\overline{c}d$ and $a\overline{b}\overline{c}d$ are minterms of example $G = ab + c\overline{d}$ among 14 others.

A new definition of an implicant is presented in Definition 2. Compared to the traditional definition (in Definition 1), the new definition adds a condition that additional constraints cannot be violated (e.g. an implicant cannot include a non-repairable component first failed and then repaired).

Definition 2. Let F_T be the top function representing the top event, π be a product, $\mathbf{A}$ be a vector of Boolean formulas and $\mathbf{A} = \mathbf{1}$ be a set of additional constraints. The product π is an implicant of the

top event if π implies F_T and there exists minterm ν such that ν implies π and each formula in **A**.

The part "there exists minterm ν such that ν implies π and each formula in **A**" means that the implicant does not contradict any additional constraint. In other words, the additional constraints can hold if the conditions belonging to the implicant are realised. For example, consider a model with top function $F_T = \overline{a}bc + b\overline{c}d + c\overline{d}f + ce\overline{f}$ and additional constraints $e+\overline{g}=1$ (g implies e) and $f+\overline{e}=1$ (e implies f). Product $ce\overline{f}$ does imply the top function F_T as stated in the introduction, but there is no minterm that implies $ce\overline{f}$, $e+\overline{g}$ and $f+\overline{e}$. This is because $f+\overline{e}=0$ if $ce\overline{f}=1$. Implicants of the top function include $\overline{a}bc$, $\overline{a}bce$, $b\overline{c}d$, $b\overline{c}df$, $\overline{a}bd$, $\overline{a}bde$, $c\overline{d}e$, $c\overline{d}ef$ and $c\overline{d}f$ among others. It should also be noticed that a literal can appear in an implicant even if it does not appear in the top function. Product $c\overline{d}g$ is an implicant, because if $c\overline{d}g=1$, then $c\overline{d}e=1$ due to additional constraint $e+\overline{g}=1$ and hence $F_T=1$.

In Definition 2, an additional constraint is introduced as a Boolean equation of form $H=1$, where H is a formula. However, additional constraints can be presented in other forms as well. Each Boolean equation can be presented in form $H^*=0$ or in form $H=1$. For example,

$$e+\overline{g}=1 \Leftrightarrow \overline{e}g=0 \Leftrightarrow e+g=e. \tag{3}$$

Now, 'L-minimal implicants' (length-minimal implicants) are defined in the same manner as prime implicants in Definition 1. The difference to Definition 1 is that in this definition, an implicant is a product that satisfies Definition 2.

Definition 3. An implicant π is an L-minimal implicant if there is no other implicant ρ of F_T such that $\rho \subset \pi$.

To continue the previous example, L-minimal implicants of the top function are $\overline{a}bc$, $b\overline{c}d$, $\overline{a}bd$, $c\overline{d}e$, $c\overline{d}g$ and $c\overline{d}f$.

Following the idea that all length-minimal implicants are not prime implicants as discussed in the Introduction, prime implicants are defined in Definition 4. In this definition, a prime implicant is an implicant that does not imply any other L-minimal implicant. For comparison, in [17], a prime implicant was defined as an implicant that does not imply any other implicant. This is an essential property of prime implicants. An implicant that implies another prime implicant is not minimal because the prime implicant it implies is.

Definition 4. An L-minimal implicant π is a prime implicant if there is no other L-minimal implicant ρ of F_T such that $\rho+\pi=\rho$ if additional constraints $\mathbf{A}=\mathbf{1}$ hold.

Note that the equation $\rho+\pi=\rho$ can be satisfied due to additional constraints even if it does not hold that $\rho \subset \pi$. Without additional constraints, $\rho+\pi=\rho$ is equivalent to $\rho \subset \pi$, and Definition 1 is equivalent to Definition 4.

In the previous example, products $\overline{a}bc$, $b\overline{c}d$, $\overline{a}bd$, $c\overline{d}e$, $c\overline{d}g$ and $c\overline{d}f$ were identified to be L-minimal implicants. Now, according to Definition 4, $c\overline{d}g$ is not a prime implicant because $e+g=e$ (see (3)), and hence, $c\overline{d}e+c\overline{d}g=c\overline{d}\cdot(e+g)=c\overline{d}e$. L-minimal implicant $c\overline{d}e$ is not a prime implicant because $e+f=e\cdot(f+\overline{e})+f=ef+e\overline{e}+f=ef+f=f$, and hence, $c\overline{d}e+c\overline{d}f=c\overline{d}\cdot(e+f)=c\overline{d}f$. Hence, the prime implicants are $\overline{a}bc$, $b\overline{c}d$, $\overline{a}bd$ and $c\overline{d}f$.

It should be noted that only L-minimal implicants must be considered in Definition 4 and not all implicants because it is possible that a prime implicant implies an implicant that is not L-minimal. For instance, in the example of Fig. 1, due to additional constraint $f_{t_1}+f_{t_2}=f_{t_2}$ (f_{t_1} implies f_{t_2}), it holds that

$$\begin{aligned} af_{t_1} &= af_{t_1}f_{t_2}+af_{t_1}=af_{t_1}f_{t_2}+af_{t_1}(f_{t_2}+\overline{f}_{t_2})=af_{t_1}f_{t_2}+af_{t_1}\overline{f}_{t_2} \\ &= af_{t_1}f_{t_2}+af_{t_1}(\overline{f}_{t_1}+\overline{f}_{t_2})=af_{t_1}f_{t_2}+af_{t_1}\overline{f}_{t_1}\overline{f}_{t_2}=af_{t_1}f_{t_2}+0=af_{t_1}f_{t_2}. \end{aligned} \tag{4}$$

3. Generalisation to multi-state reliability analysis

All previous examples of this paper were examples of binary reliability analysis in which components can have two states represented by a positive literal a and a negative literal $\overline{a}$. In reality, many complex systems are not binary, and it is restrictive to model components only with a failed state and a functioning state. Multi-state reliability analysis [18,19] allows components to have more than two states. To make the prime implicant definition applicable to multi-state reliability analysis, the concept of a literal has to be generalised to the multi-state case.

Let a literal be denoted as C_s, where C is a component and s is a state. As in the binary case, literals of this type can have value 0 or 1. For literals of a component, it holds that

$$\sum_{s\in S} C_s = 1 \tag{5}$$

and

$$C_{s_1}\cdot C_{s_2}=0, \tag{6}$$

where S is the set of all states of component C and $s_1 \neq s_2$. The number of states is assumed to be finite in this paper, and S a finite set.

Definitions 2–4 can now be applied to the multi-state case. Let $F_T = A_0B_1C_0 + B_1C_1D_1 + B_1C_2E_2 + C_1D_0E_1 + C_1D_0F_1$ be a top function and $E_1+F_1=F_1$ be an additional constraint. If component C has only states 0, 1 and 2, prime implicants are $A_0B_1C_0$, $B_1C_1D_1$, $B_1C_2E_2$, $A_0B_1D_1E_2$, $C_1D_0F_1$ and $A_0B_1D_0E_2F_1$. Product $A_0B_1D_1E_2$ is a prime implicant because constraint (5) holds for component C and either product $A_0B_1C_0$, $B_1C_1D_1$ or $B_1C_2E_2$ is therefore 1 if $A_0B_1D_1E_2=1$. Similarly, product $A_0B_1D_0E_2F_1$ is a prime implicant because either product $A_0B_1C_0$, $C_1D_0F_1$ or $B_1C_2E_2$ is 1 if $A_0B_1D_0E_2F_1=1$.

4. Dynamic flowgraph methodology

4.1. Introduction to the methodology

Dynamic flowgraph methodology (DFM) [10,12–14] is an approach for analysing systems with time dependencies and feedback loops. The reason for the development of DFM is that traditional methods, such as fault tree analysis, can describe the system's dynamic behaviour only in a limited manner. DFM can more accurately represent system's evolution in time. DFM is typically used to model and analyse digitally controlled systems that include both hardware and software components. DFM supports the modelling of multi-state components, which is an advantage in modelling digitally controlled systems because it is often practical to model some of their components with more than two states and more than one failure mode. Another advantage of DFM is that only one model is needed to represent the complete behaviour of a system and therefore different states of the system can be analysed using the same model [12].

A DFM model is a directed graph which consists of discrete-state nodes analysed at discrete time steps and edges that represent the dependencies between nodes. The nodes represent the system's components and variables. DFM models contain two

Table 1
The decision table of deterministic node C.

	Output	Inputs		
Node	C	F	N	C
Time lag		0	0	1
	0	0	−1	0
	0	0	−1	1
	0	0	0	0
	1	0	0	1
	1	0	1	0
	1	0	1	1
	0	1	−1	0
	1	1	−1	1
	0	1	0	0
	1	1	0	1
	0	1	1	0
	1	1	1	1

kinds of nodes: deterministic nodes and stochastic nodes. The state of a deterministic node is determined by states of its input nodes at specified time steps. These dependencies are typically defined by decision tables, such as in Table 1. The state of a stochastic node is determined by a probability model.

In DFM, the top event is defined as particular nodes being in particular states at particular time steps. The top function can be derived from the graph model by tracing it backwards, starting from the top event until a specified initial time. In principle, a fault tree representing the top function could be built directly without the DFM model, but it would be laborious and difficult if the system is not very simple. The top function can be derived automatically from the DFM model.

4.2. Literals in dynamic analysis

To apply Definition 4 in DFM, the concept of a literal has to be generalised even further from the case of multi-state reliability analysis, because nodes are analysed at discrete time steps. Hence, in DFM, a literal is a node in a state at a time step. For example, a literal representing node N in state s at time step $-t$ is denoted as $N_s(-t)$. For literals of a node, it holds that

$$\sum_{s \in S} N_s(-t) = 1 \tag{7}$$

and

$$N_{s_1}(-t) \cdot N_{s_2}(-t) = 0, \tag{8}$$

where S is the set of all states of node C and $s_1 \neq s_2$. The minus sign is used in the time step notation because DFM models are usually analysed deductively from effects to causes.

Stochastic nodes can also have additional time-dependent constraints, such as a non-decreasing property. A typical example is a node that determines whether a non-repairable component is failed. This node is in state 0 at the initial time, and if it turns to state 1 at a later time step, it remains in state 1 for the rest of the scenario. This non-decreasing property can be defined as Boolean equation

$$F_1(-t) \cdot F_0(-t+1) = 0, \tag{9}$$

where F is the node with non-decreasing state and t is an integer, so that $-n \leq -t < 0$ where $-n$ is the initial time step of the analysis. This constraint means that if the node is in state 1 at time step $-t$, it cannot be in state 0 at the next time step $-t+1$.

The following propositions follow from additional constraint (9). The proofs of propositions are presented in Appendix B.

Proposition 1. *From additional constraint* (9) *and literal rule* (7), *it follows that*

$$F_1(-t) \cdot F_0(-t+x) = 0 \tag{10}$$

for all $x \geq 2, x \in \mathbb{N}$.

Proposition 2. *From additional constraint* (9) *and literal rule* (7), *it follows that*

$$F_1(-t) + F_1(-t+x) = F_1(-t+x) \tag{11}$$

for all $x \geq 1, x \in \mathbb{N}$.

Proposition 3. *From additional constraint* (9) *and literal rule* (7), *it follows that*

$$F_0(-t) + F_0(-t+x) = F_0(-t) \tag{12}$$

for all $x \geq 1, x \in \mathbb{N}$.

Proposition 4. *From Eq.* (10), *it follows that*

$$F_1(-t) \cdot F_1(-t+x) = F_1(-t) \tag{13}$$

for all $x \geq 1, x \in \mathbb{N}$.

If F is a binary node that has dynamic constraint (9) and products $A_1(-u) \cdot F_1(-t)$ and $A_1(-u) \cdot F_1(-t+1)$ are implicants of the top event, $A_1(-u) \cdot F_1(-t)$ is not a prime implicant because of (11). Respectively, if products $B_1(-v) \cdot F_0(-t)$ and $B_1(-v) \cdot F_0(-t+1)$ are implicants of the top event, $B_1(-v) \cdot F_0(-t+1)$ is not a prime implicant because of (12).

Different failure modes of a component can be modelled using a multi-state node that has one state for normal operation and another for each failure mode. Each failure mode state functions similar to state 1 of the previously presented non-decreasing binary node. In other words, if the node turns to a failure mode state, it remains in that state for the rest of the scenario. If this node is M, it has additional constraints

$$M_f(-t) \cdot M_s(-t+1) = 0, \tag{14}$$

for each failure mode state f and for each state s for which $f \neq s$. This additional constraint has similar propositions to (9).

Another relevant type of dynamic constraint is a constraint that a node cannot change by more than a particular number of states between time steps [11]. For example, if a valve has states 0 ('closed'), 1 ('slightly open'), 2 ('half open'), 3 ('almost open') and 4 ('fully open'), it could have dynamic constraints

$$V_0(-t) \cdot (V_2(-t+1) + V_3(-t+1) + V_4(-t+1)) = 0, \tag{15}$$

$$V_1(-t) \cdot (V_3(-t+1) + V_4(-t+1)) = 0, \tag{16}$$

$$V_2(-t) \cdot (V_0(-t+1) + V_4(-t+1)) = 0, \tag{17}$$

$$V_3(-t) \cdot (V_0(-t+1) + V_1(-t+1)) = 0 \tag{18}$$

and

$$V_4(-t) \cdot (V_0(-t+1) + V_1(-t+1) + V_2(-t+1)) = 0. \tag{19}$$

These additional constraints have propositions such as

$$V_0(-t) \cdot V_1(-t+1) \cdot V_2(-t+2) = V_0(-t) \cdot V_2(-t+2) \tag{20}$$

and

$$V_0(-t) \cdot V_1(-t+1) \cdot V_2(-t+2) \cdot V_3(-t+3) = V_0(-t) \cdot V_3(-t+3). \tag{21}$$

This means that, for example, an implicant that contains literals $V_0(-t)$, $V_1(-t+1)$ and $V_2(-t+2)$ cannot be an L-minimal implicant because a logically equivalent product can be obtained by removing literal $V_1(-t+1)$, and hence the product is still an implicant if $V_1(-t+1)$ is removed.

4.3. Identification of prime implicants

To identify prime implicants in DFM, there are two different approaches. In [10], the top function of the DFM model is transformed into a timed fault tree from which prime implicants are identified. In another approach implemented in a DFM tool Yadrat [14], the DFM model is transformed into a binary decision diagram (BDD) [20,21] which represents the top function. Additional constraints, such as (9), need to be taken into account either when the top function is processed or when the L-minimal implicants are post-processed.

The analysis of a DFM model starts with the product of the literals that define the top event. The top function is developed by applying the equations that determine the states of the nodes in the top event and the equations that determine the states of the nodes that consequently appear in the top function. The process continues until the top function contains only literals representing deterministic nodes at the initial time and literals representing stochastic nodes.

In Yadrat [14], the DFM model is transformed into a BDD which is a data structure used to represent Boolean functions. BDDs are based on repeated application of the classic Shannon expansion formula

$$F = x \cdot F|_{x=1} + \overline{x} \cdot F|_{x=0}. \tag{22}$$

Variables with more than two states are coded into Boolean vectors so that the number of Boolean variables is minimized. Hence, all the Boolean variables in the BDD do not represent literals that appear in prime implicants. The BDD is constructed based on decision tables (see Table 1) which determine how the states of deterministic nodes depend on the states of input nodes. Decision tables are gone through systematically row by row. At each row all inputs are considered separately. If the input is an output of another decision table, the BDD construction algorithm is recursively called with the input as a parameter. The input variables of each row are combined with AND-operation and the rows are combined with OR-operation.

To account constraint (9), the Yadrat approach [14] modifies the top function (represented by the BDD) so that each literal of the form $F_1(-t)$ is replaced (when the BDD is constructed) by sum

$$F_1(-t)+F_1(-t-1)+\cdots+F_1(-n+1) \tag{23}$$

where $-n$ is the initial time. This can be done because condition $F_1(-t)$ can be caused by $F_1(-u)$, where $t<u$. If this was not done, some prime implicants might be left unidentified. Consider the following top function for example:

$$E_T = F_1(-n+2)\cdot B_1(-n)+F_0(-n+1)\cdot C_2(-n). \tag{24}$$

The prime implicants are $F_1(-n+2)\cdot B_1(-n)$, $F_0(-n+1)\cdot C_2(-n)$ and $B_1(-n)\cdot C_2(-n)$. $B_1(-n)\cdot C_2(-n)$ is a prime implicant because either $F_1(-n+2)$ or $F_0(-n+1)$ is necessarily 1 which follows from (9). The prime implicant algorithms cannot identify this prime implicant without specifically accounting constraint (9) somehow. In the Yadrat approach, the top function is modified to be

$$E_T = (F_1(-n+2)+F_1(-n+1))\cdot B_1(-n)+F_0(-n+1)\cdot C_2(-n) \tag{25}$$

and prime implicant $B_1(-n)\cdot C_2(-n)$ is identified normally because literals $F_1(-n+1)$ and $F_0(-n+1)$ are complementary.

Algorithms that Yadrat uses to identify prime implicants are based on a decomposition theorem [7], which divides L-minimal implicants into three sets with regard to a Boolean variable.

Theorem 1. *The decomposition theorem: Let F_T be a top function, x a Boolean variable and $LMI[F_T]$ a set of L-minimal implicants of F_T. Then*

$$\begin{aligned} LMI[F_T] = {} & LMI[F_T|_{x=1}\cdot F_T|_{x=0}] \\ & \cup \{\overline{x}\}\cdot LMI[F_T|_{x=0}]\backslash LMI[F_T|_{x=1}\cdot F_T|_{x=0}] \\ & \cup \{x\}\cdot LMI[F_T|_{x=1}]\backslash LMI[F_T|_{x=1}\cdot F_T|_{x=0}]. \end{aligned} \tag{26}$$

When Theorem 1 is recursively applied to a top function, all the prime implicants of the top function according to Definition 1 can be found. From this set of implicants, all the real prime implicants (according to Definition 4) can be derived by post-processing. In the post-processing, "implicants" that contradict with additional constraints are removed, unnecessary literals (literals that are implied by other literals in the same implicants) are removed from the implicants, and the implicants are compared to each other with respect to (9) in order to ascertain which of them are prime implicants.

A general algorithm to take Definition 4 and additional constraints into account in the identification of prime implicants should be developed. One possibility would be to generalise the decomposition theorem to take the multi-state logic and additional constraints into account.

4.4. Simple example

Fig. 2 presents a simple DFM model with two nodes, a deterministic node P and non-decreasing binary stochastic node PF. Node P has two states, 0 and 1, and it depends on PF via simple Boolean equations:

$$P_0(-t) = PF_1(-t)+P_0(-t-1), \tag{27}$$

$$P_1(-t) = PF_0(-t)\cdot P_1(-t-1). \tag{28}$$

If the top event is that node P is in state 0 at time step 0 and the initial time is -4, the top function is

$$\begin{aligned} F_T &= P_0(0) = PF_1(0)+P_0(-1) = PF_1(0)+PF_1(-1)+P_0(-2) \\ &= PF_1(0)+PF_1(-1)+PF_1(-2)+P_0(-3) \\ &= PF_1(0)+PF_1(-1)+PF_1(-2)+PF_1(-3)+P_0(-4). \end{aligned} \tag{29}$$

L-minimal implicants of the top event are clearly $PF_1(0)$, $PF_1(-1)$, $PF_1(-2)$, $PF_1(-3)$ and $P_0(-4)$. However, as Eq. (9) holds for PF, products $PF_1(-1)$, $PF_1(-2)$ and $PF_1(-3)$ are not prime implicants due to (11). If state 1 of PF represents the failed state of a pump, prime implicant $PF_1(0)$ indicates that the pump needs to be failed at latest at time step 0 in order to cause the top event. This condition can be satisfied by a failure that occurs either at time step -3, -2, -1 or 0. Similarly, implicant $PF_1(-2)$ indicates that the pump needs to be failed at latest at time step -2. This is a more restrictive condition, because the pump could also fail at either of two later time steps so as to cause the top event, and hence, it is not considered minimal. Prime implicant $PF_1(0)$ implies that conditions $PF_1(-1)$, $PF_1(-2)$, $PF_1(-3)$ also imply the top event. Hence, instead of four prime implicants (according to the traditional definition), only one prime implicant is needed.

5. Prime implicants of a feed water tank system

Fig. 3 presents a feed water tank system [14]. In this system, the water flow to the tank is constant. A regulation valve controls how

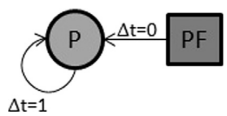

Fig. 2. A simple DFM model from a DFM tool called YADRAT [14].

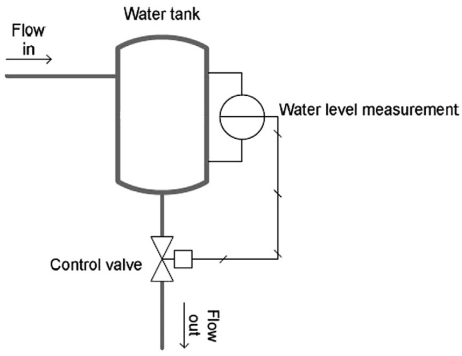

Fig. 3. A feed water tank system [14].

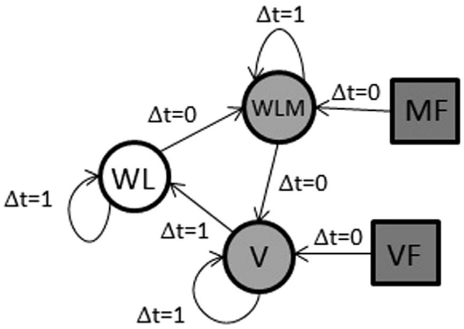

Fig. 4. A DFM model based on the feed water tank system from a DFM tool called YADRAT [14].

much water is released out of the tank. The valve is operated based on the values of the water level measurement. The goal is that the water level neither gets too high nor too low.

Fig. 4 presents a DFM model based on the system. The model contains five nodes: WL, WLM, MF, V and VF. Node WL represents the water level which can be low (state -1), medium (state 0) or high (state 1). Node WLM represents the water level measurement which is given the value of WL at the previous time step unless the water level sensor is failed. Node MF is a non-decreasing binary stochastic node and determines whether the water level measurement is failed. If MF is in state 1, WLM is frozen. Node V represents the valve which can be open (state 1) or closed (state 0). The valve opens if the water level measurement is high, and closes if the water level measurement is low. Node VF is a non-decreasing binary stochastic node and determines whether the valve is failed. If VF is in state 1, V remains in its previous state.

The input dependencies of nodes WL, WLM and V are represented by Boolean equations

$$WL_{-1}(-t) = V_1(-t-1)\cdot(WL_{-1}(-t-1)+WL_0(-t-1)), \quad (30)$$

$$WL_0(-t) = V_0(-t-1)\cdot WL_{-1}(-t-1)+V_1(-t-1)\cdot WL_1(-t-1), \quad (31)$$

$$WL_1(-t) = V_0(-t-1)\cdot(WL_0(-t-1)+WL_1(-t-1)), \quad (32)$$

$$WLM_{-1}(-t) = WL_{-1}(-t)\cdot MF_0(-t)+WLM_{-1}(-t-1)\cdot MF_1(-t), \quad (33)$$

$$WLM_0(-t) = WL_0(-t)\cdot MF_0(-t)+WLM_0(-t-1)\cdot MF_1(-t), \quad (34)$$

$$WLM_1(-t) = WL_1(-t)\cdot MF_0(-t)+WLM_1(-t-1)\cdot MF_1(-t), \quad (35)$$

$$V_0(-t) = V_0(-t-1)\cdot(VF_1(-t)+WLM_0(-t))+WLM_{-1}(-t)\cdot VF_0(-t) \quad (36)$$

and

$$V_1(-t) = V_1(-t-1)\cdot(VF_1(-t)+WLM_0(-t))+WLM_1(-t)\cdot VF_0(-t). \quad (37)$$

The top event to be analysed is that the water level is high (WL is in state 1) at time steps -1 and 0. The initial time is -4. The top function can be derived by applying Eqs. (30)–(37) to formula

$$F_T = WL_1(0)\cdot WL_1(-1) \quad (38)$$

until only initial states of WL, WLM and V and states of MF and VF appear in the formula. The top function and the prime implicant identification process are too complicated to be presented on paper. Therefore, they are skipped. The prime implicants were identified using the Yadrat tool [14] and they are presented in Table 2.

In order to compare the new definition (Definition 4) to the traditional definition (Definition 1), all L-minimal implicants can be identified based the prime implicants of Table 2. This can be done by replacing all literals representing nodes MF and VF by the sums on the right sides of equations in Table 3 and removing non-minimal implicants and those implicants that contradict with constraints (8) and (9). Examples of L-minimal implicants that are not prime implicants according to Definition 4 are presented in Table 4. Only based on prime implicants 1, 2 and 7, 29 L-minimal implicants that are not prime implicants were generated. In total, there are 68 L-minimal implicants that are not prime implicants. This comparison shows that Definition 4 allows the results to be presented in a more compact form than the traditional definition.

Inclusion of some L-minimal implicants in Table 4 in the results is also clearly not sensible. This is evident when L-minimal implicants 1.5–7 from Table 4 are compared to the first prime implicant from Table 2. The only difference between these implicants is the time step of MF_0 (the measurement not failed). If other

Table 2
Prime implicants of the top event.

No.	Prime implicant
1	$\{V_1(-4), VF_0(-3), WL_0(-4), VF_1(-1), MF_0(-3)\}$
2	$\{V_1(-4), VF_0(-3), WL_{-1}(-4), VF_1(-1), MF_0(-3)\}$
3	$\{V_1(-4), VF_0(-3), WL_0(-4), MF_1(-1), MF_0(-3)\}$
4	$\{V_1(-4), VF_0(-3), WL_{-1}(-4), MF_1(-1), MF_0(-3)\}$
5	$\{V_1(-4), VF_0(-3), WL_0(-4), WLM_{-1}(-4), VF_1(-1)\}$
6	$\{V_1(-4), VF_0(-3), WL_{-1}(-4), WLM_{-1}(-4), VF_1(-1)\}$
7	$\{V_1(-4), VF_0(-3), WL_0(-4), WLM_{-1}(-4), MF_1(-1)\}$
8	$\{V_1(-4), VF_0(-3), WL_{-1}(-4), WLM_{-1}(-4), MF_1(-1)\}$
9	$\{VF_0(-3), WL_{-1}(-4), MF_0(-3), VF_1(-2)\}$
10	$\{WL_{-1}(-4), MF_0(-3), V_0(-4), VF_1(-2)\}$
11	$\{VF_0(-3), WL_{-1}(-4), WLM_{-1}(-4), VF_1(-2)\}$
12	$\{VF_0(-3), WL_{-1}(-4), MF_0(-3), MF_1(-2)\}$
13	$\{WL_{-1}(-4), V_0(-4), VF_1(-2), WLM_0(-4)\}$
14	$\{WL_{-1}(-4), WLM_{-1}(-4), V_0(-4), VF_1(-2)\}$
15	$\{V_0(-4), VF_1(-3)\}$
16	$\{WL_{-1}(-4), MF_0(-3), V_0(-4), MF_1(-2)\}$
17	$\{VF_0(-3), WL_{-1}(-4), WLM_{-1}(-4), MF_1(-2)\}$
18	$\{WL_{-1}(-4), V_0(-4), WLM_0(-4), MF_1(-2)\}$
19	$\{WL_{-1}(-4), WLM_{-1}(-4), V_0(-4), MF_1(-2)\}$
20	$\{VF_0(-3), WLM_{-1}(-4), MF_1(-3)\}$
21	$\{V_0(-4), WLM_0(-4), MF_1(-3)\}$
22	$\{V_0(-4), WLM_{-1}(-4), MF_1(-3)\}$

Table 3
Equations to generate all L-minimal implicants.

Equation
$VF_1(-1) = VF_1(-1)+VF_1(-2)+VF_1(-3)$
$VF_1(-2) = VF_1(-2)+VF_1(-3)$
$VF_0(-3) = VF_0(0)+VF_0(-1)+VF_0(-2)+VF_0(-3)$
$MF_1(-1) = MF_1(-1)+MF_1(-2)+MF_1(-3)$
$MF_1(-2) = MF_1(-2)+MF_1(-3)$
$MF_0(-3) = MF_0(0)+MF_0(-1)+MF_0(-2)+MF_0(-3)$

Table 4
Examples of L-minimal implicants that are not prime implicants. These implicants were generated based on prime implicants 1, 2 and 7.

No.	L-minimal implicant
1.1	$\{V_1(-4), VF_0(-3), WL_0(-4), VF_1(-2), MF_0(-3)\}$
1.2	$\{V_1(-4), VF_0(-3), WL_0(-4), VF_1(-1), MF_0(-2)\}$
1.3	$\{V_1(-4), VF_0(-3), WL_0(-4), VF_1(-1), MF_0(-1)\}$
1.4	$\{V_1(-4), VF_0(-3), WL_0(-4), VF_1(-1), MF_0(0)\}$
1.5	$\{V_1(-4), VF_0(-3), WL_0(-4), VF_1(-2), MF_0(-2)\}$
1.6	$\{V_1(-4), VF_0(-3), WL_0(-4), VF_1(-2), MF_0(-1)\}$
1.7	$\{V_1(-4), VF_0(-3), WL_0(-4), VF_1(-2), MF_0(0)\}$
1.8	$\{V_1(-4), VF_0(-2), WL_0(-4), VF_1(-1), MF_0(-3)\}$
1.9	$\{V_1(-4), VF_0(-2), WL_0(-4), VF_1(-1), MF_0(-2)\}$
1.10	$\{V_1(-4), VF_0(-2), WL_0(-4), VF_1(-1), MF_0(-1)\}$
1.11	$\{V_1(-4), VF_0(-2), WL_0(-4), VF_1(-1), MF_0(0)\}$
2.1	$\{V_1(-4), VF_0(-2), WL_{-1}(-4), VF_1(-1), MF_0(-3)\}$
2.2	$\{V_1(-4), VF_0(-2), WL_{-1}(-4), VF_1(-1), MF_0(-2)\}$
2.3	$\{V_1(-4), VF_0(-2), WL_{-1}(-4), VF_1(-1), MF_0(-1)\}$
2.4	$\{V_1(-4), VF_0(-2), WL_{-1}(-4), VF_1(-1), MF_0(0)\}$
2.5	$\{V_1(-4), VF_0(-3), WL_{-1}(-4), VF_1(-1), MF_0(-2)\}$
2.6	$\{V_1(-4), VF_0(-3), WL_{-1}(-4), VF_1(-1), MF_0(-1)\}$
2.7	$\{V_1(-4), VF_0(-3), WL_{-1}(-4), VF_1(-1), MF_0(0)\}$
7.1	$\{V_1(-4), VF_0(-2), WL_0(-4), WLM_{-1}(-4), MF_1(-1)\}$
7.2	$\{V_1(-4), VF_0(-1), WL_0(-4), WLM_{-1}(-4), MF_1(-1)\}$
7.3	$\{V_1(-4), VF_0(0), WL_0(-4), WLM_{-1}(-4), MF_1(-1)\}$
7.4	$\{V_1(-4), VF_0(-3), WL_0(-4), WLM_{-1}(-4), MF_1(-2)\}$
7.5	$\{V_1(-4), VF_0(-2), WL_0(-4), WLM_{-1}(-4), MF_1(-2)\}$
7.6	$\{V_1(-4), VF_0(-1), WL_0(-4), WLM_{-1}(-4), MF_1(-2)\}$
7.7	$\{V_1(-4), VF_0(0), WL_0(-4), WLM_{-1}(-4), MF_1(-2)\}$
7.8	$\{V_1(-4), VF_0(-3), WL_0(-4), WLM_{-1}(-4), MF_1(-3)\}$
7.9	$\{V_1(-4), VF_0(-2), WL_0(-4), WLM_{-1}(-4), MF_1(-3)\}$
7.10	$\{V_1(-4), VF_0(-1), WL_0(-4), WLM_{-1}(-4), MF_1(-3)\}$
7.11	$\{V_1(-4), VF_0(0), WL_0(-4), WLM_{-1}(-4), MF_1(-3)\}$

literals occur, the top event occurs if the measurement does not fail at time step -3. Since the state of MF after time step -3 is irrelevant with regard to top event, it is clearly not sensible to include the L-minimal implicants with literals $MF_0(-2)$, $MF_0(-1)$ and $MF_0(0)$ in the results, even if they imply the top event by implying $MF_0(-3)$.

The above-mentioned L-minimal implicants from Table 4 also would distort probabilistic assessment. Since the top event occurs if the measurement does not fail at time step -3, calculating probabilities for literals $MF_0(-2)$, $MF_0(-1)$ and $MF_0(0)$ and L-minimal implicants 1.5–7 is clearly wrong. The relevant conditions for the top event to occur are in the first prime implicant in Table 2 and L-minimal implicants 1.5–7 are useless for the calculation of the top event probability.

L-minimal implicant 1.1 from Table 4 differs from prime implicant 1 in Table 2 only in that the time step of VF_1 is -2 instead of -1. From the probabilistic assessment point of view, it is correct to include the possibility of the valve failing either at time step -2 or -1. But the probability of literal $VF_1(-1)$ can cover both time steps. In other words, if the failure probability of valve in one time step is 10^{-6}, the probability of literal $VF_1(-1)$ should be approximately $2 \cdot 10^{-6}$ in prime implicant 1. The top event probability would be calculated approximately correctly if both L-minimal implicant 1.1 and prime implicant 1 were included and the probabilities of literals $VF_1(-2)$ and $VF_1(-1)$ were 10^{-6}, but L-minimal implicant 1.1 is not really needed.

6. Conclusions

The paper presented a new definition of a prime implicant that is applicable in dynamic reliability analysis. In this approach, a reliability model consists of a top function and additional constraints. The need for the new definition became evident when non-repairable components were modelled in the dynamic flowgraph methodology and such prime implicants that implied some other prime implicants appeared. The fundamental idea behind the definition is that an implicant that implies some other length-minimal implicant is not a minimal condition that causes the top event, and is, therefore, not a prime implicant.

In addition to dynamic constraints that are related to non-repairable components, reliability models can also contain additional constraints that, for example, prevent some mutually exclusive basic events from appearing in the same prime implicant or represent failure dependencies between components. A typical static reliability model does not, however, contain additional constraints. In that case, the new definition is equal to the traditional definition.

So far, additional constraints, such as the non-repairability property, have been handled using case-specific treatments in identification of prime implicants. Following this paper, development is needed on prime implicant identification algorithms because it should be possible to take additional constraints of any type into account.

The new definition provides solid mathematical foundation for prime implicants in dynamic reliability analyses, where timings of events are taken into account. The definition sets basis for further research as the interpretation of prime implicants affects the computation of risk importance measures, the modelling of component's time-dependent behaviour, the modelling of dependent failures, and the calculation of literal (basic event) probabilities and the total probability.

Acknowledgements

The author would like to thank Jan-Erik Holmberg, Kim Björkman, Ahti Salo, Antti Toppila, Ilkka Karanta and Panu Karhu for their help. The research of the paper was conducted in The Finnish Research Programme on Nuclear Power Plant Safety 2011-2014.

Appendix A. The axioms of Boolean algebra

Boolean variables satisfy the following axioms:

$$a+(b+c)=(a+b)+c, \tag{A.1}$$

$$a \cdot (b \cdot c)=(a \cdot b) \cdot c, \tag{A.2}$$

$$a+b=b+a, \tag{A.3}$$

$$a \cdot b=b \cdot a, \tag{A.4}$$

$$a+0=a, \tag{A.5}$$

$$a \cdot 1=a, \tag{A.6}$$

$$a+(b \cdot c)=(a+b) \cdot (a+c) \tag{A.7}$$

and

$$a \cdot (b+c)=(a \cdot b)+(a \cdot c). \tag{A.8}$$

Boolean variables also satisfy absorption laws:

$$a+ab=a \tag{A.9}$$

and

$$a(a+b)=a. \tag{A.10}$$

Appendix B. Proofs of propositions

B.1. Proposition 1

Proof. The proposition can be proved inductively. First, from additional constraint (9) and literal rule (7), it follows that

$$\begin{aligned}&F_1(-t)\cdot F_0(-t+2)=F_1(-t)\cdot F_0(-t+2)\cdot\\&\quad(F_0(-t+1)+F_1(-t+1))=F_1(-t)\cdot F_0(-t+1)\cdot\\&\quad F_0(-t+2)+F_1(-t)\cdot F_1(-t+1)\cdot F_0(-t+2)=0\cdot\\&\quad F_0(-t+2)+F_1(-t)\cdot 0=0.\end{aligned}\tag{B.1}$$

Second, if $F_1(-t)\cdot F_0(-t+x)=0(x\geq 2, x\in\mathbb{N})$, then

$$\begin{aligned}&F_1(-t)\cdot F_0(-t+x+1)=F_1(-t)\cdot F_0(-t+x+1)\cdot\\&\quad(F_0(-t+x)+F_1(-t+x))=F_1(-t)\cdot F_0(-t+x)\cdot\\&\quad F_0(-t+x+1)+F_1(-t)\cdot F_1(-t+x)\cdot F_0(-t+x+1)=0\cdot\\&\quad F_0(-t+x+1)+F_1(-t)\cdot 0=0.\end{aligned}\tag{B.2}$$

B.2. Proposition 2

Proof. The proposition can be proved inductively. First,

$$\begin{aligned}&F_1(-t)+F_1(-t+1)=F_1(-t)\cdot\\&\quad(F_0(-t+1)+F_1(-t+1))+F_1(-t+1)=F_1(-t)\cdot\\&\quad F_0(-t+1)+F_1(-t)\cdot F_1(-t+1)+F_1(-t+1)=0+F_1(-t)\cdot\\&\quad F_1(-t+1)+F_1(-t+1)=F_1(-t+1).\end{aligned}\tag{B.3}$$

Second, if $F_1(-t)+F_1(-t+x)=F_1(-t+x)(x\geq 1, x\in\mathbb{N})$, then

$$\begin{aligned}F_1(-t)+F_1(-t+x+1)&=F_1(-t)+(F_1(-t+x)+F_1(-t+x+1))\\&=(F_1(-t)+F_1(-t+x))+F_1(-t+x+1)\\&=F_1(-t+x)+F_1(-t+x+1)=F_1(-t+x+1).\end{aligned}\tag{B.4}$$

B.3. Proposition 3

Proof. The proposition can be proved inductively. First, from additional constraint (9) and literal rule (7), it follows that

$$\begin{aligned}&F_0(-t)+F_0(-t+1)=F_0(-t)+F_0(-t+1)\cdot\\&\quad(F_0(-t)+F_1(-t))=F_0(-t)+F_0(-t)\cdot F_0(-t+1)+F_1(-t)\cdot\\&\quad F_0(-t+1)=F_0(-t)+F_0(-t)\cdot F_0(-t+1)+0=F_0(-t),\end{aligned}\tag{B.5}$$

Second, if $F_0(-t)+F_0(-t+x)=F_0(-t)(x\geq 1, x\in\mathbb{N})$, then

$$\begin{aligned}&F_0(-t)+F_0(-t+x+1)=(F_0(-t)+F_0(-t+x))+F_0(-t+x+1)\\&\quad=F_0(-t)+(F_0(-t+x)+F_0(-t+x+1))=F_0(-t)+F_0(-t+x)\\&\quad=F_0(-t).\end{aligned}\tag{B.6}$$

B.4. Proposition 4

Proof.

$$\begin{aligned}F_1(-t)\cdot F_1(-t+x)&=F_1(-t)\cdot F_1(-t+x)+F_1(-t)\cdot F_0(-t+x)\\&=F_1(-t)\cdot(F_1(-t+x)+F_0(-t+x))\\&=F_1(-t).\end{aligned}\tag{B.7}$$

References

[1] Høyland A, Rausand M. System reliability theory: models and statistical methods. Wiley series in probability and mathematical statistics: applied probability and statistics section. New York: Wiley 1994. p. 518. ISBN 0-471-59397-4.

[2] Vesely WE, Goldberg FF, Roberts NH, Haasl DF. Fault tree handbook. Washington, DC: U.S. Nuclear Regulatory Commission; 1981. p. 202. NUREG-0492.

[3] Vesely WE, Stamatelatos M, Dugan J, Fragola J, Minarick J, Railsback J. Fault tree handbook with aerospace applications, Version 1.1. Washington, DC: NASA Office of Safety and Mission Assurance; 2002. p. 205.

[4] van der Borst M, Schoonakker H. An overview of PSA importance measures. Reliab Eng Syst Saf 2001;72:241–5.

[5] Contini S, Cojazzi GGM, Renda G. On the use of non-coherent fault trees in safety and security studies. Reliab Eng Syst Saf 2008;93:1886–95.

[6] Rauzy A. Mathematical foundation of minimal cutsets. IEEE Trans Reliab 1997;50:127–44.

[7] Rauzy A, Dutuit Y. Exact and truncated computation of prime implicants of coherent and non-coherent fault trees within Aralia. Reliab Eng Syst Saf 1997;58:127–44.

[8] Cepek O, Kucera P, Kurik S. Boolean functions with long prime implicants. Inf Process Lett 2013;113:698–703.

[9] Labeau PE, Smidts C, Swaminathan S. Dynamic reliability: towards an integrated platform for probabilistic risk assessment. Reliab Eng Syst Saf 2000;68:219–54.

[10] Garrett CJ, Guarro SB, Apostolakis GE. The dynamic flowgraph methodology for assessing the dependability of embedded software systems. IEEE Trans Syst Man Cybern 1995;25:824–40.

[11] Yau M, Apostolakis G, Guarro S. The use of prime implicants in dependability analysis of software controlled systems. Reliab Eng Syst Saf 1998;62:23–32.

[12] Al-Dabbagh AW, Lu L. Reliability modeling of networked control systems using dynamic flowgraph methodology. Reliab Eng Syst Saf 2010;95:1202–9.

[13] Al-Dabbagh AW, Lu L. Dynamic flowgraph modeling of process and control systems of a nuclear-based hydrogen production plant. Int J Hydrogen Energy 2010;35:9569–80.

[14] Björkman K. Solving dynamic flowgraph methodology models using binary decision diagrams. Reliab Eng Syst Saf 2013;111:206–16.

[15] Tyrväinen T. Risk importance measures in the dynamic flowgraph methodology. Reliab Eng Syst Saf 2013;118:35–50.

[16] Tyrväinen T, Björkman K. Modelling common cause failures and computing risk importance measures in the dynamic flowgraph methodology. In: Proceedings of the 11th International probabilistic safety assessment and management conference & the annual European safety and reliability conference, June 25–29, Helsinki, Finland. Helsinki: The International Association for Probabilistic Safety Assessment and Management (IAPSAM); 2012. 30-Th4-1.

[17] Garribba S, Guagnini E, Mussio P. Multiple-valued logic trees: meaning and prime implicants. IEEE Trans Reliab 1985;R-34:463–72.

[18] Natvig B, Sørmo S, Høgsen Holen A. Multi-state reliability theory—a case study. Adv Appl Prob 1986;18:921–32.

[19] Yueqin W, Yun F, Qihua W. A universal generating function approach for reliability analysis of multi-state systems. In: 2010 second WRI global congress on intelligent systems, vol. 3, December 16–17 2010; Wuhan, China. Los Alamitos: IEEE Computer Society; 2010. p. 207–10.

[20] Bryant RE, Bryant RE. Graph-based algorithms for Boolean function manipulation. Computers 1986;C-35:677–91.

[21] Bryant RE. Symbolic Boolean manipulation with ordered binary-decision diagrams. ACM Comput Surv 1992;24:293–318.

Publication II

Tyrväinen, T. Risk importance measures in the dynamic flowgraph methodology. *Reliability Engineering and System Safety*, Vol. 118, pp. 35-50, doi: 10.1016/j.ress.2013.04.013, October 2013.

Reliability Engineering and System Safety 118 (2013) 35–50

Contents lists available at SciVerse ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

Risk importance measures in the dynamic flowgraph methodology

T. Tyrväinen *

VTT Technical Research Centre of Finland, Systems Research, P.O. Box 1000, FI-02044 Espoo, Finland

ARTICLE INFO

Article history:
Received 18 June 2012
Received in revised form
15 April 2013
Accepted 15 April 2013
Available online 28 April 2013

Keywords:
Dynamic flowgraph methodology
Risk importance measure
Multi-state

ABSTRACT

This paper presents new risk importance measures applicable to a dynamic reliability analysis approach with multi-state components. Dynamic reliability analysis methods are needed because traditional methods, such as fault tree analysis, can describe system's dynamical behaviour only in limited manner. Dynamic flowgraph methodology (DFM) is an approach used for analysing systems with time dependencies and feedback loops. The aim of DFM is to identify root causes of a top event, usually representing the system's failure. Components of DFM models are analysed at discrete time points and they can have multiple states. Traditional risk importance measures developed for static and binary logic are not applicable to DFM as such. Some importance measures have previously been developed for DFM but their ability to describe how components contribute to the top event is fairly limited. The paper formulates dynamic risk importance measures that measure the importances of states of components and take the time-aspect of DFM into account in a logical way that supports the interpretation of results. Dynamic risk importance measures are developed as generalisations of the Fussell-Vesely importance and the risk increase factor.

1. Introduction

Risk importance measures [1–4] are important in the reliability analysis of complex systems, such as safety systems in nuclear power plants. They can be used to analyse which components or basic events are most important with regard to the system's reliability, the probability that the system does not fail. The importance of a component depends not only on the reliability of the component but also on the impact of its behaviour on the consequences of interest. Risk importance measures reveal which are the beneficial ways to improve the system's reliability.

Dynamic reliability analysis methods [5] have been studied extensively since 90s because traditional methods, such as fault tree analysis, can describe system's dynamical behaviour only in limited manner. Dynamic methods can represent system's evolution in time more accurately than traditional methods and they can be used to identify time-dependent failure condition combinations that cause the system's failure. There are well-established techniques for the computation of risk importance measures in fault tree and event tree analyses [6,7]. However, in dynamic reliability analysis, fewer importance measures have been developed [8]. The limitations are even more evident with regard to dynamic reliability analysis approaches that include components with more than two states.

This paper presents two new risk importance measures for a dynamic reliability analysis approach called dynamic flowgraph methodology (DFM) [9–12]. A brief conference paper on these importance measures was already published in 2012 [13] but this paper presents the work in a comprehensive and more general form.

Risk importance measures are typically calculated from minimal cut sets which are usually the most essential result of reliability analysis. A minimal cut set is a minimal combination of basic events that is sufficient to cause the top event. If one of the basic events is taken away from a minimal cut set, the remaining combination of basic events is not sufficient to cause the top event anymore.

Two risk importance measures are generalised into the dynamic and multi-state case of DFM: the Fussell-Vesely measure and the risk increase factor (also known as the risk achievement worth). They are the most often used risk importance measures in the reliability analysis of nuclear power plants [1]. These two measures form a combination that can describe fully the influence of the component's unavailability. Fussell-Vesely measures how large portion of the top event probability is caused by the minimal cut sets that contain a given basic event and the risk increase factor measures how much the probability of the top event increases if a given basic event occurs. Hence, Fussell-Vesely measures the direct effect of the component's unavailability, whereas the risk increase factor depends more on the component's position in the system's structure and the reliability of other components.

* Tel.: +358 407760841.
E-mail address: tero.tyrvainen@vtt.fi

0951-8320/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.ress.2013.04.013

If P, V, L and M are basic events and PL, VL, PM and VM are minimal cut sets, Fussell-Vesely of basic event V can be calculated (using so-called "MCS upper bound" to calculate the top event probability) as presented in (1) and the risk increase factor as presented in (2).

$$I^{FV}(V)=\frac{Q^{V}_{TOP}}{Q_{TOP}} =\frac{1-(1-Q(VL))\cdot(1-Q(VM))}{1-(1-Q(PL))\cdot(1-Q(VL))\cdot(1-Q(PM))\cdot(1-Q(VM))}, \quad (1)$$

where the notation $Q(CS)$ means the probability of minimal cut set CS and Q_{TOP} is the top event probability.

$$I^{I}(V)=\frac{Q_{TOP}(V=1)}{Q_{TOP}} =\frac{1-(1-Q(L))\cdot(1-Q(M))}{1-(1-Q(PL))\cdot(1-Q(VL))\cdot(1-Q(PM))\cdot(1-Q(VM))}. \quad (2)$$

Components can usually fail in more than one way. For example, a valve could be failed open or close. The state in which a component is failed is called a failure state in this paper. The paper concentrates on calculating the new risk importance measures for failure states of components in DFM.

The paper is structured as follows. Section 2 briefly presents dynamic flowgraph methodology. Section 3 reviews previous research and specifies the objectives of the paper. New dynamic risk importance measures are formulated in Sections 4 and 5 and a case study is presented in Section 6. The significance of the dynamic risk importance measures and possibilities for further research are discussed in Section 7, and Section 8 concludes the study.

2. Dynamic flowgraph methodology

Dynamic flowgraph methodology [9–12] is an approach for analysing systems with time dependencies and feedback loops. It is typically used to model and analyse digitally controlled systems which include both hardware and software components. For example, modern nuclear power plants include digitally controlled safety systems. The multi-state logic of DFM is an advantage in modelling of that kind of systems because their components generally do not behave in binary manners. Another advantage of DFM is that only one model is needed to represent the complete behaviour of a system and different states of the system can be analysed using the same model [10].

A DFM model is a directed graph which consists of nodes that represent the system's components and variables and edges that represent the dependencies between nodes. A node can have a finite number of states and the state of a node is determined either by a probability model or by states of its input nodes at specified time steps. Input dependencies of a node are represented in a decision table which is an extension of a truth table.

The aim of DFM is to identify root causes for a top event, which is defined as a condition that particular nodes are in particular states at particular time steps. The result is a set of prime implicants which are generalisations of minimal cut sets. A prime implicant is a minimal combination of basic events and other conditions that is sufficient to cause the top event. In DFM analysis, a basic event or a condition is represented as a literal which is a node in a state at a time step and a prime implicant is a set literals. Hence, prime implicants of DFM can be understood as timed minimal cut sets.

An example on DFM analysis results is provided next. To keep the focus on concepts that are most relevant with regard to this paper, the actual DFM model is not presented at this point. Expression $F(-2)=0$ is a literal representing node F in state 0 at time step −2, and $\{N(-3)=-1, T(-3)=1, R(-3)=1, F(-2)=0, F(-1)=1\}$ and $\{C(-3)=0, T(-3)=-1, R(-3)=0, R(-2)=1\}$ are prime implicants of top event $\{T(-1)=1, T(0)=1\}$ of an example system that is presented later in Section 4.5 if the initial time is −3. If the state of a node is determined by its input nodes, the node can appear in prime implicants only at the initial time, such as N, T, and C in this example. Negative time steps are used in this paper, because the same notation has been used in previous DFM papers [9–12] because DFM models are mostly analysed deductively from effects to causes.

3. Towards dynamic risk importance measures

3.1. Previous research

In DFM modelling, risk importance measures need to be constructed so that they map information from prime implicants to values that represent the significances of different components. Thus, the time aspect of DFM should be taken into account in dynamic risk importance measures as well as the multi-state logic.

In the context of DFM, not much research has been conducted on risk importance measures. Refs. [14] and [15] present some DFM importance measures as generalisations of fault tree analysis importance measures. The importance measures presented in [14] measure the importances of different nodes in DFM models. They do not consider which state or states of a node appear in prime implicants even though the state information can play an important role in the interpretation of DFM results. Significances of nodes they provide can be useful but their ability to describe how components contribute to the top event is fairly restricted in many cases.

In [15], importance measures are formulated for literals of DFM models. They consider each time point separately while analysts are mainly interested in the overall importances of nodes and states of nodes. For example, if literals $V(-t_1)=s$ and $V(-t_2)=s$ represent node V in state s at time steps $-t_1$ and $-t_2$, importance measures are only calculated for these literals separately even though they represent the same condition. The importance of node V or state s is not directly provided. This paper aims to develop risk importance measures that measure the importances of states of nodes and still maintain the information about time steps of literals in the results.

3.2. Other methodologies

Markov models constitute a dynamic reliability analysis approach that is comparable to DFM [16]. Markov models can be used to analyse dynamic multi-state systems as DFM models. Some studies have been carried out on risk importance measures for Markov models [8,17,18]. However, it would not be practical to use similar importance measures in DFM because they rely on the perturbation of transition rate matrices of Markov models and DFM models are not based on transition rates.

There are two types of importance measures for multi-state systems [19]. Measures of type 1 are formulated for components and they measure the significance that a component has to the system's reliability as a whole. Type 1 measures are useful when analysing whether the number of redundant components needs to be increased. Measures of type 2 are formulated for states of components and they measure how a certain state or states of a component affect the system's reliability. Different states of a component can assume quite different values of a type 2 measure. For example, the top event probability might increase if a valve is in state 'failed-close' but the same analysis might show that the top event probability decreases if the valve is in state 'failed-open'. Type 2 measures provide guidance on how a component should be

changed so that the system's reliability improves. Many risk importance measures of traditional fault tree analysis can be generalised for multi-state systems.

An approach of type 2 is to transform a multi-state component into a binary component by dividing the states into two sets with regard to a specified performance level [20]. For example, if a pump can function in 'low', 'medium', 'high' or 'very high' power, states 'low' and 'medium' could form one group and states 'high' and 'very high' one group. When a multi-state component is treated as a binary component, traditional risk importance measures can be applied to it. Another approach to measure the importance of a multi-state component is composite importance measures [21], which are weighted averages of type 2 state importances and represent type 1 measures. For example, an average of the risk increase factors of valve's 'failed-close', 'failed-open' and possible other states weighted with state probabilities could be calculated to get a value that measures the significance of the valve as a whole. Both approaches could be applied in DFM.

The modelling of the time-dependent failure behaviour of a component can be performed similarly in the reliability analyses of multi-phase missions [22] and in DFM, even though only a single mission is considered in DFM. In multi-phase missions, basic events can occur during different phases similarly to time steps in DFM. In [23], importance measures are formulated for phase specific events. Correspondingly, in DFM, importance measures can be formulated for time step specific events.

3.3. Conclusion on objectives

The dynamic risk importance measures, the dynamic Fussell-Vesely and the dynamic risk increase factor, are formulated for the different states of nodes. The dynamic Fussell-Vesely is formulated also for different time steps and the dynamic risk increase factor takes the time aspect into account in its own way.

The paper especially focuses on computing dynamic risk importance measures for failure states of components. It is presented how the information about failure states in prime implicants can be tracked by tracing the graph model backwards and utilised to calculate risk importance measures to provide more specific information on how components contribute to the top event probability.

4. The dynamic Fussell-Vesely

4.1. The basic form

Dynamic risk importance measures need to take the multi-state logic and time aspect of DFM into account in a logical way that supports the interpretation of results. For coherent systems, Fussell-Vesely can be interpreted as how much the top event probability would relatively decrease if a component was perfect. In coherent systems, only component failures can cause the top event but in incoherent systems, a failure of a component may actually prevent the top event from occurring and the act of repairing might cause the top event [24]. In multi-state reliability analysis, a system can be defined as coherent if only one state per node appears in prime implicants. Even though DFM models are usually incoherent, they can be coherent with regard to some nodes.

The dynamic Fussell-Vesely (DFV) should be constructed in such a way that the idea about the decrease of the top event probability is maintained. For systems that are coherent with regard to the considered state of the node, the DFV should be possible to interpret so that it indicates how much the top event probability would decrease if the node could be made not to be in the considered state at least until a particular time step. Thus, as the definition of Fussell-Vesely deals with minimal cut sets that include a particular component failure, the definition of the DFV should consider prime implicants that include a particular node in a particular state before or at a particular time step.

Let us assume that the time step of the latest literal in the top event is 0 meaning that it is the last time step of the analysis and the initial time is $-n$ ($n \in \mathbb{N}$). These notations are valid for the rest of the paper. The dynamic Fussell-Vesely is defined in its basic form in Definition 1.

Definition 1. The dynamic Fussell-Vesely measure of state s of node i at time step $-t$ is

$$I^{DFV}(i(-t)=s) := \frac{Q_{TOP}^{i(-t)=s}}{Q_{TOP}}, \tag{3}$$

where Q_{TOP} is the top event probability and $Q_{TOP}^{i(-t)=s}$ is the probability that a prime implicant, including node i in state s before or at time step $-t$ ($0 \le t \le n$), causes the top event.

When time steps of literals are not considered interesting, all the attention can be paid to $I^{DFV}(i(0)=s)$ because it takes all time steps into account.

4.2. Importances of failure states

In DFM, components are often modelled with two nodes: one that represents the functional state of a component and the other that determines if the component is failed or not. Let the following definitions apply for the rest of the paper:

- The node whose state determines if the component is failed or not is called a 'failure node'.
- A component is failed when the failure node is in state 1 and it functions normally when the failure node is in state 0.
- The initial state of a failure node is 0.
- The time lag of a failure node is 0.
- The node that defines the functional state of a component is called a 'component node'.

These definitions are used in the DFM tool YADRAT [12]. With these definitions, a failure event can be interpreted as a change of failure node's state from 0 to 1. When a failure node is in state 0, the component is in one of its normal states determined by the component node. The failure state is defined by the combination of a failure node being in state 1 and the state of the component node. For example, if a component node of a water level measurement sensor has states 'low', 'medium' and 'high', the water level measurement sensor component has normal states 'low', 'medium' and 'high' and failure states 'failed-low', 'failed-medium' and 'failed-high'.

If a component is modelled using two nodes, the failure state of a component cannot directly be read from a prime implicant. A prime implicant only shows that the failure node is in state 1. The failure state can depend on the initial states of nodes and other literals that appear in the prime implicant. The dynamic Fussell-Vesely cannot therefore directly be calculated for failure states according to Definition 1. Eq. (4) presents the specific definitions of the dynamic Fussell-Vesely for a failure state of a component:

$$I_{fs}^{DFV}(i(-t)=s) := \frac{Q_{TOP}^{f(-t)=1,\ i(-t)=s}}{Q_{TOP}}, \tag{4}$$

where f is a failure node connected to component node i and $Q_{TOP}^{f(-t)=1,\ i(-t)=s}$ is the probability that a prime implicant, including a failure in state s of component node i before or at time step $-t$ ($0 \le t < n$), causes the top event.

4.3. Measuring incoherency of a component

When a failure node is in state 0, the corresponding component is functioning as it is meant to. It might be interesting to know if a system is incoherent with regard to a failure of a given component. It could therefore be useful to measure how much state 0 of a failure node contributes to the top event. However, if the dynamic Fussell-Vesely of state 0 of a failure node was calculated according to Definition 1, the interpretation of the result would not come naturally because a failure node is defined to be initially in state 0. But, let the time aspect be inverted so that the definition considers prime implicants that contain a failure node in state 0 at time step $-t$ or later instead of at time step $-t$ or before. In this case, the measure can be interpreted as how much the top event probability relatively decreases if the component fails at a given time step at the latest, if the system is coherent with regard to state 0 of the failure node. The DFV measure is formulated for state 0 of a failure node:

$$I_0^{DFV}(f(-t)=0) := \frac{Q_{TOP}^{f(-t)=0}}{Q_{TOP}}, \tag{5}$$

where $Q_{TOP}^{f(-t)=0}$ is the probability that a prime implicant, including a literal representing failure node f in state 0 at time step $-t$ or later, causes the top event.

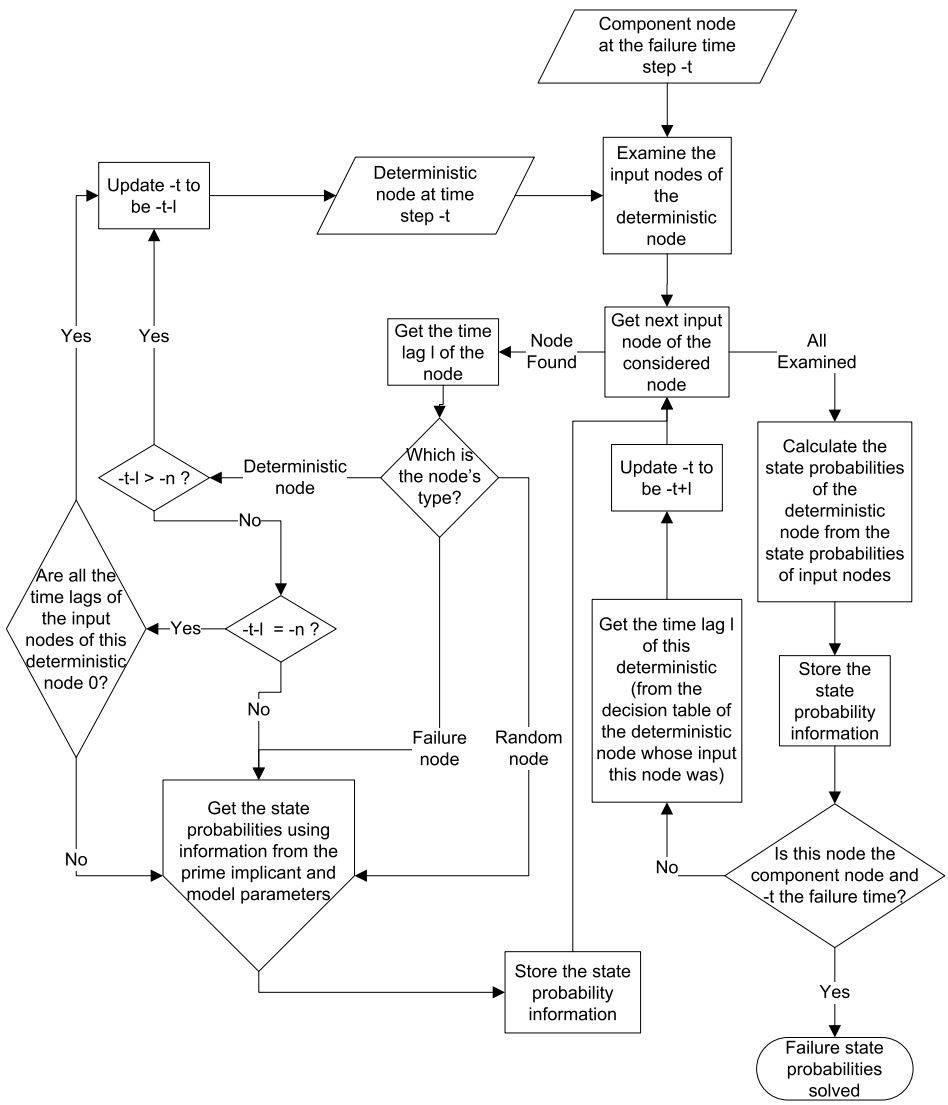

Fig. 1. A flow chart of the backtracking process to solve failure state probabilities. The calculation of the state probabilities of failure nodes, random nodes and deterministic nodes at the initial time is illustrated in its own flow chart in Fig. 2.

4.4. Computation

In the calculation of the dynamic Fussell-Vesely, each prime implicant is examined. If a prime implicant contains the considered node in the considered state (or the considered component in the considered failure state) in the considered time frame, its contribution is added to the DFV.

It is possible that the failure state of the considered component is not unambiguous in a prime implicant. A failure can lead to different states of the component node in separate scenarios. Probabilities for different failure states are therefore solved and a prime implicant's contribution to a failure state DFV presented in (4) is the probability of the prime implicant multiplied by the probability that the component fails to the considered failure state.

The failure state probabilities can be solved deductively by backtracking the model or inductively by simulating different scenarios. The deductive approach is chosen here for the same reason of why DFM models are most often analysed deductively: the number of scenarios to be simulated grows easily large with complex models. The backtracking process to solve failure state probabilities starts from the component node at the failure time step. The state probabilities of the component node can be calculated when the state probabilities of its input nodes are known. Hence, the state probabilities of the input nodes are

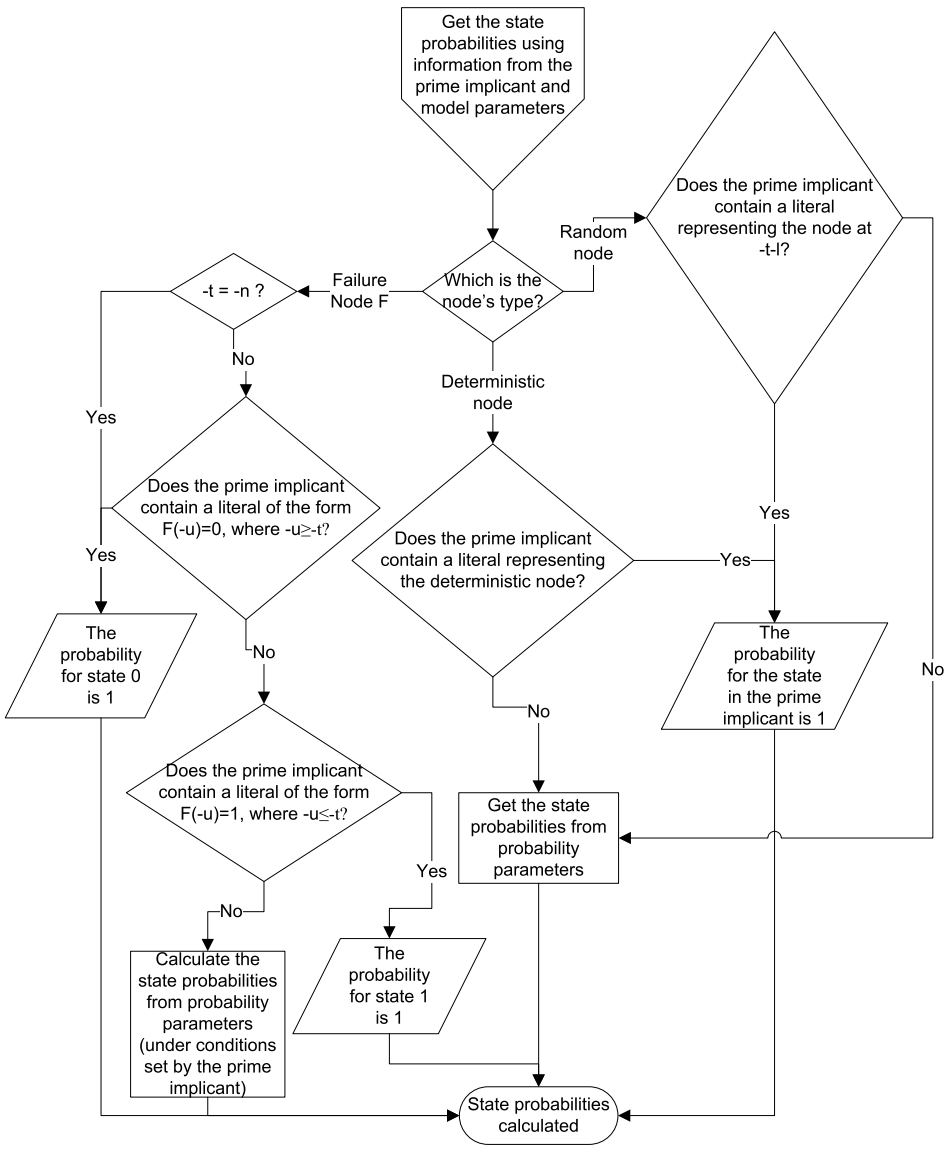

Fig. 2. A flow chart of the calculation of the state probabilities of failure nodes, random nodes and deterministic nodes at the initial time. This flow chart is a subpart of the flow chart presented in Fig. 1 and it takes the node, time step −t and time lag l as inputs from Fig. 1.

needed and they can be calculated from the state probabilities of the input nodes of the input nodes of the component node. Because of this, the backtracking algorithm is based on the recursive calling of a function that calculates the state probabilities of an output node from the state probabilities of the input nodes. When the state probabilities of input nodes are known, the state combination probabilities of input nodes can be calculated and the probability of an output state is the sum of probabilities of those input state combinations that lead to the considered output state.

The recursive calling of the function continues till the initial time is reached or the considered node is a stochastic node. In these cases, the state probabilities are obtained from the probability model of the node. In the calculation of the DFV, the backtracking is performed under the conditions set by a prime implicant. This means that the states of some nodes are known and the state probabilities are obtained from this information, not from the probability model or calculated as conditional probabilities if the prime implicant information does not imply a certain state but affects the probability model.

The backtracking algorithm implemented in the YADRAT tool is illustrated using flow charts in Figs. 1 and 2. YADRAT contains three types of nodes: deterministic nodes, failure nodes and random nodes. The state of a deterministic node is determined by its input nodes through a decision table, except at the initial time step at which the state is determined by a probability model. A failure node is a non-decreasing binary node that is initially in state 0 and whose state is determined by a probability model. A random node is a multi-state stochastic node whose state is determined by a probability model. Each node type is treated differently in the backtracking.

4.5. Backtracking example

Fig. 3 shows an example of a DFM model based on a tank system with a digitally controlled valve and Table 1 gives an example of a decision table. In the model, node C represents the functional state of a valve, N represents water level measurement value and T represents water level. Nodes F and R determine if the valve and the water level measurement are failed and they change states stochastically. Each row of the decision table represents a state combination of input nodes (F, N and C) and the output column determines to which state of the output node C each state combination of input nodes leads to. The time lag row determines the delays in the dependencies between the input nodes and the output node. In Table 1, node C depends on its own state at the previous time step because the time lag is 1. From Table 1, it can be seen that the valve remains in failure state 'failed-0' or 'failed-1' after the failure, because node C stays in the same state it was at the previous time step when F is in state 1.

A set of literals $\pi=\{N(-3)=-1, T(-3)=1, R(-3)=1, F(-2)=0, F(-1)=1\}$ is a prime implicant of top event $\{T(-1)=1, T(0)=1\}$ of

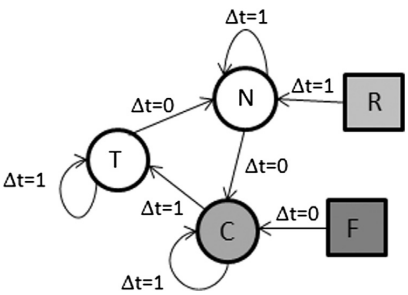

Fig. 3. A DFM model from the DFM tool YADRAT.

Table 1
The decision table of component node C.

Node	Output	Inputs		
	C	F	N	C
Time lag		0	0	1
	0	0	−1	0
	0	0	−1	1
	0	0	0	0
	1	0	0	1
	1	0	1	0
	1	0	1	1
	0	1	−1	0
	1	1	−1	1
	0	1	0	0
	1	1	0	1
	0	1	1	0
	1	1	1	1

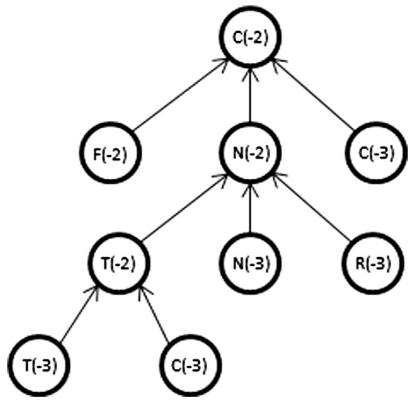

Fig. 4. The backtracking tree starting from the node C at time step −2.

the system presented in Fig. 3 when the initial time is −3. Node F is the failure node of the valve. The failure time is therefore −1. Hence, the failure state probabilities are the state probabilities of $C(-1)$. Under the failure condition, component node C remains in the same state as it was at the previous time step. Thus, the failure state probabilities are the state probabilities of $C(-2)$. As node C is a deterministic node, its state probabilities are defined by the state probabilities of its input nodes. The state probabilities of $F(-2)$, $N(-2)$ and $C(-3)$ therefore have to be solved. Probability $Q(F(-2)=0 \mid \pi)=1$ because prime implicant π contains a literal $F(-2)=0$. Node N is a deterministic node. Hence, the state probabilities of its input nodes should be examined to determine its state probabilities at time step −2. The initial state of component node C does not appear in prime implicant π. The state probabilities of $C(-3)$ are therefore obtained from the probability parameters. If $Q(N(-2)=s \mid \pi)=n_{2,s}$ for all $s\in\{-1,0,1\}$, $Q(C(-3)=r)=c_{3,r}$ for all $r\in\{0,1\}$, $n_{2,-1}+n_{2,0}+n_{2,1}=1$ and $c_{3,0}+c_{3,1}=1$, Table 1 indicates that $Q(C(-2)=0 \mid \pi)=n_{2,-1}+n_{2,0}c_{3,0}$ and $Q(C(-2)=1 \mid \pi)=n_{2,1}+n_{2,0}c_{3,1}$. The backtracking structure of this example is illustrated in Fig. 4. The progression of the backtracking algorithm is illustrated more closely in Appendix A.

4.6. Solving accurate failure state probabilities

Sometimes the solving of failure state probabilities has to be divided into different scenarios to obtain correct results. When the

backtracking process diverges, different branches are examined independently even though they can contain the same nodes. This type of dependencies are taken into account by backtracking the model to identify those nodes that appear in multiple branches and solving failure state probabilities separately in different scenarios related to them. Here, a scenario means that the nodes are set to particular states at particular time steps and the backtracking is performed under that state combination assumption. The probabilities of the state combinations are calculated and accurate failure state probabilities are computed as weighted sums of failure state probabilities related to different scenarios.

There are cases in which the accurate failure state probabilities can be solved by a single backtracking without dividing the solving process into different scenarios and only applying the algorithm presented in Figs. 1 and 2. If a single backtracking gives an unambiguous failure state, the result is always accurate because the backtracking covers all the possible scenarios. However, in some cases, the backtracking can imply a possibility of a failure state that is not really possible. Hence, it is computationally most efficient to calculate the failure state probabilities first with a single backtracking and if the failure state is not unambiguous, calculate accurate probabilities by examining different scenarios separately.

4.7. The non-decreasing property of failure nodes

The non-decreasing property of a failure node has to be taken into account in the DFV calculation. Let F be a failure node and $-t$ a time step. Literal $F(-t)=1$ represents a condition that the corresponding component is failed at time step $-t$. Due to the non-decreasing property of failure nodes, this condition can be satisfied by a failure that occurs at time step $-t$ or earlier. Hence, when DFV values are calculated for the state 1 of F, it must be taken into account that condition $F(-t)=1$ can be caused by a failure at an earlier time step. For this reason, a prime implicant that includes condition $F(-t)=1$ also contributes to the DFV values of earlier time steps than $-t$. If prime implicant π includes literal $F(-t)=1$ and $-u<-t$, the contribution of π to $I^{DFV}(F(-u)=1)$ is $Q(F(-u)=1\mid\pi)\cdot Q(\pi)$. Conditional probability $Q(F(-u)=1\mid\pi)$ is usually

$$\frac{Q(F(-u)=1)}{Q(F(-t)=1)} \tag{6}$$

but if prime implicant π includes, for example, literal $F(-u)=0$, $Q(F(-u)=1\mid\pi)=0$ as condition $F(-u)=0$ implies that the component must be functioning at time step $-u$.

Let s be a state of component node C. The contribution of prime implicant π to $I^{DFV}_{fs}(C(-u)=s)$ is

$$Q(F(-u)=1\mid\pi)\cdot Q(\pi)\cdot Q(C(-u)=s\mid\pi_{F(-u)=1}), \tag{7}$$

where $\pi_{F(-u)=1}$ is a modified version of prime implicant π that includes literal $F(-u)=1$ instead of $F(-t)=1$.

5. The dynamic risk increase factor

5.1. Definition

The risk increase factor measures how much the unavailability of a system increases if a component fails. Thus, in the calculation of the risk increase factor it must be assumed that a component is failed. In DFM, a component can fail at different time steps. The failure of a component does not usually cause system's failure immediately. To provide all the available time for the failure to affect the system, let the dynamic risk increase factor (DRIF) be defined so that the component fails at the earliest possible time step. When a component is failed, it remains failed for the rest of the scenario.

To formulate a more general version of DRIF, let the idea that the condition lasts the whole scenario be applied to a state of a node and the dynamic risk increase factor be formulated so that it measures how much the top event probability increases if the considered node is in the considered state at all time steps. The definition is presented in Definition 2.

Definition 2. The dynamic risk increase factor for a state of a node is

$$I^{DI}(i=s):=\frac{Q_{TOP}(i(-t)=s,\ \forall\ t\in\{0,1,\ldots,m-1,m\})}{Q_{TOP}}, \tag{8}$$

where $Q_{TOP}(i(-t)=s,\ \forall\ t\in\{0,1,\ldots,m-1,m\})$ is the probability that the top event occurs assuming that a node i is in state s at every time step starting from $-m$ ($0\le m\le n$) which is the earliest possible time step for the node i to be in state s considering the initial conditions.

The earliest possible time step for a node to be in a state can vary. For example, a failure node is defined to be initially in state 0. Hence, it can be in state 1 only starting from time step $-n+1$. Similarly, for random nodes and component nodes, all states might not be initially possible.

5.2. Computation

In the calculation of the DRIF, a conditional top event probability is needed. Possibilities are to identify new prime implicants of the conditional top event either by deriving them from originally identified prime implicants or performing a completely new DFM analysis with a modified top event or develop an algorithm to calculate the conditional top event probability "directly" from the model without identifying prime implicants first. A completely new DFM analysis for each DRIF value would be too time-consuming. The computation of the top event probability without identifying prime implicants is an interesting problem that could be examined in the future but in this paper, prime implicants of the conditional top event are derived from the original prime implicants which is more straightforward.

Prime implicants of the conditional top event can be derived from the original prime implicants in the following way. The prime implicants are examined one by one. Those prime implicants that contradict with the condition (contain a node at a specific time step in a wrong state) are removed and the literals that appear in the condition are removed from the prime implicants because their conditional probability is 1. After this, if accurate results are wanted, all duplicate prime implicants and implicants that are not prime implicants anymore have to be removed from the set. This can be done by comparing changed prime implicants with each other and comparing the untouched prime implicants with the remaining changed prime implicants. Fig. 5 presents a flow chart to illustrate the identification of new prime implicants.

Table 2 presents the prime implicants of the top event $\{T(-1)=1, T(0)=1\}$ of the system presented in Fig. 3. When the DRIF is calculated for the failure of the valve, the condition is $\{F(-2)=1, F(-1)=1, F(0)=1\}$. When the new prime implicants are derived, prime implicants 10 and 13 from Table 2 are removed because they contain literal $F(-2)=0$. Literals $F(-2)=1$ and $F(-1)=1$ are removed from other prime implicants. 11 non-minimised implicants are left (Table 3). When these implicants are compared to each other, it is noticed that only $\{C(-3)=0\}$ is a prime implicant and others are non-minimal implicants.

The dynamic risk increase factor can be calculated for a failure state of a component assuming that the failure node is in state 1 starting from the first time step after the initial time and that the

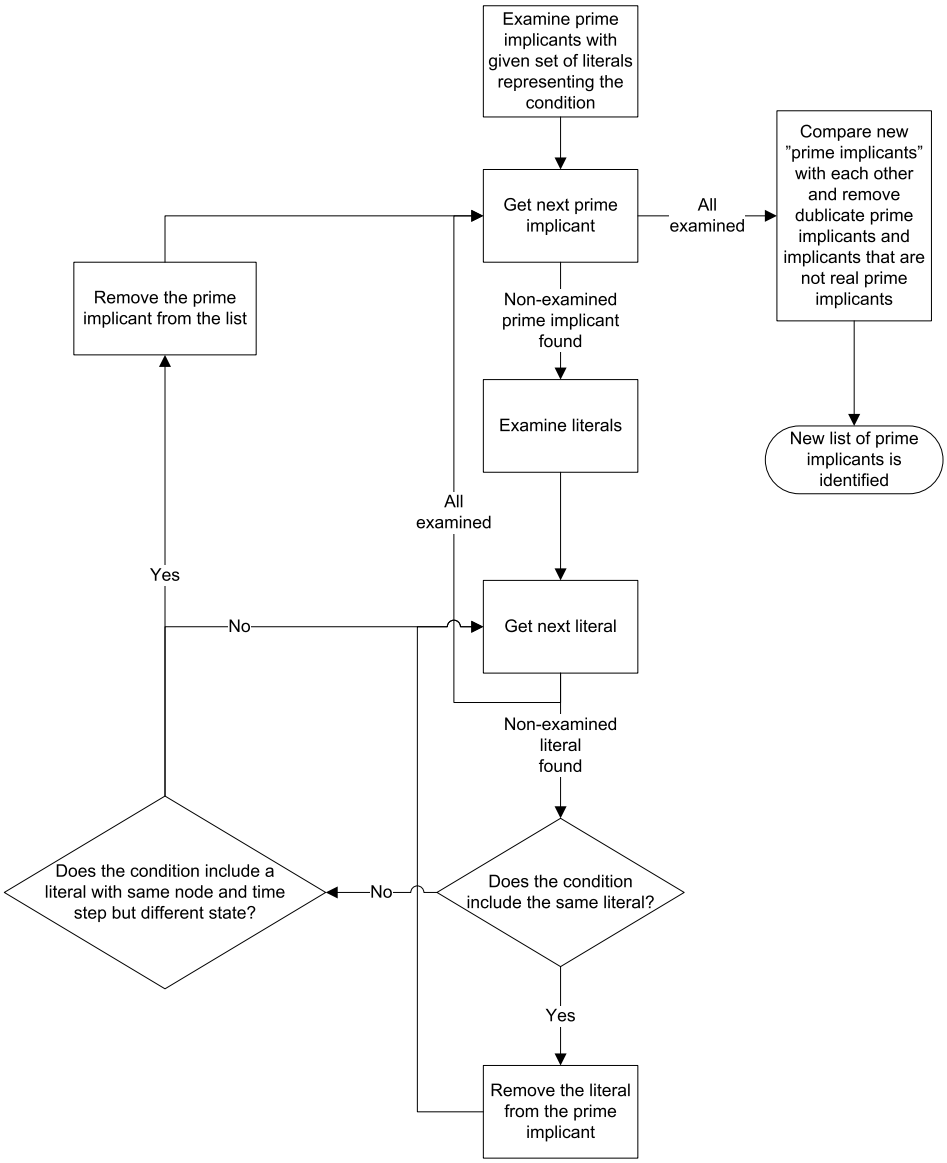

Fig. 5. A flow chart representing the identification process of prime implicants of the conditional top event.

component node remains in the corresponding state after the failure. The failure node in state 1 does not always guarantee a particular failure state. It must therefore be assumed that the initial conditions are such that the component fails to the considered failure state. If the failure causes the component node to remain in its previous state, it is assumed that the initial state of the component node is the state that corresponds to the considered failure state. If the component always fails to the same failure state, no initial condition assumption is needed. If the component node can change state after the failure, the DRIF should only be calculated for state 1 of the failure node because the component does not necessarily remain in the same failure state for the whole scenario.

Component node C in Table 1 is stuck in its previous state when failure node F turns to state 1. Hence, in the calculation of $I^{DI}(C=0)$ with the initial time -3, the condition is that $F(-t)=1$ for all $-t\in\{0,-1,-2\}$ and $C(-3)=0$ and similarly, in the calculation of $I^{DI}(C=1)$, the condition is that $F(-t)=1$ for all $-t\in\{0,-1,-2\}$ and $C(-3)=1$.

There are also cases in which the failure state at time step $-n+1$ is determined in a more complicated way and more complex initial state assumptions are needed. Generally, the initial conditions that are needed to produce the failure state need to be identified. The result is a set of initial state combinations. The DRIF is calculated separately with each initial state combination assumption and the final DRIF for the failure state is calculated as a weighted average of

the DRIFs of the initial state combinations that lead to the failure state. These cases are not considered further in this paper because more simple cases are more common.

Other than for failure states of components, there is no easy and efficient way to calculate the DRIF for states of deterministic nodes because only initial states of deterministic nodes appear in prime implicants and there is therefore no easy way to derive the prime implicants for the conditional top event from the original prime implicants. The conditions that would be needed to produce the condition that the considered deterministic node would be in same state at all time steps should be solved. After that, the DRIF should be calculated assuming each of those conditions separately, and finally, a weighted average should be calculated. This could not be efficiently done with complex models because the number of different conditions would be large in many cases. However, a risk increase factor can be calculated for initial states of deterministic nodes by only assuming that a node is in a particular state at the initial time.

Table 2
The prime implicants of the top event $\{T(-1)=1, T(0)=1\}$ of the system presented in Fig. 3.

No.	Prime implicant
1	$\{C(-3)=0, F(-2)=1\}$
2	$\{C(-3)=0, F(-1)=1, T(-3)=-1, R(-3)=0\}$
3	$\{C(-3)=0, F(-1)=1, T(-3)=-1, N(-3)=0\}$
4	$\{C(-3)=0, F(-1)=1, T(-3)=-1, N(-3)=-1\}$
5	$\{C(-3)=0, T(-3)=-1, R(-3)=0, R(-2)=1\}$
6	$\{C(-3)=0, T(-3)=-1, N(-3)=0, R(-2)=1\}$
7	$\{C(-3)=0, T(-3)=-1, N(-3)=-1, R(-2)=1\}$
8	$\{C(-3)=0, F(-1)=1, N(-3)=0, R(-3)=1\}$
9	$\{C(-3)=0, F(-1)=1, N(-3)=-1, R(-3)=1\}$
10	$\{N(-3)=-1, R(-3)=1, F(-2)=0, T(-3)=1, F(-1)=1\}$
11	$\{C(-3)=0, N(-3)=0, R(-3)=1, R(-2)=1\}$
12	$\{C(-3)=0, N(-3)=-1, R(-3)=1, R(-2)=1\}$
13	$\{N(-3)=-1, R(-3)=1, F(-2)=0, T(-3)=1, R(-2)=1\}$

Table 3
The non-minimised implicants of the conditional top event.

No.	Prime implicant
1	$\{C(-3)=0\}$
2	$\{C(-3)=0, T(-3)=-1, R(-3)=0\}$
3	$\{C(-3)=0, T(-3)=-1, N(-3)=0\}$
4	$\{C(-3)=0, T(-3)=-1, N(-3)=-1\}$
5	$\{C(-3)=0, T(-3)=-1, R(-3)=0, R(-2)=1\}$
6	$\{C(-3)=0, T(-3)=-1, N(-3)=0, R(-2)=1\}$
7	$\{C(-3)=0, T(-3)=-1, N(-3)=-1, R(-2)=1\}$
8	$\{C(-3)=0, N(-3)=0, R(-3)=1\}$
9	$\{C(-3)=0, N(-3)=-1, R(-3)=1\}$
10	$\{C(-3)=0, N(-3)=0, R(-3)=1, R(-2)=1\}$
11	$\{C(-3)=0, N(-3)=-1, R(-3)=1, R(-2)=1\}$

6. Importance analysis of an emergency core cooling system

6.1. Emergency core cooling system

In this section, an emergency core cooling system of a boiling water reactor [12] is analysed with the DFM tool YADRAT. The system is shown in Fig. 6. The purpose of this system is to provide adequate water cooling of a reactor core if the ordinary cooling system is not functioning. An on–off control system regulates the water level in the pressure vessel by controlling a pump and a regulation valve. Sensors measure the water level and the pressure which are utilised in controlling of the valve, while only the water level measurement affects controlling of the pump. The water level can decrease due to evaporation. If the water level is low, more water is pumped into the pressure vessel until an upper limit is reached. The regulation valve is opened if both water level and pressure are measured to be under lower limit.

6.2. System reliability model

Fig. 7 presents the node structure of a DFM model from the YADRAT tool based on the emergency core cooling system [12]. This model contains one pump line that includes four components: a water level sensor modelled with component node WLM and failure node WLM-fail, a pressure sensor modelled with component node PM and failure node PM-fail, a regulation valve modelled with component node V and failure node V-fail and a pump modelled with component node P and failure node P-fail. Component node P has two states: 'on' and 'off' but the pump can

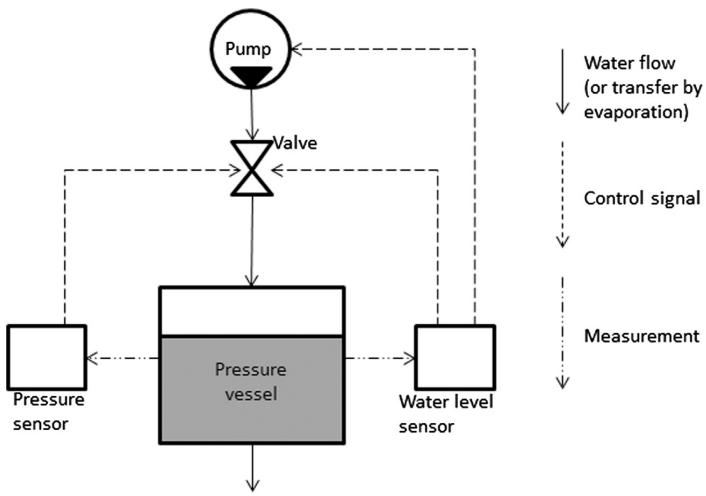

Fig. 6. An emergency core cooling system of a boiling water reactor.

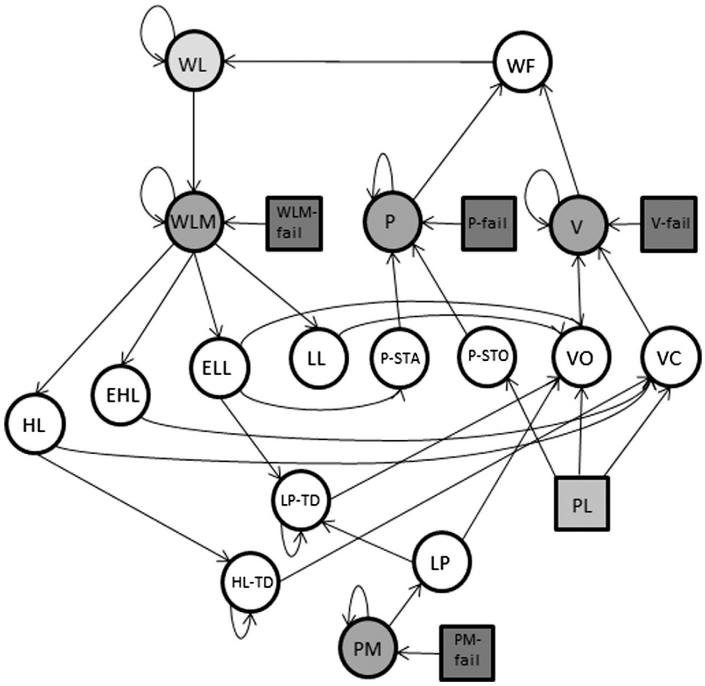

Fig. 7. A DFM model based on the emergency core cooling system.

Table 4
The failure rates of the components.

Failure node	Failure rate $[1/\Delta t]$
P-fail	0.01
V-fail	0.02
WLM-fail	0.03
PM-fail	0.04

only fail to failure state 'failed-off', which means that it does not pump any water. The valve can be failed in state 'failed-close' or 'failed-open'. The water level measurement can be frozen in state 'failed-low', 'failed-medium' or 'failed-high', while the pressure measurement can be frozen in state 'failed-low' or 'failed-high'.

An exponential model is used for failure probabilities. The failure rates are presented in Table 4. A failure rate is here the probability that the component fails during one time step.

The pump line also contains a random node PL that represents a pump leakage signal and several deterministic nodes that represent the signals between the sensors, the control logic and the actuators. The model also includes two deterministic nodes to represent the water inflow (WF) and the reactor water level (WL). The nodes of the model are described more in Appendix B.

6.3. Results

The analysed case was that the water level is low four time steps in a row (top event $\{WL(-3)=low, WL(-2)=low, WL(-1)=low, WL(0)=low\}$) which is a long enough time to be critical with regard to cooling of the core. The initial time was chosen to be −5 because earlier experiences had shown that all the relevant prime implicants can be identified using this time frame and same patterns are only repeated in prime implicants using a longer time frame.

Table 5
Examples of prime implicants.

No.	tot-pr.	pr.	Node	t	State
1	3.5E−2	0.330	WL	−5	High
		0.500	V	−5	Close
		0.500	PL	−5	True
		0.500	PM	−5	Low
		0.885	PM-fail	−2	0
		0.970	WLM-fail	−4	0
2	1.0E−2	0.500	V	−5	Close
		0.020	V-fail	−4	1
3	1.0E−2	0.500	V	−5	Close
		0.500	PM	−5	High
		0.040	PM-fail	−4	1
4	9.9E−3	0.500	V	−5	Close
		0.500	PL	−5	True
		0.040	V-fail	−3	1
5	1.4E−5	0.330	WL	−5	Low
		0.500	P	−5	On
		0.500	PL	−5	True
		0.100	PL	−4	True
		0.059	V-fail	−2	1
		0.941	WLM-fail	−3	0
		0.030	WLM-fail	−2	1

The number of identified prime implicants was 338 and five of them are presented in Table 5. These five prime implicants were chosen because they represent different prime implicant types. They are not necessarily the most important prime implicants. In prime implicants 2 and 4, the regulation valve fails in state 'failed-close'. In prime implicant 3, the pressure measurement is frozen in state 'failed-high'. In prime implicant 5, the failure states of the

water level measurement and valve are not unambiguous but can be different in different scenarios. The water level measurement is frozen in state 'failed-low' with a probability of 0.5 and in state 'failed-medium' with a probability of 0.5, and the valve is failed in state 'failed-open' with a probability of 0.17 and in state 'failed-close' with a probability of 0.83. Prime implicant 1 includes only an initial condition of the water level WL, initial conditions of components, a pump leakage signal at the initial time, a condition that the pressure measurement is functioning at time step −2 and a condition that the water level measurement is functioning at time step −4.

Table 6 presents both accurate and approximated DFV values for failure states of components. Approximated results were calculated using a single backtracking for each failure in a prime implicant in the solving of failure state probabilities and accurate results were calculated by dividing the failure state probability solving process into different scenarios as described in Section 4.6.

Fig. 8 presents the DFV values for component failures and pump leakage signal and Fig. 9 presents the DFV values for 0-states of failure nodes in the form of a graph. Table 7 presents the DRIF values for failure states of components, Table 8 for component failures and Table 9 for the states of the random node.

The valve clearly has more significant effects to the system's reliability than other components. It has the largest DFV values except for time step −4 (Fig. 8) and also according to the dynamic risk increase factor, its failure in state 'failed-close' has the worst effect on the system (Table 7). However, the valve's failure in state 'failed-open' decreases the top event probability significantly and the failure of the valve therefore has the smallest DRIF value of the component failures (Table 8). According to both importance measures, the pump is more important than the sensors even though its failure rate is smaller. This is logical because the pump

Table 6
The dynamic Fussell-Vesely values for failure states. Those DFV values that are 0 are left out.

Component node	Failure state	Time	DFV accu.	DFV appr.
P	Failed-off	0	0.198	0.198
	Failed-off	−1	0.198	0.198
	Failed-off	−2	0.184	0.184
	Failed-off	−3	0.151	0.151
	Failed-off	−4	0.089	0.089
V	Failed-open	0	0.00004	0.0003
	Failed-open	−1	0.00004	0.0003
	Failed-open	−2	0.00003	0.0001
	Failed-open	−3	0.00003	0.00003
	Failed-open	−4	0.00003	0.00003
	Failed-close	0	0.289	0.289
	Failed-close	−1	0.289	0.289
	Failed-close	−2	0.262	0.262
	Failed-close	−3	0.197	0.197
	Failed-close	−4	0.087	0.087
WLM	Failed-high	0	0.060	0.062
	Failed-high	−1	0.060	0.062
	Failed-high	−2	0.060	0.062
	Failed-high	−3	0.060	0.060
	Failed-high	−4	0.036	0.036
	Failed-medium	0	0.083	0.081
	Failed-medium	−1	0.083	0.081
	Failed-medium	−2	0.083	0.081
	Failed-medium	−3	0.059	0.059
	Failed-medium	−4	0.034	0.034
	Failed-low	0	0.00004	0.00004
	Failed-low	−1	0.00004	0.00004
	Failed-low	−2	0.00004	0.00004
PM	Failed-high	0	0.119	0.119
	Failed-high	−1	0.119	0.119
	Failed-high	−2	0.119	0.119
	Failed-high	−3	0.084	0.084
	Failed-high	−4	0.042	0.042

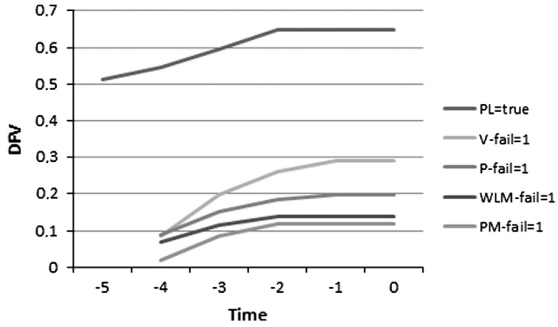

Fig. 8. The dynamic Fussell-Vesely values for component failures and pump leakage signal. On the right side of the graph, curves are ordered according to their end points.

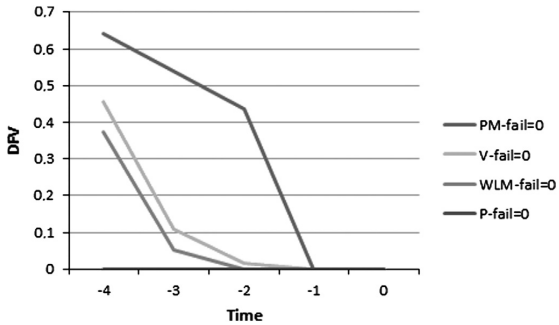

Fig. 9. The dynamic Fussell-Vesely values for 0-states of failure nodes. On the right side of the graph, curves are ordered according to their starting points.

Table 7
The dynamic risk increase factor values for failure states.

Component node	Failure state	DRIF
P	Failed-off	1.51
V	Failed-open	0.17
	Failed-close	1.81
WLM	Failed-high	1.50
	Failed-medium	1.44
	Failed-low	0.30
PM	Failed-high	1.42
	Failed-low	0.46

Table 8
The dynamic risk increase factor values for component failures (1-states of failure nodes).

Failure node	DRIF
P-fail	1.51
V-fail	0.99
WLM-fail	1.20
PM-fail	1.10

Table 9
The dynamic risk increase factor values for states of the random node.

Random node	State	DRIF
PL	true	1.51
	false	0.55

and the valve directly affect the water flow. However, the top event is most often caused by the pump leakage signal (PL in the 'true' state).

When analysing the time-dependent behaviour of DFV values in Table 6 and Fig. 8, it has to be remembered that DFV values are calculated cumulatively. The DFV values indicate that early failures of time steps −4 and −3 contribute most to the top event in all cases. This is logical because they have a more long lasting effect on the water level than later failures. The pump is the easiest component to analyse here because it always fails in the same way and its failure always has the same impact. The earlier the pump fails the more likely it is causing the top event as can be seen from the DFV values in Table 6. The DFV values of PM and WLM are also consistent with the intuitive idea that the earlier failure is more likely to cause the top event than a later failure.

The valve however has the largest addition to the cumulative DFV at time step −3. The reason for this is that component node V is more likely to be in state 'close' at time step −4 than at the initial time −5, and hence, the valve is more likely to be stuck in the harmful failure state 'failed-close' at time step −3 than at time step −4. High pump leakage signal probability 0.5 at the initial time is the main reason why component node V is more likely to be in state 'close' at time step −4. If component node V is in state 'open', the valve is closed due to the pump leakage signal unless it is fails.

For the pump and valve, failures of time step −1 can still contribute to the top event but failures of time step 0 cannot. This is because failures of the valve and the pump affect the water level with a delay of one time step. Measurement failures have to occur at time step −2 at the latest to contribute to the top event because they affect the water level with a delay of two time steps.

All possible failure states except the 'failed-low' state of the pressure measurement appear in the prime implicants. State 'low' of WLM, together with state 'low' of PM, sends the valve a signal that it needs to open so that more water can be pumped into the pressure vessel. Hence, failure state 'failed-low' cannot contribute to decreasing the water level but only increasing it. Failure state 'failed-low' of the water level measurement had small dynamic Fussell-Vesely values even though it cannot really cause the water level to decrease. In some of those prime implicants that had the water level measurement failed, such as in prime implicant 5 in Table 5, the situation was that either the sensor failed in state 'failed-medium' and partly caused the top event, or the sensor failed in state 'failed-low' and some other conditions caused the top event. None of the prime implicants implied that the water level measurement was frozen in state 'failed-low' with certainty. Hence, failure state 'failed-low' did appear in some prime implicants but did not really contribute to the top event. In those cases, the failure of the water level measurement appeared in the prime implicants because in some scenarios, the failure in the 'failed-medium' state was needed to cause the top event. Failure state 'failed-open' of the valve had small DFV values for the same reason.

It seems worth highlighting that failure state 'failed-medium' of the water level measurement had larger overall DFV value (DFV value of time step 0) than failure state 'failed-high'. This was because the failure in state 'failed-medium' at time step −2 can contribute to the top event but the failure in state 'failed-high' at time step −2 cannot. If the water level measurement is frozen in state 'failed-high' at time step −2, it means that the water level must have been 'high' at time step −4 and thus, the water level could not have been 'low' at time step −3.

Some differences appeared between accurate and approximated failure state DFV values because failure states were not calculated correctly using a single backtracking in every case. For example, approximated results imply that the failure in state 'failed-high' of the water level measurement at time step −2 contributes a little to the top event even though it is impossible as explained in the previous paragraph. The reason for this was that node WLM-fail at time step −4 appeared in two different branches of backtracking. The positive probability for failure state 'failed-high' came from an impossible scenario in which WLM-fail was in state 1 in one branch and 0 in another. Wrong failure state probabilities are always a result of taking this kind of impossible scenarios into account. These impossible scenarios appear in the calculations more likely if there are more time steps to backtrack. In other words, wrong results are more likely with late failures than with earlier failures because the backtracking process is longer and nodes are analysed at more time steps. In this example, wrong failure state probabilities were calculated only for failure times-2 and -1 as can be seen from the DFV values in Table 6.

As the wrong failure state probabilities are more likely to appear with late failures, those prime implicants for which wrong failure state probabilities are calculated are likely to have small probabilities and small effect to DFV values. Because of this, approximated DFV values are most often very close to accurate values. Also, even if impossible scenarios appear in backtracking, they do not necessarily lead to wrong failure state probabilities. In this example, the most important prime implicant for which wrong probabilities were calculated formed the portion of 0.89% of the top event probability, and as total, such prime implicants formed the portion below 2.5% of the top event probability. The only real issue that wrong failure state probabilities caused to complicate the analysis of results was the implication that the failure in state 'failed-high' of the water level measurement at time step −2 could contribute to the top event and even that is not very significant with regard to final conclusions drawn about the system.

The system is not coherent with regard to failure events, except the failure of the pump. The results of Fig. 9 indicate that some prime implicants include failure nodes in state 0. State 0 of PM-fail, in particular, contributes significantly to the top event. When this condition appears in prime implicants, it ensures that the pressure measurement cannot fail in state 'failed-low', which would prevent the top event from occurring. Hence, according to the DFV values, failure state 'failed-low' would prevent the top event from occurring in many cases.

From the dynamic risk increase factor values (Tables 7 and 8), it can be seen that some failure states increase the top event probability and some decrease it. Failure state 'failed-open' of the valve, failure state 'failed-low' of the water level measurement and failure state 'failed-low' of the pressure measurement decrease the top event probability because they can only cause the water level to increase not decrease. Those failure states that had significant DFV values in Table 6 can cause the water level to decrease and hence, they have DRIF values larger than 1. The failure of the regulation valve in state 'failed-close' at time step −4 causes the top event to occur with certainty.

The pump leakage signal has the highest DFV values (0.648 when all time steps are taken into account) and same or higher DRIF value (1.51) than any of the component failures. The best way to improve the reliability of this system would therefore be to reduce the probability of the pump leakage signal which could be done by reducing the probability of the spurious signal and the probability of the pump leakage. It would also be beneficial to improve the reliability of the valve and the pump. Results also indicate that the top event probability would lower if the control system was changed so that the valve was more open and the measurement sensors displayed low values for the water level and the pressure all the time. However, this type of change is neither practically possible nor sensible and it might cause the water level to be high all the time which could be harmful too. Because of this, it would be worthwhile to analyse the system with the top

event $\{WL(-3)=high, WL(-2)=high, WL(-1)=high, WL(0)=high\}$ as well. With this top event, the results would be quite different.

7. Discussion

7.1. Benefits

The dynamic risk importance measures are an important contribution to the dynamic flowgraph modelling because the previously developed importance measures [14,15] were not designed to measure significances of node's states properly while the states of nodes often play an important role in the interpretation of DFM results. The dynamic risk importance measures take the time aspect of DFM into account in a logical way that supports the interpretation of results. In addition, they can provide detailed information on how components modelled with two nodes contribute to the top event. In principle, they could also be applied in all dynamic methods that rely on variables with a finite number of states. The time aspect of the dynamic risk importance measures could also be generalised to the case of continuous time.

7.2. The dynamic Fussell-Vesely

The Fussell-Vesely measure is the importance measure used most often because it is simple to compute and it encapsulates purely the information from minimal cut sets or prime implicants. The dynamic Fussell-Vesely has the same qualities. The dynamic Fussell-Vesely takes into account both the probability that the node is in the considered state and how the node and the state interconnect with other nodes. However, the DFV does not take the incoherency of a system into account because it does not consider that the prime implicants can include the node in different states. This limits the interpretation of the DFV values calculated for component failures. The incoherency can only be taken into account by calculating separate DFV values for the state 0 of a failure node. For example, the results of Fig. 8 could not indicate that the system was incoherent with regard to some failure nodes and the incoherency only came evident when the results of Fig. 9 were analysed.

If the system is coherent with regard to a failure, other risk importance measures that depend on the conditional top event probability with a condition that the considered component is functioning can also be derived from Fussell-Vesely. The same cannot be done in the incoherent case. The fractional contribution [25] gives same results as Fussell-Vesely in coherent case and takes the incoherency into account in incoherent case. But, the dynamic fractional contribution would be computationally more demanding as new prime implicants of the conditional top event should be identified.

7.3. The failure state approach

The failure state approach used in the calculation of the dynamic risk importance measures for components provides information about the state of a failed component as discussed in Section 4.2. This information cannot directly be read from prime implicants. The failure state approach is useful because a failure state (or failure mode) is really an important factor when analysing causes of a top event even if only the failure is the fundamental cause from the mathematical point of view. There are also other ways to model different failure modes in DFM. One way is to use a so-called “multi-state failure node” that contains separate states for different failure modes. If a “multi-state failure node” is used instead of modelling presented in this paper, there is no need to solve failure states. This requires own failure node state for each failure state, and hence, the decision table of the component grows large, which makes the DFM model computationally more demanding.

The solving of accurate failure state probabilities is based on an examination of different scenarios related to nodes that appear in different branches of backtracking as discussed in Section 4.6. This can be very demanding for components of complex systems because the number of different scenarios can be large. In some cases, it is better to compute approximations by a single backtracking than to examine all the different scenarios. In the example case presented in this paper, accurate failure state probabilities, which affected the DFV results in Table 6, were calculated in 2 s. The same model was also analysed with −6 as the initial time instead of −5 (results not presented here) and in that case, the computation lasted 2 min while approximated results were obtained in a second. Thus, one time step more to backtrack makes a significant addition to computation demands. But, the effect that inaccurate failure state probabilities have on DFV values was small in all the examined example cases such as in the results presented in Table 6. Hence, when only approximations of failure state DFV values are needed, the use of a single backtracking may be sufficient.

7.4. The dynamic risk increase factor and other importance measures

Like the traditional risk increase factor, the dynamic risk increase factor mainly depends on how other components can keep the system operating while the considered component is failed or node is in a given state. When used independently, the DRIF gives a fairly restricted view on how a state of a node contributes to the top event but it is a good complement to the DFV. The DRIF can be used to derive some other dynamic risk importance measures that rely on the conditional top event probability with an assumption that a node is in a given state at all time steps.

There are many other importance measures that could also be generalised in dynamic and multi-state cases. Other often-used importance measures include Birnbaum importance, the risk decrease factor (also known as the risk reduction worth), the criticality importance and the partial derivative [1]. For some risk importance measures, there is more than one way in which the generalisation to the dynamic case can be made. In the dynamic risk increase factor, it is assumed that the condition starts at the first possible time step. The DRIF could also be generalised for other time steps. A similar idea has been considered in relation to multi-phase missions [23]. This could bring worthwhile additional information in some cases when the system is incoherent with regard to the considered node but mostly not. The computation of the DRIF for a failure state with an assumption of a late failure would be significantly more demanding than the computation of the DRIF with the failure at the first possible time step, because not only the initial conditions would affect the failure state but also the states of the random nodes and failure nodes at earlier time steps than the considered failure time. All the different state combinations of affecting nodes at relevant time steps should be considered when the assumption of a failure state is made. Computation demands would, of course, be much smaller if failure states were not considered.

Many risk importance measures rely on the calculation of a conditional top event probability. In those importance measures that include a failure assumption, the failure condition can be replaced with an assumption that a node is in a particular state at particular time steps. An assumption that a component is functioning can also be replaced with an assumption that a node is not in a particular state at particular time steps. In these cases, a new set of prime implicants is identified for the conditional top event. If accurate results are to be produced, the identification of new

prime implicants is computationally demanding when the original set of prime implicants is large.

The computation of differential risk importance measures, such as the partial derivative and the differential importance measure [26], would be easier in DFM because prime implicants of the conditional top event would be the same as the original prime implicants. Only manipulation of probability parameters and recalculation of probabilities would be required in the calculation of the conditional top event probability. It would however be difficult to apply differential risk importance measures to failure states unless a separate probability is assigned to each failure state.

7.5. Measuring the importance of a node

This paper focused on importance measures that measure the importance of a state of a node. Sometimes, analysts are more interested in the overall importances of nodes so that the most critical components of a system can be identified. To measure the overall importances of nodes, the composite importance measure approach presented in [21] could be applied to the dynamic risk increase factor. The Fussell-Vesely presented in [21] differs significantly from the dynamic Fussell-Vesely as it relies on the computation of the top event probability with an assumption that the node is in a certain state. Instead, the Fussell-Vesely from [14] could be generalised to take the time aspect into account. In the risk increase factor presented in [14], it is assumed that the node is in its worst state at each time step. The worst state can be different at different time steps. In the dynamic risk increase factor, it is assumed that a node is in the same state at each time step, which makes these measures fundamentally different. However, the worst state approach could be applied to the dynamic risk increase factor to formulate an overall DRIF measure for a node. If the worst state was the same at every time step, these measures would give the same value, but if the worst state differed, the risk increase factor from [14] would give a larger value.

7.6. DFM tool development

Other YADRAT tool related research includes modelling of common cause failures and other dependencies between failures as well as a study of different component reliability models. Group importance measures are investigated in relation to dependent failures. Dynamic risk importance measures, for example, can be formulated separately for each failure state combination of a common cause failure group. An interesting question is how the time aspect of DFM is considered in group importance measures. In addition to failure nodes and random nodes, different stochastic node types will be developed when some component reliability models are implemented in YADRAT. The computation of dynamic risk importance measures has to be studied in relation to different dynamic constraints of stochastic nodes.

The main challenge in DFM tool development is to provide trustworthy results in a reasonable calculation time. In the dynamic risk importance measure calculation, this means that it is usually better to compute approximations rather than to try to aim for accurate values. It is also important in the development of dynamic risk importance measures in DFM to consider what information is actually useful, as the main objective of risk importance measures is to provide guidance for the system's design. The information given by importance measures needs to be kept simple enough so that the analysts can interpret it.

8. Conclusion

The dynamic risk importance measures use all the information that is available in prime implicants of DFM to measure significances of node's states unlike any other importance measure. With dynamic risk importance measures calculated for different failure states of components and states of failure nodes, the component's influence on the system's reliability can be analysed more comprehensively than with just risk importance measures calculated for failure events. As the dynamic Fussell-Vesely is calculated for time steps, it is also possible to judge at which points of the time line certain failures and conditions need to occur to contribute to the top event.

Acknowledgments

The author would like to thank Kim Björkman, Jan-Erik Holmberg, Ahti Salo, Antti Toppila and other participants of the seminar on scientific publishing for their help, not to forget Ilkka Männistö.

Appendix A. Backtracking algorithm example

This section presents how the backtracking algorithm presented in Figs. 1 and 2 progresses step by step in the backtracking example of Section 4.5. Some parts of the process are cut out to keep the length of the example moderate. The state probabilities of node A at time step $-u$ are represented by a vector $SP(A(-u)) = (a_{u,0}, a_{u,1}, \ldots, a_{u,b})$, where b is the number of states of A and $a_{u,s}$ is the probability of state s. All calculated state probabilities are always stored. The backtracking process starts from the component node at the failure time:

1. Component node C at $-t = -1$.
2. The input nodes are F, N and C.

2.1. Examine F.
- 2.1.1. The time lag $l = 0$.
- 2.1.2. The node type is failure node.
- 2.1.3. $-t = -1 \neq -3$.
- 2.1.4. The prime implicant does not contain a literal of the form $F(-u) = 0, -u \geq -1$.
- 2.1.5. The prime implicant contains literal $F(-1) = 1$.
- 2.1.6. The probability for state 1 is 1.
- 2.1.7. $SP(F(-1)) = (0, 1)$.

2.2. Examine N.
- 2.2.1. The time lag $l = 0$.
- 2.2.2. The node type is deterministic node.
- 2.2.3. $-t-l = -1 > -3$.
- 2.2.4. $-t = -t-l = -1$.
- 2.2.5. Deterministic node N at $-t = -1$.
- 2.2.6. The input nodes are R, T and N
  - 2.2.6.1. Examine R.
    - 2.2.6.1.1. The time lag $l = 1$.
    - 2.2.6.1.2. The node type is random node.
    - 2.2.6.1.3. The prime implicant does not contain a literal representing R at $-t-l = -2$.
    - 2.2.6.1.4. State probabilities $r_{2,0}$ and $r_{2,1}$ are obtained from the probability parameters.
    - 2.2.6.1.5. $SP(R(-2)) = (r_{2,0}, r_{2,1})$.
  - 2.2.6.2. Examine T.
    - 2.2.6.2.1. The time lag $l = 0$.

⋮

2.2.6.2.x. $SP(T(-1)) = (t_{1,0}, t_{1,1}, t_{1,2})$.

⋮

2.2.6.3. Examine N.

2.2.6.3.1. The time lag $l=1$.

⋮

2.2.6.3.x. $SP(N(-2)) = (n_{2,-1}, n_{2,0}, n_{2,1})$.

⋮

⋮

2.2.x. $SP(N(-1)) = (n_{1,-1}, n_{1,0}, n_{1,1})$.

⋮

2.3. Examine C.

2.3.1. The time lag $l=1$.

2.3.2. The node type is deterministic node.

2.3.3. $-t-l=-2 \geq -3$.

2.3.4. $-t=-t-l=-2$.

2.3.5. Deterministic node N at $-t=-2$.

2.3.6. The input nodes are F, N and C.

⋮

2.3.8. $SP(C(-2)) = (c_{2,0}, c_{2,1}) = (n_{2,-1} + n_{2,0}c_{3,0}, n_{2,1} + n_{2,0}c_{3,1})$ (calculated in Section 4.5).

2.3.9. This node is component node C but $-t=-2$ is not the failure time −1.

2.3.10. The time lag $l=1$.

2.3.11. $-t=-t+l=-1$.

3. State probabilities are calculated by summing the probabilities of the rows in Table A1:

$$c_{1,0} = n_{1,-1}c_{2,0} + n_{1,0}c_{2,0} + n_{1,1}c_{2,0} = c_{2,0} = n_{2,-1} + n_{2,0}c_{3,0} \quad (A.1)$$

and

$$c_{1,1} = n_{1,-1}c_{2,1} + n_{1,0}c_{2,1} + n_{1,1}c_{2,1} = c_{2,1} = n_{2,1} + n_{2,0}c_{3,1}. \quad (A.2)$$

4. $SP(C(-1)) = (n_{2,-1} + n_{2,0}c_{3,0}, n_{2,1} + n_{2,0}c_{3,1})$.
5. This node is C and $-t=-1$.
6. Failure state probabilities are $n_{2,-1} + n_{2,0}c_{3,0}$ and $n_{2,1} + n_{2,0}c_{3,1}$.

Table A1
Probabilities of the rows in the decision table of C.

Node	Output	Inputs			Prob
	C	F	N	C	
Time lag		0	0	1	
	0	0	−1	0	$0 \cdot n_{1,-1} \cdot c_{2,0} = 0$
	0	0	−1	1	$0 \cdot n_{1,-1} \cdot c_{2,1} = 0$
	0	0	0	0	$0 \cdot n_{1,0} \cdot c_{2,0} = 0$
	1	0	0	1	$0 \cdot n_{1,0} \cdot c_{2,1} = 0$
	1	0	1	0	$0 \cdot n_{1,1} \cdot c_{2,0} = 0$
	1	0	1	1	$0 \cdot n_{1,1} \cdot c_{2,1} = 0$
	0	1	−1	0	$1 \cdot n_{1,-1} \cdot c_{2,0} = n_{1,-1}c_{2,0}$
	1	1	−1	1	$1 \cdot n_{1,-1} \cdot c_{2,1} = n_{1,-1}c_{2,1}$
	0	1	0	0	$1 \cdot n_{1,0} \cdot c_{2,0} = n_{1,0}c_{2,0}$
	1	1	0	1	$1 \cdot n_{1,0} \cdot c_{2,1} = n_{1,0}c_{2,1}$
	0	1	1	0	$1 \cdot n_{1,1} \cdot c_{2,0} = n_{1,1}c_{2,0}$
	1	1	1	1	$1 \cdot n_{1,1} \cdot c_{2,1} = n_{1,1}c_{2,1}$

Table B1
The nodes of the emergency core cooling system model (Fig. 8).

Node	Description
EHL	Is true if the water level measurement is high. The node is used in controlling the valve.
ELL	Is true if the water level measurement is low. The node is needed for starting the pump and opening the valve.
HL	Is true if the water level measurement is high. The valve can open only if this node is true.
HL-TD	Time delay condition for the closing of the valve.
LL	Is true if the water level measurement is low. The node is used in controlling the valve.
LP	Is true if the pressure measurement is low. The node is needed for opening the valve.
LP-TD	Time delay condition for the first opening of the valve.
P	Represents pump which can be on or off. The pump is controlled by start and stop commands.
PL	Is true if pump leakage signal is sent. If the signal is sent, the pump is commanded to stop and the valve commanded to close.
PM	Represents pressure measurement which can be low or high. It is assumed that the pressure is high/low every other time step.
P-STA	Start command is sent to the pump when this node is true.
P-STO	Stop command is sent to the pump when this node is true.
V	Represents valve which can be open or closed. The valve is controlled by open and close commands.
VC	Close command is sent to the valve when this node is true.
VO	Open command is sent to the valve when this node is true.
WF	Water flow is high if the pump line pumps water into the pressure vessel.
WL	Represents water level which can be low, medium or high. Level rises if water flow is high and lowers if water flow is low.
WLM	Water level measurement measures the water level from previous time step.

Appendix B. Emergency core cooling system model

Each node of the emergency core cooling system model presented in Fig. 8 is briefly described in Table B1.

References

[1] Van Der Borst M, Schoonakker H. An overview of PSA importance measures. Reliability Engineering and System Safety 2001;72:241–5.

[2] Zio E. Risk importance measures. In: Pham H, editor. Safety and risk modeling and its applications. London: Springer-Verlag; 2011. p. 151–95.

[3] Kuo W, Zhu X. Some recent advances on importance measures in reliability. IEEE Transa\hskip2pcctions on Reliability 2012;61:344–60.

[4] Kuo W, Zhu X. Relations and generalizations of importance measures in reliability. IEEE Transactions on Reliability 2012;61:659–74.

[5] Labeau PE, Smidts C, Swaminathan S. Dynamic reliability: towards an integrated platform for probabilistic risk assessment. Reliability Engineering and System Safety 2000;68:219–54.

[6] Cepin M. Fault tree analysis. In: Cepin M, editor. Assessment of power system reliability—methods and applications. London: Springer; 2011. p. 61–87.

[7] Rauzy A. Binary decision diagrams for reliability studies. In: Mistra KB, editor. Handbook of performability engineering. London: Springer; 2008. p. 381–96.

[8] Do Van P, Barros A, Bérenguer C. Reliability importance analysis of Markovian systems at steady state using perturbation analysis. Reliability Engineering and System Safety 2008;93:1605–15.

[9] Garrett CJ, Guarro SB, Apostolakis GE. The dynamic flowgraph methodology for assessing the dependability of embedded software systems. IEEE Transactions on Systems, Man and Cybernetics 1995;25:824–40.

[10] Al-Dabbagh AW, Lu L. Reliability modeling of networked control systems using dynamic flowgraph methodology. Reliability Engineering and System Safety 2010;95:1202–9.

[11] Al-Dabbagh AW, Lu L. Dynamic flowgraph modeling of process and control systems of a nuclear-based hydrogen production plant. International Journal of Hydrogen Energy 2010;35:9569–80.

[12] Björkman K. Solving dynamic flowgraph methodology models using binary decision diagrams. Reliability Engineering and System Safety 2013;111:206–16.

[13] Tyrväinen T, Björkman K. Modelling common cause failures and computing risk importance measures in the dynamic flowgraph methodology. In: Proceedings of the 11th international probabilistic safety assessment and management conference & the annual European safety and reliability conference, 25–29 June 2012, Helsinki, Finland. Helsinki: The International

Association for Probabilistic Safety Assessment and Management (IAPSAM); 2012. 30-Th4-1.
[14] Karanta I. Importance measures for the dynamic flowgraph methodology. Espoo (Finland): VTT Technical Research Centre of Finland, Systems Research.
[15] Houtermans MJM. A method for dynamic process hazard analysis and integrated process safety management. Doctoral thesis. Eindhoven (Netherlands): Technische Universiteit Eindhoven; 2001 May. ⟨http://alexandria.tue.nl/extra2/200111699.pdf⟩.
[16] Aldemir T, Guarro SB, Mandelli D, Kirschenbaum J, Mangan LA, Bucci P, et al. Probabilistic risk assessment modeling of digital instrumentation and control systems using two dynamic methodologies. Reliability Engineering and System Safety 2010;95:1011–39.
[17] Huang CY, Chang YR. An improved decomposition scheme for assessing the reliability of embedded systems by using dynamic fault trees. Reliability Engineering and System Safety 2007;92:1403–12.
[18] Do Van P, Barros A, Bérenguer C. From differential to difference importance measures for Markov reliability models. European Journal of Operational Research 2010;204:513–21.
[19] Ramirez-Marquez JE, Rocco CM, Gebre BA, Coit DW, Tortorella M. New insights on multi-state component criticality and importance. Reliability Engineering and System Safety 2006;91:894–904.
[20] Levitin G, Podofillini L, Zio E. Generalised importance measures for multi-state elements based on performance level restrictions. Reliability Engineering and System Safety 2003;82:287–98.
[21] Ramirez-Marquez JE, Coit DW. Composite importance measures for multi-state systems with multi-state components. IEEE Transactions on Reliability 2005;54:517–29.
[22] Burdick GR, Fussell JB, Rasmuson DM, Wilson JR. Phased mission analysis: a review of new developments and an application. IEEE Transactions on Reliability 1977;R-26:43–9.
[23] Vaurio JK. Importance measures for multi-phase missions. Reliability Engineering and System Safety 2011;96:230–5.
[24] Contini S, Cojazzi GGM, Renda G. On the use of non-coherent fault trees in safety and security studies. Reliability Engineering and System Safety 2008;93:1886–95.
[25] Mankamo T, Pörn K, Holmberg JE. Uses of risk importance measures. Technical report. Espoo (Finland): VTT Technical Research Centre of Finland; 1991. VTT Research notes 1245. ISBN 951-38-3877-3.
[26] Borgonovo E, Apostolakis GE. A new importance measure for risk-informed decision making. Reliability Engineering and System Safety 2001;72:193–212.

Publication III

Tyrväinen, T. Common cause failures in the dynamic flowgraph methodology. *Manuscript*, 19+17 pages, 2017.

Common cause failures in the dynamic flowgraph methodology

Tero Tyrväinen
VTT Technical Research Center of Finland Ltd, P.O. Box 1000, FI-02044 VTT, Finland
tero.tyrvainen@vtt.fi

Abstract

In this paper, common cause failures in dynamic reliability analysis are examined. Common cause failures are important in the risk analysis of complex systems including redundancies. Modeling of common cause failures has been well-studied in the context of static methods, such as fault trees; they cannot however represent the behavior of dynamic systems sufficiently. Dynamic flowgraph methodology is a method for analyzing dynamic systems containing feedback loops. Like fault tree analysis, dynamic flowgraph methodology seeks to identify root causes for system failure. In the dynamic flowgraph methodology, component failures occurring at different time points can together cause system failure. This paper extends the dynamic flowgraph methodology by presenting common cause failure models that take failure times of components into account. Common cause failures are added to the results in the post-processing phase of the analysis, which has been proven to be an efficient approach in the context of dynamic flowgraph methodology. An illustrative model of an emergency core cooling system is presented to demonstrate common cause failure modeling in dynamic flowgraph methodology.

1 Introduction

Common cause failures (CCFs) [1, 2] are important in the reliability analysis of complex systems including redundant components. If dependencies are not taken into account, the risk of system failure can be underestimated. In [3], a CCF is defined using the following criteria:

"1. Two or more individual components fail, are degraded (including failures during demand or in-service testing), or have deficiencies that would result in component failures if a demand signal had been received.

2. Components fail within a selected period of time such that success of the probabilistic risk assessment (PRA) mission would be uncertain.

3. Components fail because of a single shared cause and coupling mechanism.

4. Components fail within the established component boundary."

Dynamic flowgraph methodology (DFM) [4, 5, 6, 7, 8, 9, 10, 11] analyzes dynamic systems containing feedback loops. Like fault tree analysis, DFM aims to identify which conditions can cause the top event. DFM has been developed because traditional methods are able to represent the dynamic behavior of a system only to a limited extent. DFM can represent how the values of the system's variables evolve in time. It has most often been applied to digital control systems so that the interactions between the control system and the controlled process have been included in the model.

In a DFM model, variables can have multiple states, which is an advantage when modeling components with multiple failure modes, and the effects that failures have on process variables. In addition, only one DFM model is needed for the analysis of different states of the system. DFM has been highlighted in [12] as one of the promising dynamic reliability analysis approaches, along with Markov models [13, 14].

DFM analysis generates a set of prime implicants [15] as its main result. Prime implicants are minimal combinations of events and conditions that are sufficient to cause the top event. DFM considers at which time points events must occur in order to cause the top event. Compared to static fault tree analysis, DFM provides more accurate information about the development of accident scenarios, resulting in more accurate probability calculations.

Although CCFs are an important part of the theory of probabilistic risk analysis, to the author's knowledge there are no earlier publications concerning CCFs in a DFM context. The model in [16] included CCFs, but they were not really discussed in the paper. One special characteristic of DFM is that components can fail at different time points contributing to the same top event. Although a CCF event is often interpreted as a simultaneous failure of similar components, NUREG/CR-6268 [3] defines that components need to fail only during PRA mission time, which is typically 24 hours. This definition is used both in data collection and PRA analysis. In data collection, if multiple failures occur within 24 hours, they are interpreted as a CCF. In addition, 50% of such events where the time between failures is 24-48 hours are counted as CCFs, i.e. a timing factor of 0.5 is used [3].

In traditional fault tree analysis, it is not considered whether the components fail simultaneously or non-simultaneously during the mission time, but in DFM, non-simultaneous CCFs can be considered, because DFM divides the mission time into smaller time intervals. This paper takes the possibility of such CCFs into account and generalizes two parametric models [17, 18], β- and α-factor models, for DFM analysis. It appears that the selected approach leads to slightly higher CCF probabilities compared to the case in which CCFs are assumed to be simultaneous. This contribution can be viewed as an extension of a conference paper published in 2012 [19].

The structure of the paper is as follows. Section 2 presents relevant details of the DFM. Section 3 discusses CCF modeling techniques. In Section 4, new CCF models are presented, and Section 5 analyzes an example system. Section 6 discusses and concludes the article.

2 Dynamic flowgraph methodology

2.1 Models and analysis

A DFM model [4, 5, 6, 7, 8, 9, 10, 11] is a graph representation of the analyzed system. Nodes in the model represent the components and variables of the system, and edges connecting the nodes represent causal and other dependencies between the nodes. There can be time delays in those dependencies, and nodes can have two or more states. If a node does not depend on any other node, it is a stochastic node whose state is determined by a discrete probability distribution at each time step. The state of a deterministic node is determined based on the states of input nodes. Each deterministic node has a decision table which specifies the output state for each state combination of the input nodes.

Figure 1 shows a simple DFM model of a tank system with a valve that is controlled based on water level measurement, and Table 1 gives the decision table of node V as an example. There are also decision tables for nodes L and M, which are not presented here. In the model, node V represents the functional state of a valve (state 0 for closed and 1 for open), L represents water level and M represents water level measurement value. Nodes M and L have three states -1, 0 and 1 indicating water levels low, medium and high. Nodes S and F are stochastic nodes determining whether the water level measurement and the valve have failed. A row in the decision table specifies a combination of states of the input nodes, and the corresponding state of the output node. Delays in the dependencies are shown in the time lag row.

The primary target of DFM is to identify prime implicants of the analyzed top event. A prime implicant is a minimal combination of conditions that is sufficient to cause the top event [15]. In DFM, these conditions are represented by literals. In this context, a literal is a triplet consisting of a variable V, state s and time point $-t$, and denoted as $V_s(-t)$. The top event is also defined as a set of literals.

A DFM model is typically analyzed by tracing event sequences backwards from effects to causes. Deductive analysis starts from the top event and proceeds backwards in time until a defined initial time. The graph model is traced backwards in the cause-and-effect flow, and at the same time, a binary reliability model is produced for the top event, e.g. fault tree [4] or binary decision diagram [7]. Prime implicants are then solved from the binary reliability model. The prime implicants of a top event can contain initial states of deterministic nodes and states of stochastic nodes at any time step.

A failure node is a stochastic node with two states, a "functioning state" (0 or 'false') and a "failure state" (1 or 'true'). It cannot turn from state 1 to state 0, and it is fixed to state 0 at the initial time step.

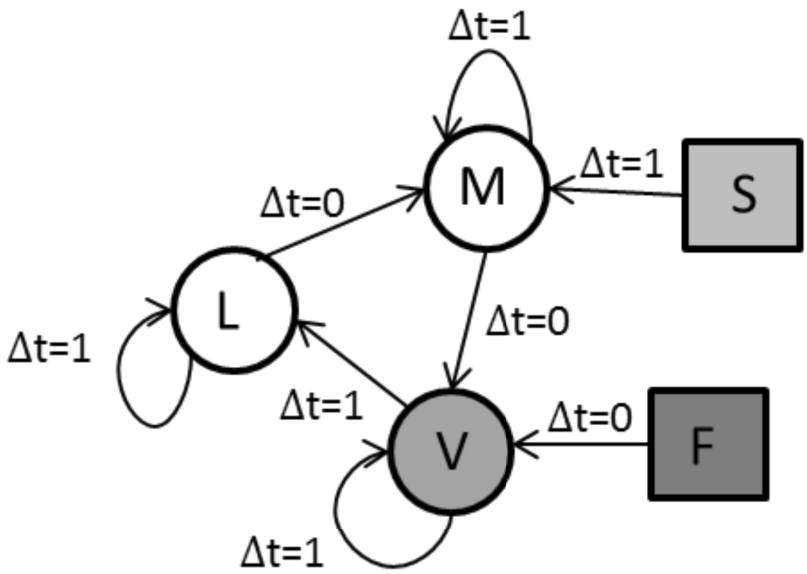

Figure 1: A DFM model with five nodes.

Table 1: The decision table of node V.

	Output	Inputs		
Node	V	F	M	V
Time lag		0	0	1
	0	0	−1	0
	0	1	−1	0
	0	0	0	0
	0	1	0	0
	1	0	1	0
	0	1	1	0
	0	0	−1	1
	1	1	−1	1
	1	0	0	1
	1	1	0	1
	1	0	1	1
	1	1	1	1

2.2 Failure probabilities

In parametric CCF models, CCF probabilities are calculated based on the probabilities of individual failures. Therefore, this section presents how individual failure event probabilities can be calculated in DFM.

In this paper, failure probabilities are presented using an exponential model with a constant failure rate, which is a commonly used model and suitable for demonstration purposes. More generally, the CCF probabilities are always calculated with the same formula (presented later in Section 4) regardless of how single failure probabilities are calculated. If a component fails during a time step with probability λ, and $-n$ is the initial time, the component is functioning at time step $-t$ ($> -n$) with probability $(1-\lambda)^{n-t}$ and is in failed state with probability $1-(1-\lambda)^{n-t}$.

The failure probability needs to be calculated for the situation in which the prime implicant includes the failure. If literals are assumed to be independent, the probability of prime implicant π is the product of the probabilities of the literals

$$P(\pi) = \prod_{x \in \pi} P(x).$$

A prime implicant can include two literals which represent the same failure node at different time steps. The literal with an earlier time step always has state 0 and the literal with a later time step has state 1. The dependency between these literals must be taken into account when the probability of the prime implicant is computed. The probability that two dependent conditions/events occur can be calculated using a conditional probability:

$$P(C \cap D) = P(C) \cdot P(D \mid C).$$

Let prime implicant π contain literals $F_0(-u)$ (failure node F in state 0 at time $-u$) and $F_1(-t)$ (failure node F in state 1 at time $-t$), where $-n < -u < -t$. In other words, the component needs to function at time $-u$ and then be failed at time step $-t$ in order that the top event may occur because of this prime implicant. The probability that both of these literals are true can be calculated as

$$\begin{aligned} &P(F_0(-u), F_1(-t)) \\ &\quad = P(F_0(-u)) \cdot P(F_1(-t) \mid F_0(-u)). \end{aligned}$$

The probability of the first literal is

$$P(F_0(-u)) = (1-\lambda)^{n-u}.$$

In the calculation of the probability of the literal $F_1(-t)$, it must be taken into account that the component is functioning at time step $-u$. Hence, the probability is $1-(1-\lambda)^{u-t}$

rather than $1-(1-\lambda)^{n-t}$. When the probability is calculated for a literal that represents a failure node in state 1, it must be checked whether the prime implicant includes the same failure node in state 0. Thus, the probability of literal $F_1(-t)$ can be calculated as a conditional probability $P(F_1(-t) \mid \pi \setminus \{F_1(-t)\})$, where π is the prime implicant and $\pi \setminus \{F_1(-t)\}$ is the prime implicant excluding literal $F_1(-t)$. To shorten the notation, conditional probability $P(F_1(-t) \mid \pi \setminus \{F_1(-t)\})$ is denoted as $P_\pi(F_1(-t))$. When π does not contain the failure node in state 0, it follows that

$$P_\pi(F_1(-t)) = P(F_1(-t)) = 1-(1-\lambda)^{n-t}.$$

In other words, literal $F_1(-t)$ is independent of the other literals in the prime implicant. If prime implicant π also includes another failure node G in state 1, the product of literal probabilities includes terms $P_\pi(F_1(-t))$ and $P_\pi(G_1(-v))$. Even though the notation appears to imply that the probabilities of these literals depend on one another, the failures are in fact mutually independent, and the probabilities depend on the separate literals only.

Let $\sigma = \{F_0(-u), F_1(-t), G_0(-h), G_1(-v)\}$ be a prime implicant. The probability of σ is calculated as

$$\begin{aligned} P(\sigma) =& P(F_0(-u)) \cdot P(F_1(-t) \mid F_0(-u)) \\ & \cdot P(G_0(-h)) \cdot P(G_1(-v) \mid G_0(-h)) \\ =& P(F_0(-u)) \cdot P_\sigma(F_1(-t)) \\ & \cdot P(G_0(-h)) \cdot P_\sigma(G_1(-v)). \end{aligned}$$

3 Modeling techniques for common cause failures

Markov/cell-to-cell-mapping is a dynamic reliability analysis method that has been compared to DFM [20]. Multi-state and time-dependent modeling is also possible with Markov models. In [21], CCFs were incorporated into a Markov model by adding CCF states and corresponding state transitions to the model. Similar state transitions could be modeled in DFM. However, this would probably make the models more complex and increase the computation time considerably.

In fault tree analysis [22], CCFs are typically modeled with basic events that are separate from individual component failure basic events, such as in Figure 2. Similarly, in DFM, individual failures and CCFs can be modeled using separate failure nodes. It appears that this technique was used in the DFM model in [16]. However, the addition of CCF nodes increases the complexity of the model, and the computation times of DFM are rather sensitive to the model's complexity.

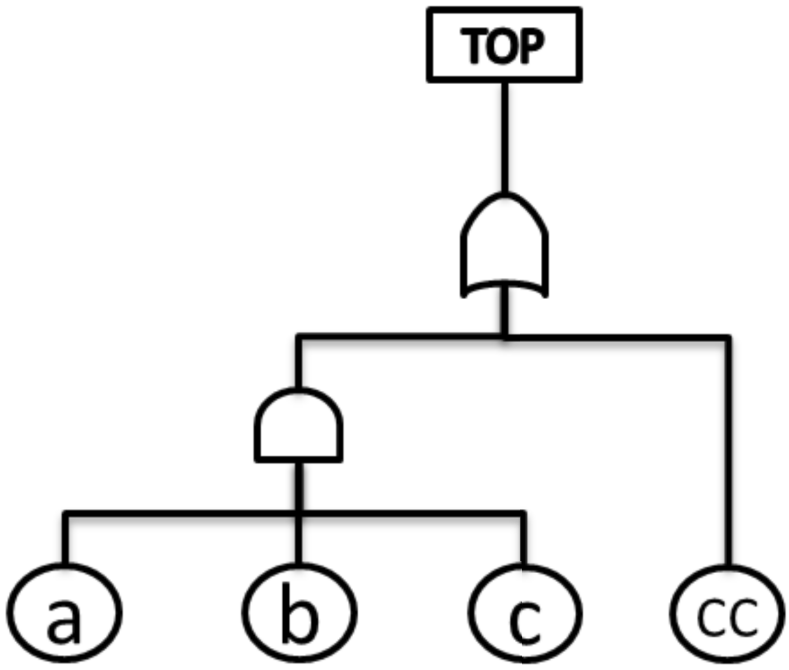

Figure 2: A fault tree containing basic events a, b and c, and a CCF between them (CC).

In [23], CCFs were incorporated into dynamic fault trees with an approach in which the system's fault tree model was reduced and analyzed separately with each possible CCF scenario. It is possible to use this type of technique in DFM. This would simplify DFM analysis, which would be advantageous, but the model should be solved many times and the prime implicants should be combined and re-minimized in the end. It is likely that the model would not be simplified enough in the separate analysis of different CCF scenarios to compensate for the drawback that the model should be solved many times.

One way to incorporate CCFs is to analyze the system model first without CCFs and then to add the effects of CCFs to the results. This approach has been used in the GO-FLOW methodology [24]. It does not add to the complexity of the DFM model. This is an advantage because computation times are sensitive to the model's complexity in DFM. Hence, this idea is used in this paper.

4 Common cause failures in DFM

4.1 Traditional common cause failure models

Consider a group of m identical components with a common failure mechanism. When CCFs are modeled using the β-factor model, it is assumed that a component can fail either independently or as a result of a CCF of all m components. If a component fails, the failure is a CCF with probability β. Hence, if the component fails with probability Q, the probability of independent failure is $Q^1 = (1-\beta)\cdot Q$, and the probability of a CCF of all m components is $Q^m = \beta \cdot Q$.

The α-factor model considers the possibility that a subset of m components can fail due to a common cause, i.e. CCFs between different component combinations are possible. The formulas for the α-factor model are

$$Q^k = \frac{k}{\binom{m-1}{k-1}} \frac{\alpha_k}{\alpha_{tot}} Q, \tag{1}$$

$$\alpha_{tot} = \sum_{k=1}^{m} k\alpha_k, \tag{2}$$

where α_k is the fraction of CCFs with k components, i.e. given that a failure event occurs, it is a CCF of k components with probability α_k if the sum of the parameters is 1.

4.2 Common cause failure models for DFM

When failures of multiple similar components can jointly cause the top event, the possibility of a CCF needs to be considered. This is the case when the failures of similar components appear in the same prime implicant in DFM. The failure times of the components can be different in DFM. For example, assume that a set of literals $\{A_1(-3), F_1^1(-1), F_1^2(-2)\}$ is a prime implicant, F^1 and F^2 are failure nodes representing identical components, and the initial time is -3. The first component has failure time -1, and the second component has failure time -2. Assuming that a CCF is possible, there should be a prime implicant that is otherwise similar except that those individual failures are replaced by a CCF. The CCF could be such that both components fail at time step -2. However, the first component represented by failure node F^1 could also fail at time step -1 so that the top event would occur. We introduce the CCF literal $F_1(-1,-2)$ that indicates that the first component is failed at time -1 and the second component at time -2. The probability of this literal is the sum of the following two probabilities:

1. Probability that the components fail simultaneously at time step -2,

2. Probability that the first component fails at time step -1 and the second component fails at time step -2.

The probability that the components fail simultaneously at time step -2 can be calculated with the β-factor model as

$$P(F_1(-2,-2)) = \beta \cdot P(F_1^1(-2)) = \beta \cdot P(F_1^2(-2)).$$

The author does not have knowledge of any existing timing-related CCF parameter estimation work. However, simultaneous CCFs can be expected to be more probable than non-simultaneous CCFs in most cases. Therefore, the possibility of a non-simultaneous CCF is accounted for by the following computation formula:

$$P(F_1(-1,-2)) = \beta \cdot \frac{P(F_1^1(-1)) + P(F_1^2(-2))}{2}.$$

The CCF probability is the β-factor multiplied by the average of the individual component failure probabilities. The probability that the failures are non-simultaneous is therefore

$$\beta \cdot \frac{P(F_1^1(-1)) - P(F_1^1(-2))}{2},$$

which is half of the probability that the components fail simultaneously at time step -1 due to a common cause. Because simultaneous CCFs are expected to be more likely, this probability estimate is considered to be conservative.

The more general computation formula can be presented as follows. Consider a literal $C_1(-t_1, -t_2, ..., -t_m)$ that represents a CCF of m components represented by failure nodes F^1, F^2,..., F^m with failure times $-t_1, -t_2, ..., -t_m$. Using the β-factor model, the CCF probability is

$$P_\pi(C_1(-t_1, -t_2, ..., -t_m)) = \beta \cdot \frac{1}{m} \sum_{i=1}^{m} P_\pi(F_1^i(-t_i)), \tag{3}$$

where π is the prime implicant that includes the CCF literal. This is because the prime implicant can include conditions that some components are functioning at some time steps, i.e. the prime implicant can contain failure nodes F^i in state 0. Probabilities $P_\pi(F_1^i(-t_i))$ are calculated as discussed in Section 2.2.

The computation with the α-factor model is similar. In equation (3), β is changed to $\frac{k}{\binom{m-1}{k-1}} \frac{\alpha_k}{\alpha_{tot}}$ (see equation (1)), and the average is calculated based on the literals of those components which belong to the CCF combination.

4.3 Incorporating CCFs into DFM results

CCF models can be considered in prime implicant post-processing after computing the initial prime implicants. It is possible to construct all prime implicants containing CCFs

from original prime implicants that include individual failures (failure nodes in state 1). Therefore, prime implicants can be generated first using the original DFM model that does not contain CCFs. Prime implicants with CCFs can be created afterwards.

Here, a CCF combination is a set of failure nodes representing components between which a CCF can occur. A β-factor group contains only one CCF combination, whereas an α-factor group contains a CCF combination for each subset containing at least two components of the group. In the following algorithm, all CCF combinations related to the DFM model are examined in order to create all CCF prime implicants. After completing a step in the algorithm, move to the next step if no other instructions are given.

1. Get the next CCF combination.
2. Get the first prime implicant.
3. Check whether the prime implicant contains any of the failure nodes in the CCF combination in state 1. If not, go to step 14.
4. Write down the latest failure time in the prime implicant related to the CCF combination.
5. Initialize a CCF literal based on the CCF combination.
6. Get the first failure node in the CCF combination.
7. Check whether the prime implicant contains the failure node in state 1. If not, go to step 9.
8. Set the time step corresponding to the failure node in the CCF literal to be the same as in the prime implicant. Go to step 11.
9. Check whether the prime implicant contains the failure node in state 0 at the same or later time step as the latest of the failure times that appear in the prime implicant. If it does, go to step 14.
10. Set the time step corresponding to the failure node in the CCF literal to be the same as the latest of the failure times that appear in the prime implicant.
11. Check whether all failure nodes in the CCF combination have been examined. If not, get the next failure node and go to step 7.
12. Create a new prime implicant in which the literals representing failures related to the CCF combination are replaced by the CCF literal.

13. Add the new prime implicant to the list of new prime implicants.

14. Check whether all prime implicants have been examined. If not, get the next prime implicant and go to step 3.

15. Add the new prime implicants to the list of prime implicants.

16. Check whether all CCF combinations have been examined. If not, go to step 1.

Non-minimal implicants and duplicate prime implicants can be created in this process. They originate only from cases for which step 10 is applied. Even if the failure of the component is part of the CCF in the new prime implicant in such a case, it is not really relevant for the occurrence of the top event. CCF prime implicants must be compared to each other so that non-minimal implicants and duplicate prime implicants can be removed. The comparisons can be performed for each CCF combination separately after the corresponding CCF prime implicants have been created, so that the number of comparisons can be kept minimal. The comparisons can also be targeted to check only those CCF prime implicants that can theoretically be non-minimal or duplicate.

Table 2 presents examples of how prime implicants with CCFs are created. In case 4, a prime implicant is not created because literal $F_0^1(-2)$ indicates that the first component needs to function at time step -2. In case 5, implicant $\{A_2(-3), V_1(-2), F_1(-2,-2)\}$ is first created, but it is then removed because it is not minimal compared to the prime implicant of case 1.

Table 2: Examples of creation of CCF prime implicants.

Case	Original prime implicant	CCF prime implicant
1	$\{A_2(-3), F_1^1(-2), F_1^2(-2)\}$	$\{A_2(-3), F_1(-2,-2)\}$
2	$\{A_1(-3), F_1^1(-1), F_1^2(-2)\}$	$\{A_1(-3), F_1(-1,-2)\}$
3	$\{A_2(-3), R_1(-2), F_1^2(-1)\}$	$\{A_2(-3), R_1(-2), F_1(-1,-1)\}$
4	$\{B_0(-3), V_1(-1), F_0^1(-2), F_1^2(-2)\}$	–
5	$\{A_2(-3), V_1(-2), F_1^2(-2)\}$	–

In the following, components represented by failure nodes G^1, G^2 and G^3 belong to an α-factor group, and $\{Z_0(-3), G_1^1(-1), G_1^2(-2)\}$ is a prime implicant. For each combination of G^1, G^2 and G^3, a new prime implicant with the CCF is created. The new prime implicants are presented in Table 3. In addition, assume that each component fails during one time step with probability 0.01, α parameters are $\alpha_1 = 0.5$, $\alpha_2 = 0.05$ and $\alpha_3 = 0.1$, and the initial time is -3. The CCF probabilities are presented in Table 4.

Table 3: CCF prime implicants for α-factor group.

Comb.	Prime implicant
1,2	$\{Z_0(-3), G_1^{1,2}(-1,-2)\}$
1,3	$\{Z_0(-3), G_1^{1,3}(-1,-1), G_1^2(-2)\}$
2,3	$\{Z_0(-3), G_1^1(-1), G_1^{2,3}(-2,-2)\}$
1,2,3	$\{Z_0(-3), G_1^{1,2,3}(-1,-2,-1)\}$

Table 4: Probabilities of CCF events.

CCF	Probability
$G_1^{1,2}(-1,-2)$	8.31E-4
$G_1^{1,3}(-1,-1)$	1.11E-3
$G_1^{2,3}(-2,-2)$	5.56E-4
$G_1^{1,2,3}(-1,-2,-1)$	5.53E-3

5 An example of an emergency core cooling system

This section analyzes an emergency core cooling system (ECCS) of a boiling water reactor plant using the DFM tool Yadrat [7]. A similar system was analyzed in [8], but this version contains two pump lines instead of one. The system is illustrated in Figure 3. The system injects cooling water to the reactor core if the ordinary cooling system does not work. Water level and pressure measurements are utilized in controlling pumps and valves. The pressure vessel's water level decreases due to evaporation if more water is not injected. If the water level is measured to be too low, the pumps are started and the valves are opened in order to inject water into the pressure vessel. The water injection is continued until the water level reaches an upper limit. The valves can be opened only if the pressure is measured to be low.

Figure 4 shows the node structure of the DFM model constructed for the system. The decision tables and descriptions of the nodes are presented in Appendix A. There are two pump lines. Only one pump line needs to function so that the water level (represented by node WL) can be kept within required limits. For a pump line, four components are included in the model: a water level sensor (WLM), a pressure sensor (PM), a regulation valve (V) and a pump (P). Each of these components is modeled with two nodes: a failure node and a deterministic node representing the component's functional state. Both pump lines also include a stochastic node for the leakage signal of the pump (PL). In addition, the model includes a deterministic node WF for water flow as well as nodes representing the control logic and other signals.

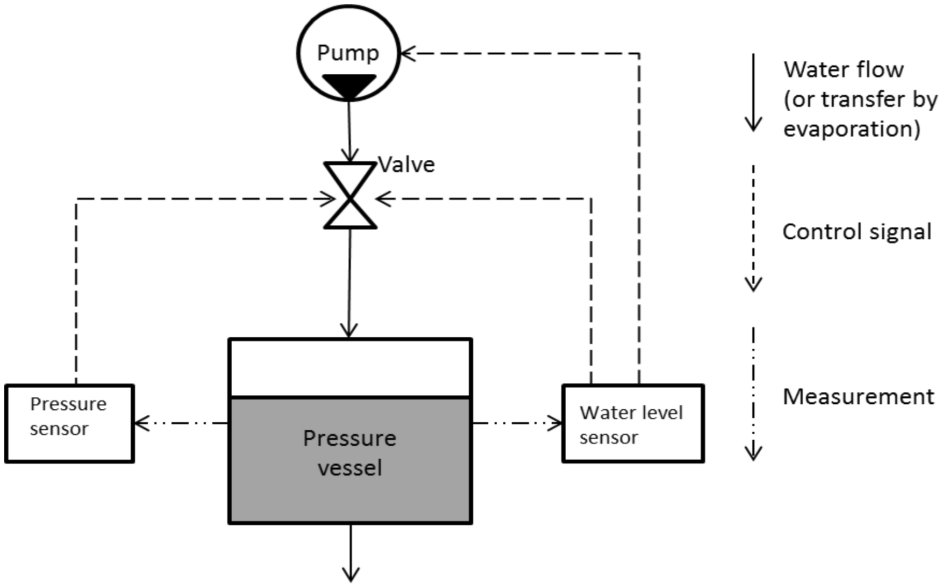

Figure 3: Process flow diagram of an ECCS. [8]

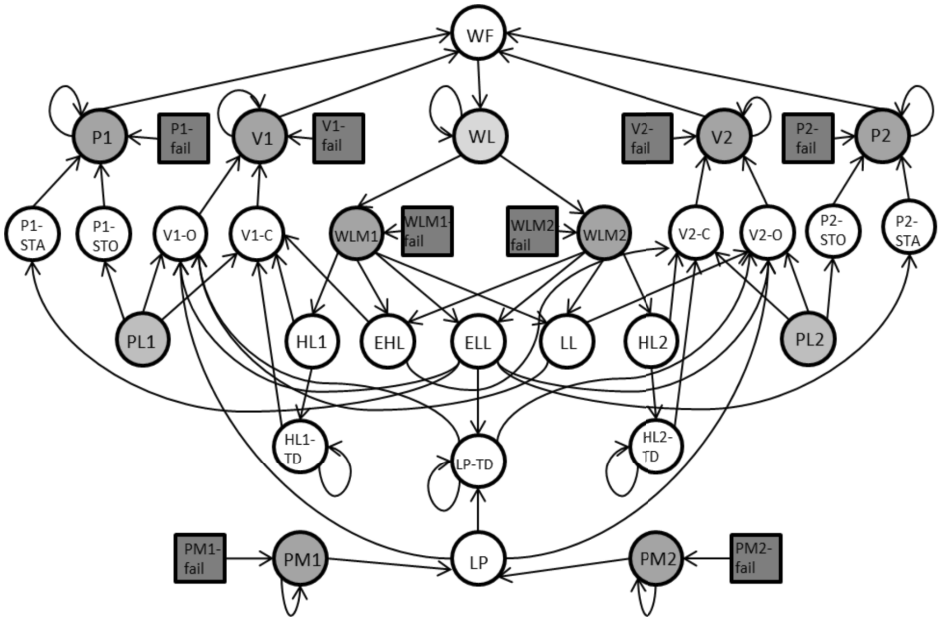

Figure 4: A DFM model for the ECCS.

The scenario that is analyzed is that the water level is too low for too long. The top event is $\{WL_{low}(-1), WL_{low}(0)\}$, and the initial time is -5. The β-factor model is assigned to pumps (P1 and P2), valves (V1 and V2), water level sensors (WLM1 and WLM2) and pressure sensors (PM1 and PM2). Each component fails with probability 0.005 during one time step, and the generic value 0.1 is used for the parameter β in each case.

A total of 42 prime implicants were identified for the top event before CCFs were added. Examples of prime implicants are presented in Table 5. These prime implicants were selected, because they are representative, and CCF prime implicants are created based on them in the next step of the analysis. They are not the prime implicants with the highest probabilities, except in the case of the first two.

Table 5: Examples of prime implicants.

	Total prob.	**Prob.**	**Node**	**Time step**	**State**
1	1.8E-4	1.3E-2	P1-fail	-2	1
		1.3E-2	P2-fail	-2	1
2	1.8E-4	1.3E-2	V1-fail	-2	1
		1.3E-2	P2-fail	-2	1
3	2.0E-5	0.995	PM1-fail	-4	0
		0.995	PM2-fail	-4	0
		4.5E-3	PM1-fail	-3	1
		4.5E-3	PM2-fail	-3	1
4	2.4E-6	1.3E-2	V1-fail	-2	1
		1.8E-2	V2-fail	-1	1
		1.0E-2	PL2	-3	true
5	2.4E-6	1.3E-2	P1-fail	-2	1
		1.8E-2	V2-fail	-1	1
		1.0E-2	PL2	-3	true

Taking CCFs into account, 56 prime implicants were identified, and the top event probability was 6.0E-3. The top event probability is dominated by CCFs. Table 6 presents the CCF prime implicants, which are based on the prime implicants in Table 5.

Table 6: Examples of prime implicants with CCFs.

	Total prob.	**Prob.**	**Node**	**Time step**	**State**
1	1.5E-3	1.5E-3	P-ccf	$-2, -2$	1
2	1.5E-3	1.5E-3	V-ccf	$-2, -2$	1
3	5.0E-4	0.995	PM1-fail	-4	0
		0.995	PM2-fail	-4	0
		5.0E-4	PM-ccf	$-3, -3$	1
4	1.7E-5	1.7E-3	V-ccf	$-2, -1$	1
		1.0E-2	PL2	-3	true
5	2.7E-7	1.3E-2	P1-fail	-2	1
		2.0E-3	V-ccf	$-1, -1$	1
		1.0E-2	PL2	-3	true

6 Discussion and conclusions

This paper has presented an approach for modeling CCFs in DFM. Traditional β- and α-factor models were used, and the same β and α parameters were used as in the traditional case. Therefore, ordinary CCF data [25] can be used in DFM analysis, and no new parameter estimation work is required. However, the analysis could be made more accurate by estimating probability parameters for the timings of failures, for example the probability that the difference between failure times is within a specific interval. Data about failure times is in fact already collected [3], but is utilized only in the classification of events.

Compared to fault tree analysis, DFM analysis is computationally significantly more demanding, but the approach to incorporate CCFs presented in this paper does not complicate the analysis excessively. CCFs do not need to be accounted for when the prime implicants are first solved, and all the prime implicants with CCFs can be created based on the original prime implicants. This approach was chosen so that the graph model itself would not become more complex, which in turn would significantly increase the computational demands and make the analysis more time consuming.

In this paper, component failures were modeled using so-called failure nodes. Failures can be modeled even in other ways in DFM, such as using multi-state nodes with a separate state for each failure mode. The CCF modeling approach can easily be applied to different failure modeling cases.

This paper presented two parametric CCF models, but the presented approach can be generalized for all parametric CCF models that calculate CCF probabilities based on

single component failure probabilities, for example the multiple Greek letter model. In addition, regardless of which CCF model is used, CCFs can be incorporated into the DFM results in the same way as long as the corresponding individual component failures are included in the model.

The chosen CCF modeling approach accounts for the possibility that failure events can be non-simultaneous in a CCF, whereas CCFs have often been interpreted as simultaneous. The average of the probabilities of individual component failures is used in the computation of the CCF probability. The method is simple and, in most cases, conservative, because simultaneous CCFs are more likely. If non-simultaneous CCFs are ignored in the analysis, some CCF probabilities are underestimated, assuming that non-simultaneous CCFs are in fact possible, and some prime implicants are also left out.

In practice, it is component specific how probable non-simultaneous and simultaneous CCFs actually are. Some components may generally fail simultaneously, whereas for some components the fraction of non-simultaneous failures may be significant. If timing related CCF parameters were estimated, the computation formula of the CCF probability could be made more accurate. Data analyses would be needed to study to which failure modes time-dependent models should be applied.

In the past, DFM has been considered too complex to be applied to large systems, and most applications reported in the literature are rather small. However, more efficient DFM tools are being developed, and computers are becoming more and more powerful. Recent DFM models have been larger [11, 16] and this development will probably continue in the future. When larger systems with redundancies are modeled using DFM, CCFs are also more important. This paper has presented how CCFs can be incorporated in the analysis in a simple and efficient manner.

References

[1] A. Høyland, and M. Rausand, "Dependent failures," in A. Høyland, and M. Rausand, *System reliability theory: Models and statistical methods*. New York: Wiley Series in probability and mathematical statistics: Applied probability and statistics section, pp. 325-354, 1994.

[2] A. Mosleh, K. N. Fleming, G. W. Parry, H. M. Paula, D. H. Worledge, and D. M. Rasmuson, "Procedures for treating common cause failures in safety and reliability studies: Procedular framework and examples," U.S. Nuclear regulatory commission, Division of reactor and plant systems, Washington D.C., NUREG/CR-4780 EPRI NP-5613 Vol. 1., Jan. 1988.

[3] T. E. Wierman, D. M. Rasmuson, and A. Mosleh, "Common-cause failure database and analysis system: Event data collection, classification, and coding," U.S. Nuclear regulatory commission, Division of risk assessment and special projects, Washington D.C., NUREG/CR-6268, Rev. 1 INL/EXT-07-12969, Sep. 2007.

[4] C. J. Garrett, S. B. Guarro, and G. E. Apostolakis, "The dynamic flowgraph methodology for assessing the dependability of embedded software systems," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 25, pp. 824-840, 1995.

[5] A. W. Al-Dabbagh, and L. Lu, "Reliability modeling of networked control systems using dynamic flowgraph methodology," *Reliability Engineering and System Safety*, vol. 95, pp. 1202-1209, 2010.

[6] A. W. Al-Dabbagh, and L. Lu, "Dynamic flowgraph modeling of process and control systems of a nuclear-based hydrogen production plant," *International Journal of Hydrogen Energy*, vol. 35, pp. 9569-9580, 2010.

[7] K. Björkman, "Solving dynamic flowgraph methodology models using binary decision diagrams," *Reliability Engineering and System Safety*, vol. 111, pp. 206-216, 2013.

[8] T. Tyrväinen, "Risk importance measures in the dynamic flowgraph methodology," *Reliability Engineering and System Safety*, vol. 118, pp. 35-50, 2013.

[9] S. Guarro, M. Yau, and S. Dixon, "Applications of the dynamic flowgraph methodology to dynamic modeling and analysis," in *Proceedings of the 11th International Probabilistic Safety Assessment and Management Conference & The Annual European Safety and Reliability Conference*, Helsinki, Finland, Jun. 25-29, 2012;

[10] S. Guarro, M. Yau, and S. Dixon, "Advanced risk modeling and risk-informed testing of digital instrumentation and control systems," in *Proceedings of the Probabilistic Safety Assessment Conference (PSA-11)*, Wilmington, NC, Mar. 13-17, 2011.

[11] P. McNelles, Z. C. Zeng, G. Renganathan, G. Lamarre, Y. Akl, and L. Lu, "A comparison of fault trees and the dynamic flowgraph methodology for the analysis of FPGA-based safety systems part 1: reactor trip logic loop reliability analysis," *Reliability Engineering and System Safety*, vol. 153, pp. 135-150, 2016.

[12] T. Aldemir, D. W. Miller, M. P Stovsky, J. Kirschenbaum, P. Bucci, A. W. Fentiman, and L. T. Mangan, "Current state of reliability modeling methodologies for digital systems and their acceptance criteria for nuclear power plant assessment," U.S. Nuclear regulatory commission, division of fuel, engineering and radiological research, Washington D.C., NUREG/CR-6901, Feb. 2006.

[13] P. Bucci, J. Kirschenbaum, L. A. Mangan, T. Aldemir, C. Smith, and T. Wood, "Construction of event-tree/fault-tree models from a Markov approach to dynamic system reliability," *Reliability Engineering and System Safety*, vol. 93, pp. 1616-1627, 2008.

[14] I. B. Gomes, P. L. C. Saldanha, and P. F. F. Frutuoso e Melo, "A cell-to-cell Markovian model for the reliability of a digital control system of a steam generator," in *Proceedings of the 2013 International Nuclear Atlantic Conference - INAC 2013*, Recife, Brazil, Nov. 24-29, 2013.

[15] T. Tyrväinen, "Prime implicants in dynamic reliability analysis," *Reliability Engineering and System Safety*, vol. 146, pp. 39-46, 2016.

[16] M. Yau, S. Dixon, and S. Guarro, "Application of the dynamic flowgraph methodology to the space propulsion system benchmark problem," in *Proceedings of the 12th International Probabilistic Safety Assessment and Management Conference*, Sheraton Waikiki, Honolulu, Hawaii, USA, Jun. 22-27, 2014.

[17] M. Chebila, and F. Innal, "Unification of common cause failures' parametric models using a generic Markovian model," *Journal of Failure Analysis and Prevention*, vol. 14, pp. 426-434, 2014.

[18] G. A. Ershov, Y. L. Ermakovich, A. A. Kalinkin, A. I. Kalinkin, and B. K Buharin, "Evaluation of NPP protection against common-cause failures," *Atomic Energy*, vol. 114, pp. 391-398, 2013.

[19] T. Tyrväinen, and K. Björkman, "Modelling common cause failures and computing risk importance measures in the dynamic flowgraph methodology," in *Proceedings of the 11th International Probabilistic Safety Assessment and Management Conference & The Annual European Safety and Reliability Conference*, Helsinki, Finland, Jun. 25-29, 2012.

[20] T. Aldemir, S. Guarro, D. Mandelli, J. Kirschenbaum, L. A. Mangan, P. Bucci, M. Yau, E. Ekici, D. W. Miller, X. Sun, and S. A. Arndt, "Probabilistic risk assessment modeling of digital instrumentation and control systems using two dynamic methodologies," *Reliability Engineering and System Safety*, vol. 95, pp. 1011-1039, 2010.

[21] H. Guo, and X. Yang, "Automatic creation of Markov models for reliability assessment of safety instrumented systems," *Reliability Engineering and System Safety*, vol. 93, pp. 807-815, 2008.

[22] W. E. Vesely, M. Stamatelatos, J. Dugan, J. Fragola, J. Minarick, and J. Railsback, "Fault tree handbook with aerospace applications, Version 1.1," NASA Office of Safety and Mission Assurance, Washington D.C., 2002.

[23] L. Xing, A. Shrestha, L. Meshkat, and W. Wang, "Incorporating common-cause failures into the modular hierarchical systems analysis," *IEEE transactions on Reliability*, vol. 58, pp. 10-19, 2009.

[24] T. Matsuoka, and M. Kobayashi, "The GO-FLOW reliability analysis methodology - Analysis of common cause failures with uncertainty," *Nuclear Engineering and Design*, vol. 175, pp. 205-14, 1997.

[25] A. Mosleh, D. M. Rasmuson, and F. M. Marshall, "Guidelines on modelling common-cause failures in probabilistic risk assessment," U.S. Nuclear regulatory commission, Safety programs division, Washington D.C., NUREG/CR-5485, INEEL/EXT-97-01327, Nov. 1998.

A Emergency core cooling system model

Table A-1 presents descriptions for the nodes of the ECCS model presented in Figure 4. In the node names in the table, 'x' can be replaced by 1 or 2. Tables from A-2 to A-28 present the decision tables of the nodes.

Table A-1: The nodes of the ECCS model (Figure 4).

Node	**Description**
EHL	Extremely high water level signal
ELL	Extremely low water level signal
HLx	High water level signal
HLx-TD	Time delay condition for the high water level signal
LL	Low water level signal
LP	Low pressure signal
LP-TD	Time delay condition for the low pressure signal
Px	Pump
PLx	Pump leakage signal
PMx	Pressure measurement
Px-STA	Pump start signal
Px-STO	Pump stop signal
Vx	Valve
Vx-C	Valve close signal
Vx-O	Valve open signal
WF	Water flow
WL	Water level
WLMx	Water level measurement

Table A-2: The decision table of node WL.

	Output	Inputs	
Node	WL	WF	WL
Time lag		1	1
	high	high	high
	high	high	medium
	medium	high	low
	medium	low	high
	low	low	medium
	low	low	low

Table A-3: The decision table of node WF.

	Output	Inputs			
Node	WF	P1	V1	P2	V2
Time lag		0	0	0	0
	high	on	open	on	open
	high	on	open	on	close
	high	on	open	off	open
	high	on	open	off	close
	high	on	close	on	open
	low	on	close	on	close
	low	on	close	off	open
	low	on	close	off	close
	high	off	open	on	open
	low	off	open	on	close
	low	off	open	off	open
	low	off	open	off	close
	high	off	close	on	open
	low	off	close	on	close
	low	off	close	off	open
	low	off	close	off	close

Table A-4: The decision table of P1.

	Output	Inputs			
Node	P1	P1-fail	P1-STA	P1-STO	P1
Time lag		0	1	1	1
	off	1	true	true	on
	off	1	true	true	off
	off	1	true	false	on
	off	1	true	false	off
	off	1	false	true	on
	off	1	false	true	off
	off	1	false	false	on
	off	1	false	false	off
	off	0	true	true	on
	off	0	true	true	off
	on	0	true	false	on
	on	0	true	false	off
	off	0	false	true	on
	off	0	false	true	off
	on	0	false	false	on
	off	0	false	false	off

Table A-5: The decision table of P2.

	Output	Inputs			
Node	P2	P2-fail	P2-STA	P2-STO	P2
Time lag		0	1	1	1
	off	1	true	true	on
	off	1	true	true	off
	off	1	true	false	on
	off	1	true	false	off
	off	1	false	true	on
	off	1	false	true	off
	off	1	false	false	on
	off	1	false	false	off
	off	0	true	true	on
	off	0	true	true	off
	on	0	true	false	on
	on	0	true	false	off
	off	0	false	true	on
	off	0	false	true	off
	on	0	false	false	on
	off	0	false	false	off

Table A-6: The decision table of V1.

	Output	Inputs			
Node	V1	V1-fail	V1-O	V1-C	V1
Time lag		0	1	1	1
	open	1	true	true	open
	close	1	true	true	close
	open	1	true	false	open
	close	1	true	false	close
	open	1	false	true	open
	close	1	false	true	close
	open	1	false	false	open
	close	1	false	false	close
	close	0	true	true	open
	close	0	true	true	close
	open	0	true	false	open
	open	0	true	false	close
	close	0	false	true	open
	close	0	false	true	close
	open	0	false	false	open
	close	0	false	false	close

Table A-7: The decision table of V2.

	Output	Inputs			
Node	V2	V2-fail	V2-O	V2-C	V2
Time lag		0	1	1	1
	open	1	true	true	open
	close	1	true	true	close
	open	1	true	false	open
	close	1	true	false	close
	open	1	false	true	open
	close	1	false	true	close
	open	1	false	false	open
	close	1	false	false	close
	close	0	true	true	open
	close	0	true	true	close
	open	0	true	false	open
	open	0	true	false	close
	close	0	false	true	open
	close	0	false	true	close
	open	0	false	false	open
	close	0	false	false	close

Table A-8: The decision table of P1-STA.

	Output	Input
Node	P1-STA	ELL
Time lag		0
	true	true
	false	false

Table A-9: The decision table of P1-STO.

	Output	Input
Node	P1-STO	PL1
Time lag		0
	true	true
	false	false

Table A-10: The decision table of P2-STA.

	Output	Input
Node	P2-STA	ELL
Time lag		0
	true	true
	false	false

Table A-11: The decision table of P2-STO.

	Output	Input
Node	P2-STO	PL2
Time lag		0
	true	true
	false	false

Table A-12: The decision table of V1-O.

	Output	Inputs				
Node	V1-O	LL	LP	ELL	PL1	LP-TD
Time lag		0	0	0	0	0
	false	true	true	true	true	true
	false	true	true	true	true	false
	true	true	true	true	false	true
	true	true	true	true	false	false
	false	true	true	false	true	true
	false	true	true	false	true	false
	false	true	true	false	false	true
	false	true	true	false	false	false
	false	true	false	true	true	true
	false	true	false	true	true	false
	false	true	false	true	false	true
	false	true	false	true	false	false
	false	true	false	false	true	true
	false	true	false	false	true	false
	false	true	false	false	false	true
	false	true	false	false	false	false
	false	false	true	true	true	true
	false	false	true	true	true	false
	true	false	true	true	false	true
	false	false	true	true	false	false
	false	false	true	false	true	true
	false	false	true	false	true	false
	false	false	true	false	false	true
	false	false	true	false	false	false
	false	false	false	true	true	true
	false	false	false	true	true	false
	false	false	false	true	false	true
	false	false	false	true	false	false
	false	false	false	false	true	true
	false	false	false	false	true	false
	false	false	false	false	false	true
	false	false	false	false	false	false

Table A-13: The decision table of V1-C.

	Output	Inputs			
Node	V1-C	PL1	EHL	HL1	HL1-TD
Time lag		0	0	0	0
	true	true	true	true	true
	true	true	true	true	false
	true	true	true	false	true
	true	true	true	false	false
	true	true	false	true	true
	true	true	false	true	false
	true	true	false	false	true
	true	true	false	false	false
	true	false	true	true	true
	false	false	true	true	false
	false	false	true	false	true
	false	false	true	false	false
	false	false	false	true	true
	false	false	false	true	false
	false	false	false	false	true
	false	false	false	false	false

Table A-14: The decision table of V2-O.

	Output	Inputs				
Node	V2-O	LL	LP	ELL	PL2	LP-TD
Time lag		0	0	0	0	0
	false	true	true	true	true	true
	false	true	true	true	true	false
	true	true	true	true	false	true
	true	true	true	true	false	false
	false	true	true	false	true	true
	false	true	true	false	true	false
	false	true	true	false	false	true
	false	true	true	false	false	false
	false	true	false	true	true	true
	false	true	false	true	true	false
	false	true	false	true	false	true
	false	true	false	true	false	false
	false	true	false	false	true	true
	false	true	false	false	true	false
	false	true	false	false	false	true
	false	true	false	false	false	false
	false	false	true	true	true	true
	false	false	true	true	true	false
	true	false	true	true	false	true
	false	false	true	true	false	false
	false	false	true	false	true	true
	false	false	true	false	true	false
	false	false	true	false	false	true
	false	false	true	false	false	false
	false	false	false	true	true	true
	false	false	false	true	true	false
	false	false	false	true	false	true
	false	false	false	true	false	false
	false	false	false	false	true	true
	false	false	false	false	true	false
	false	false	false	false	false	true
	false	false	false	false	false	false

Table A-15: The decision table of V2-C.

	Output	Inputs			
Node	V2-C	PL2	EHL	HL2	HL2-TD
Time lag		0	0	0	0
	true	true	true	true	true
	true	true	true	true	false
	true	true	true	false	true
	true	true	true	false	false
	true	true	false	true	true
	true	true	false	true	false
	true	true	false	false	true
	true	true	false	false	false
	true	false	true	true	true
	false	false	true	true	false
	false	false	true	false	true
	false	false	true	false	false
	false	false	false	true	true
	false	false	false	true	false
	false	false	false	false	true
	false	false	false	false	false

Table A-16: The decision table of HL1-TD.

	Output	Inputs	
Node	HL1-TD	HL1	HL1-TD
Time lag		0	1
	false	true	true
	true	true	false
	false	false	true
	false	false	false

Table A-17: The decision table of HL2-TD.

	Output	Inputs	
Node	HL2-TD	HL2	HL2-TD
Time lag		0	1
	false	true	true
	true	true	false
	false	false	true
	false	false	false

Table A-18: The decision table of LP-TD.

	Output	Inputs		
Node	LP-TD	LP	ELL	LP-TD
Time lag		0	0	1
	false	true	true	true
	true	true	true	false
	false	true	false	true
	false	true	false	false
	false	false	true	true
	false	false	true	false
	false	false	false	true
	false	false	false	false

Table A-19: The decision table of LL.

	Output	Inputs	
Node	LL	WLM1	WLM2
Time lag		0	0
	false	high	high
	false	high	medium
	true	high	low
	false	medium	high
	false	medium	medium
	true	medium	low
	true	low	high
	true	low	medium
	true	low	low

Table A-20: The decision table of ELL.

	Output	Inputs	
Node	ELL	WLM1	WLM2
Time lag		0	0
	false	high	high
	false	high	medium
	true	high	low
	false	medium	high
	false	medium	medium
	true	medium	low
	true	low	high
	true	low	medium
	true	low	low

Table A-21: The decision table of EHL.

	Output	Inputs	
Node	EHL	WLM1	WLM2
Time lag		0	0
	true	high	high
	true	high	medium
	true	high	low
	true	medium	high
	false	medium	medium
	false	medium	low
	true	low	high
	false	low	medium
	false	low	low

Table A-22: The decision table of HL1.

	Output	Inputs
Node	HL1	WLM1
Time lag		0
	true	high
	false	medium
	false	low

Table A-23: The decision table of HL2.

	Output	Inputs
Node	HL2	WLM2
Time lag		0
	true	high
	false	medium
	false	low

Table A-24: The decision table of LP.

	Output	Inputs	
Node	LP	PM1	PM2
Time lag		0	0
	false	high	high
	true	high	low
	true	low	high
	true	low	low

Table A-25: The decision table of WLM1.

	Output	Inputs		
Node	WLM1	WLM1-fail	WLM1	WL
Time lag		0	1	1
	high	1	high	high
	high	1	high	medium
	high	1	high	low
	medium	1	medium	high
	medium	1	medium	medium
	medium	1	medium	low
	low	1	low	high
	low	1	low	medium
	low	1	low	low
	high	0	high	high
	medium	0	high	medium
	low	0	high	low
	high	0	medium	high
	medium	0	medium	medium
	low	0	medium	low
	high	0	low	high
	medium	0	low	medium
	low	0	low	low

Table A-26: The decision table of WLM2.

	Output	Inputs		
Node	WLM2	WLM2-fail	WLM2	WL
Time lag		0	1	1
	high	1	high	high
	high	1	high	medium
	high	1	high	low
	medium	1	medium	high
	medium	1	medium	medium
	medium	1	medium	low
	low	1	low	high
	low	1	low	medium
	low	1	low	low
	high	0	high	high
	medium	0	high	medium
	low	0	high	low
	high	0	medium	high
	medium	0	medium	medium
	low	0	medium	low
	high	0	low	high
	medium	0	low	medium
	low	0	low	low

Table A-27: The decision table of PM1.

	Output	Inputs	
Node	PM1	PM1-fail	PM1
Time lag		0	1
	low	0	high
	high	0	low
	high	1	high
	low	1	low

Table A-28: The decision table of PM2.

	Output	Inputs	
Node	PM2	PM2-fail	PM2
Time lag		0	1
	low	0	high
	high	0	low
	high	1	high
	low	1	low

Aalto-DD 154/2017
VTT SCIENCE 161

ISBN 978-952-60-7571-6 (printed)
ISBN 978-952-60-7570-9 (pdf)
ISSN-L 1799-4934
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)

978-951-38-8565-6 (printed)
978-951-38-8564-9 (pdf)
2242-119X
2242-119X (printed)
2242-1203 (pdf)

Aalto University
School of Science
Department of Mathematics and Systems Analysis
www.aalto.fi

BUSINESS + ECONOMY

ART + DESIGN + ARCHITECTURE

SCIENCE + TECHNOLOGY

CROSSOVER

DOCTORAL DISSERTATIONS