A method for dynamic process hazard analysis and integrated process safety management

A Method for Dynamic Process Hazard Analysis and Integrated Process Safety Management

CIP-DATA LIBRARY TECHNISCHE UNIVERSITEIT EINDHOVEN

Houtermans, Michel J.M.

A method for dynamic process hazard analysis and integrated process safety management / by Michel J.M. Houtermans. - Eindhoven : Universiteit Eindhoven, 2001.
Proefschrift. - ISBN 90-386-2812-9
NUGI 841
Trefwoorden: veiligheidssystemen / Dynamic Flowgraph Methodology / procesgevaren ; risico analyse / procesveiligheid ; management / real-time alarm management systeem
Subject headings: safety systems / Dynamic Flowgraph Methodology / process hazard analysis / process safety management / real-time alarm management system; importance measures

First Printing May 2001

Printed by: University printing office, Eindhoven

A Method for Dynamic Process Hazard Analysis and Integrated Process Safety Management

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven, op gezag van de Rector Magnificus, prof.dr. M. Rem, voor een commissie aangewezen door het College voor Promoties in het openbaar te verdedigen op woensdag 23 mei 2001 om 14.00 uur

door

Michel J.M. Houtermans

geboren te Schinveld

Dit proefschrift is goedgekeurd door de promotoren:

prof.dr.ir. A.C. Brombacher
en
prof. G.E. Apostolakis, Ph.D.

Copromotor:
D.M. Karydas, P.E. , Ph.D.

Abstract

The goal of this thesis is to support hazard analysis and safety management by developing a new method that is capable of modeling and analyzing all technical aspects that can affect the safe operation of a manufacturing plant. The dynamic flowgraph methodology (DFM) is used to develop a lifecycle safety management prioritization method that enables analyses to understand a (manufacturing) process in terms of deviations (hazards), which affect safety. Safety is addressed in terms of safety and health of people and the environment. Although the focus in this thesis is mainly on the safety aspects it is demonstrated that safety and quality analysis can be integrated.

The safety management prioritization method is based on existing probabilistic and newly developed non-probabilistic importance measures. These importance measures are used to define filters and rules that support the design and verification of industrial processes. To support the safe operation of the process a real-time alarm management system is developed using the concept of residual probability. This real-time alarm management system will enable the operator to make risk informed decisions on how to operate the process in a safe manner.

This work will be used to demonstrate that the method can be used to provide crucial information for optimization of the design and operation of the manufacturing plant in terms of safety. The usefulness and validation of the method is demonstrated with practical examples. The theories provided in this thesis can be generalized and are applicable to any phase of the lifecycle and any technical aspect of a manufacturing plant.

Samenvatting

Het doel van dit proefschrift is de ontwikkeling van een nieuwe methodiek, ter ondersteuning van proces gevaren analyse en veiligheidsmanagement. Deze methodiek maakt het mogelijk om technische aspekten die effect kunnen hebben op de veilige operatie van een fabricageproces te modelleren en analyseren. De dynamic flowgraph methodology (DFM) wordt gebruikt om prioriteiten vast te kunnen stellen gedurende de levenscyclus van een veiligheidssysteem. De ontwikkelde methodiek maakt het mogelijk om een (produktie) proces te analyseren met betrekking tot de relatie tussen afwijkingen in het proces en het effect van deze afwijkingen op de veiligheid van dit proces. Met veiligheid wordt hier bedoeld de veiligheid en gezondheid van mensen en hun omgeving. De focus in dit proefschrift is op veiligheid maar het wordt aangetoond dat het mogelijk is om veiligheids- en kwaliteitsanalyse te integreren.

De ontwikkelde methode is gebaseerd op bestaande probabilistische en nieuw ontwikkelde niet-probabilistische "importance criteria". Deze criteria worden gebruikt om regels te definiëren die het ontwerp en de verificatie van industriële processen kunnen ondersteunen. Een real-time alarm management systeem, gebaseerd op analyse van restkansen werd ontwikkeld. Dit systeem kan gebruikt worden ter ondersteuning van de veiligheid van het proces tijdens operatie. Met dit real-time alarm management systeem kan de operator beslissingen nemen met betrekking tot de veiligheid van het proces gebaseerd op real-time risico-informatie.

Dit proefschrift laat zien dat de ontwikkelde methode gebruikt kan worden om essentiële veiligheidsinformatie te verschaffen reeds tijdens het ontwerp maar ook gedurende de operatie van een industrieel proces. De methode wordt gevalideerd en de bruikbaarheid van de methode wordt aangetoond door middel van praktische voorbeelden. Het is mogelijk de theorieën in dit proefschrift te generaliseren naar een meer algemene toepassing bruikbaar gedurende meerdere fases van de levenscyclus van een industrieel proces.

Acknowledgement

The time has come to thank everybody that has supported me in the creation of this thesis. Many people, directly or indirectly, supported me and they deserve a special thank you, without them there would be no thesis.

First of all I want to thank Aarnout Brombacher, my professor and first supervisor. Thank you for believing in me and giving me the opportunity to work with you in Eindhoven, Norwood, Munich, Danvers, in Crete, in L.A, from Singapore, in person, or by phone, by mobile phone, by email, by internet video conference, from the office, from home, during lectures, in the plane, on airports, and yes, sometimes even via a piece of paper with some notes (how old fashioned that was). Nothing is too crazy for you to support your students in achieving their goals and no student should be without your enthusiastic support, creative solutions, reviews and comments.

I want to thank George Apostolakis, my second supervisor. Your expertise in this field is invaluable and I consider myself very lucky to have you as part of my committee. I could not have finished my thesis without your support, suggestions, comments, and reviews. I promise that the dinner will be good.

I tried to find them, but existing superlatives are just not big enough to express my gratitude to Dimitrios Karydas. It is good to know that people like you still exist in this world. You have supported me from day one, on academic, professional and even on a personal level. Although it didn't go fast enough, you had all the patience, and more, a student needs. Your academic interests in combination with your practical insights are invaluable. I want to thank you not only for your guidance with my thesis but also for your guidance with my life. I am even more thankful that you were always willing to share your limited time when needed, even when most of that was on the weekends.

I want to thank Peter Sander for his reviews and comments to my thesis. They were very valuable and straightforward. Although, if I can choose between a thesis review and a dinner in New Orleans, I choose the latter.

A special thank you goes out to Sergio Guarro and Michael Yau for letting me use the DFM tool. Without your tool my thesis would not have existed. Special thanks goes out to Michael for helping, supporting, correcting, and reviewing my DFM models. Also Chris Garrett I want to thank for getting me started with DFM.

Another special thanks goes out to my colleagues at the University. I want to thank Jan Rouvroye for all the reviews he did and valuable discussion we had and comments you added to my work. I am very pleased that you continue to work at the university. The students don't know how lucky they are. Too bad we cannot go to conferences anymore, but who knows what the future will bring. I also want to thank Elly van den Bliek and Willie ter Elst. You both did not hesitate a second to help me in the final phase of my thesis and take away a lot of the "regelwerk". I would not have been able to finish everything on time without your support.

For me there is only one guru when it comes to safety and programmable electronic systems, and that is Rainer Faller. I have yet to meet another safety expert who comes even close to you. I want to thank you because your "faster-than-light-decision-making" has certainly helped with finishing my thesis. I am extremely

thankful that you made me part of your team. I learned a lot. I hope that one day everybody understands safety of programmable electronic systems as you do. I want to thank Stephan Aschenbrenner. I remember the day we met like yesterday. It was the first hour of the first day for you at work in the US. We shared a lot over here, from business opportunities, to buffalo wings, to soccer matches. I know it took longer than we first expected but thanks for supporting me in finishing it. I also want to thank Peter Mueller. I enjoyed working with you, whether it was in Munich, Danvers, Oslo, or Vasteras. Your house was always open for me, and there was even space for me to work on my thesis, thanks. All three of you never behaved as my bosses as working with you always felt like teamwork. I certainly appreciate that. I definitely won't forget Jose Marsano. Thanks for reviewing, correcting and commenting on my work. I hope that you find use in the Dutch I taught you, I certainly make use of your Spanish lessons.

I want to thank Tisha Gomez and Stephan Voss for dragging me out when I most needed it. I am very happy to have you as my friends. From now on no more Velcro boy! The same counts for Yizheng Gong and Marc Caluwe. I also want to thank Jim Mitchell, Leon Banchik, and Vedran Peric. You don't know it but you supported me in the quiet way. Thanks guys.

A special thank you goes out to my champion in long distance support Celina Vidal. I cannot belief that there are still people in this world that truly can enjoy the little things in life. Everybody can learn something from you. I value very much your openness and straightforwardness, it saves a lot of time. Your look on life has certainly influenced mine. I finished my project and its time to enjoy. I hope you don't mind traveling.

And of course, last but certainly not least, I want to sincerely thank my parents and my brother. You always believed in me and supported me from start to finish. You were always there when I needed you. I couldn't have done it without you. And of course, also because you supplied me with everything I needed from Holland, like koffie, drop, nasi and bami kruiden, and more. Keep it coming, wherever I am going to be in the future.

Free at last!

Preface

This document describes the work that has been carried out as part of the author's thesis to obtain the degree of Doctor of Philosophy in Mechanical Engineering, at Eindhoven University of Technology, The Netherlands.

Chapter 1 introduces the reader to the general concepts of risk and safety, and how society and industry deal with these topics. It further explains the importance of risk management and addresses hazard and risk analysis. It explains the motivation of this thesis and outlines the chapters that follow.

Chapter 1 demonstrates that safety and quality of a manufacturing process require a thorough understanding of the process. The required understanding to improve or maintain safety is no different than what is needed to improve quality as well. It is possible to integrate the analysis for safety and quality in one approach. The common elements of the integrated analysis will be identified and the main attribute of an integrated tool that can address quality and safety will be defined.

Chapter 3 introduces the general framework used in industry to implement safety. A practical example is introduced to demonstrate the existence of this framework. This example is used throughout this thesis to demonstrate the explained concepts and findings.

Chapter 4 explains the theory and concepts of the dynamic flowgraph methodology. The dynamic flowgraph methodology forms the basis of the process hazard analysis and safety management method explained in this work. As the dynamic flowgraph methodology is a vital part of the method it is explained in depth.

Chapter 5 describes the so-called importance measures useful for safety management. Newly developed non-probabilistic and existing probabilistic importance measures are introduced. These importance measures are integrated with the dynamic flowgraph methodology. The chapter explains the concepts behind the importance measures and summarizes the advantages and disadvantages of the use of these measures.

Chapter 6 outlines how a safe process can be designed utilizing the importance measures. The dynamic flowgraph methodology is used to design different layers of protection that can exist in a manufacturing plant, i.e., not only the hardware that carries out the manufacturing process but also the hardware and software that monitors the process for safety. The concept for a real-time alarm management system is introduced to assist safe operation of the process. The real-time alarm management system can be used to make risk informed decision about the safety of the process during its operation, maintenance and repair.

Chapter 7 demonstrates the concepts described in Chapter 5 and Chapter 6 with examples based on the case study described in Chapter 3.

Chapter 8 summarizes the conclusions of this work and lists recommendations for further research.

Table of contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Introduction

There is no such thing as a risk-free life [1,2]. There are hazards, and thus risks, involved in all human activities [1]. It is not clear whether the risk magnitude has really changed over all this time or not. Are humans more exposed to risk today than they were a thousand years ago, or ten thousand years ago? Risk is definitely not a new phenomenon for mankind, however the kinds of risks humans are exposed to have changed considerably over time, as risk changes, as the societal and natural environment of mankind changes [3]. Natural disasters like hurricanes, floods and earthquakes, and diseases like cholera and plague are examples of major hazards humans had to worry about in the past. Not all-natural problems are eliminated yet and beside natural hazards, new man-made hazards arise almost everyday. Computer viruses are for example fairly recent hazards. In the year 2000 the world got introduced to a computer virus called the “love bug”. This virus has caused an estimated damage of $7 billion US dollars [4]. Our current hazards are not simply natural or technological hazards anymore, as they are often the results of complex interactions between all kind of systems, e.g., technological, ecological, sociopolitical and cultural systems [2,5,6,7]. The exposure to risk is in every corner of our lives, whether we want it or not and whether we know it or not, and technology is one of the systems driving it.

The nature of the hazards influences the perception of the associated risk in an important manner. The following quotation from the American social psychologist W.I. Thomas is very true: "Things which are perceived as real will be real in their consequences" [8]. Most people feel very comfortable driving a car even though deadly accidents happen worldwide on a daily basis. Still most people accept the risks involved in driving a car. Conversely, some people consider the risks associated with air travel unacceptable. Statistics show that the probability of loss of life while flying is lower than the probability of dying while driving. Still, many people perceive the risks involved with flying as higher.

It appears that there is a positive correlation between risk perception and the attitude of humans towards minimizing risk. For example, the French understood already in the early eighteen hundreds that the risk associated with manufacturing of explosives is very high. They introduced strict laws that required the manager of the explosives plant to live on the premises with his family. The Dupont Company has an excellent safety record, because in compliance with that law, it established a safety program that is in place for the last 170 years [9]. Other risks associated with hazards, although more severe, have not been perceived with equivalent response. For example, the risks associated with smoking. Therefore, consistent and rational decisions should not be based on the perception of risk, but on an objective measurement of risk. This is the basic rule necessary to manage risk within industrial facilities.

1.2 Importance of risk management

The concept of risk and its assessment is not new. Documented evidence of risk perception, risk assessment and risk management involving civic behavior and

critical decisions of the State is found already in Thucydides (431 B.C.) [10]. When Pericles, the leader of Athens, delivered his funeral oration speech to honor the soldiers who had died in the war between Athens and Sparta, he outlined the essence of the Athenian democracy and, among other things, said [11]:

> *"We Athenians, in our own persons, take our decisions on policy and submit them to proper discussion. The worst thing is to rush into action before the consequences have been properly debated. And this is another point where we differ from other people. We are capable at the same time of taking risks and of estimating them before hand. Others are brave out of ignorance; and when they stop to think, they begin to fear. But the man who can most truly be accounted brave is he who best knows the meaning of what is sweet in life and what is terrible, and then goes out undeterred to meet what is to come."*

This concept of risk, applied 2500 years ago within the context of the Athenian democracy, is found in our modern times equally applicable in many facets of our complex society. In this present work, risk and its management is addressed within the constraints of industrial activities.

There are two reasons why industrial facilities deal with risk and thus safety management issues. Either they are forced by legal requirements or they see the economic benefits of sound risk management [12,13]. Government bodies, such as the Environmental Protection Agency (EPA) and the Occupational Safety and Health Agency (OSHA) in the USA, require plant owners and operators to evaluate their processes, identify risk and take the necessary precaution to prevent any accidents that are harmful to people and the environment. EPA and OSHA have reflected these requirements in the risk management program [14] and process safety management [15] regulations respectively. The risk management program needs to be implemented, if a facility handles, manufactures, uses, or stores toxic and flammable substances above specified threshold quantities in a process [14]. The process safety management program follows similar requirements. The goal of both programs is to force facility owners and operators to evaluate their processes, identify risks, and take steps to prevent serious accidents.

Two other stakeholders, corporations and insurance companies, address the risk management issue from the economical point of view. Accidents can result in loss of production and give corporations a bad image, resulting in loss of customers and thus revenues. Insurance can provide reimbursement for damaged property and lost income, but it cannot protect market share [16]. Facility owners and insurance companies both recognize the economic benefits of implementing sound risk management programs. These benefits are multifaceted and include:

- Decreasing operating costs by minimizing injuries, property damage, and loss of production;
- Prevention of the potential for accidental releases to the environment;
- Good public image through improved community relations;
- More efficient business performance if carried out correctly;
- Lower insurance premiums because of lower losses.

For the different stakeholders the benefits of risk management are clear, however before risk can be managed it is necessary to understand the notion of risk.

Risk is a set of triplets that answers three questions [17]. What can happen? How likely is that to happen? And if it happens, what are the consequences? For an industrial facility the answer to the first question means that all possible scenarios that can lead to a loss or accident need to be identified. For each potential hazard it has to be determined how this hazard can turn into an undesirable risk [18]. If each scenario is determined it is possible to analyze the risk as a combination of the remaining two questions, i.e., what is the likelihood of a scenario and what is the severity of the consequences of this scenario. Thus, the risk involved with a possible scenario can be influenced by a change in consequence or likelihood. Unless a hazard is eliminated, the associated risk can never be zero. Risk can be made smaller if it is possible to add some additional safeguards, decreasing either the consequences or the likelihood. For the different stakeholders the concept of risk management deals with establishing an organization that is able to effectively manage these three questions to ultimately achieve an acceptable level of risk in terms of safety and/or economic efficiency and viability (or profitability).

There are many aspects that influence the risk associated with the operation of industrial facilities. At the core managing this risk deals with the identification of hazards as well as with the prevention, evaluation, and mitigation of the consequences of accidents that could occur in any phase of the safety lifecycle as a result of failures of equipment, human error, or other threats [1]. On a higher level, industrial risk management also addresses accident and loss prevention policies and procedures of the industrial corporations with the insurance industry, as well as corporate relations with the local, state and federal governments.

It is important to have an organization in place that is capable of handling the organizational and technical aspects of risk management. Therefore, it is very common to follow a safety life cycle to assure that all aspects of risk management are dealt with in a systematic manner [1]. Risk management programs should consider the process design, process technology, installation and commissioning, operational and maintenance activities and procedures, non-routine activities and procedures, emergency preparedness and response plans and procedures, training programs, and other elements which impact the safe operation of industrial plants [15]. If a risk management program wants to be effective, it requires a systematic approach to evaluating the complete industrial process. At the basis of this evaluation is usually a hazard and risk analysis.

1.3 Hazard and risk analysis

Hazard and risk analysis (HRA) is an all-encompassing term that describes actually two distinct sequential steps in a safety lifecycle of a plant [1]. It starts with a hazard analysis, which deals with the identification of the hazard. A successful hazard analysis requires a thorough understanding of the system subject to the analysis and it is not uncommon to involve a multi-disciplinary team of experts to identify all hazards for a complex system. A hazard analysis alone cannot assure safety, but it is a necessary step to identify and thus control or eliminate hazards [2]. The hazards concerned may be to people, to the environment, or to the operational integrity of the equipment involved [19].

Once all hazards are identified it is necessary to investigate the possible accident scenarios. This is the first step in carrying out a risk analysis. A risk analysis puts the hazard in the light of the environment. For a process or manufacturing plant

the environment mainly means taking into account the operational conditions or circumstances. Every potential hazard does not always lead to an accident. For example, a spark is considered a hazard. A vessel containing nitrate acid and pentaerythritol is also considered to be a hazard. The spark on its own cannot cause an accident. A bad mixture of the two materials in combination with an abnormal high temperature can generate explosive gases, which on their own cannot cause an accident. Bringing the spark and the gases together can cause a tremendous explosion and thus a serious accident in a process plant. From this example it can be seen that a mere hazard does not mean an accident or a loss. It is a necessary condition for an accident but no sufficient. Other conditions need to be present for the hazard to turn into a loss.

Risk analysis involves the quantification of the risk. Each accident scenario has associated a frequency of occurrence and consequences. The frequency and the consequence determine together the risk associated with the accident scenario. In order to determine the risk associated with a process plant it is necessary to determine the risk associated with each hazard, and with each accident scenario. Only if the hazards and risks are understood it is possible to manage a process from a safety point of view.

1.4 Objective of this thesis

1.4.1 Motivation

The goal of this thesis is to support process hazard analysis and safety management by developing a new method that is capable of modeling and analyzing all technical aspects that can affect the safe operation of hazardous manufacturing or processing plants. The dynamic flowgraph methodology (DFM) is used to develop a lifecycle safety management prioritization method that enables analysts to understand a (manufacturing) process in terms of deviations or conditions, which affect safety and profitability. Safety is addressed in terms of safety and health of people and the environment. Profitability is addressed in terms of quality and efficiency of the manufacturing process. The focus in this thesis is on the safety aspects, but it will be argued that safety and quality can be integrated and addressed with one and the same method.

The safety management prioritization method is based on existing probabilistic and newly developed non-probabilistic importance measures. These importance measures can be used to define rules that support the analysis of hazards during design and verification of industrial processes in terms of safety. The importance measures are implemented via a software tool.

As it is not enough to only design a safe process also a real-time alarm management system is developed to support safe operation of the process. The real-time alarm management system is based on the concept of residual probability. This real-time alarm management system will enable the operator to make risk informed decisions on how to operate the process in a safe manner.

This work will demonstrate that the method can be used to provide significant information for optimization of the design and operation of the manufacturing plant in terms of safety. The usefulness and validation of the method will be demonstrated with practical examples. The concepts used and findings derived in this thesis are

general in nature and are applicable throughout any phase of the lifecycle and any aspect of a manufacturing plant, including quality.

1.4.2 Outline of this thesis

Chapter 1 demonstrates that safety and quality of a manufacturing process require a thorough understanding of the process. The required understanding to improve or maintain safety is no different than what is needed to improve quality as well. It is possible to integrate the analysis for safety and quality in one approach. The common elements of the integrated analysis will be identified and the main attribute of an integrated tool that can address quality and safety will be defined.

Chapter 3 introduces the general framework used in industry to implement safety. A practical example is introduced to demonstrate the existence of this framework. This example is used throughout this thesis to demonstrate the explained concepts and findings.

Chapter 4 explains the theory and concepts of the dynamic flowgraph methodology. The dynamic flowgraph methodology forms the basis of the process hazard analysis and safety management method explained in this work. As the dynamic flowgraph methodology is a vital part of the method it is explained in depth.

Chapter 5 describes the so-called importance measures useful for safety management. Newly developed non-probabilistic and existing probabilistic importance measures are introduced. These importance measures are integrated with the dynamic flowgraph methodology. The chapter explains the concepts behind the importance measures and summarizes the advantages and disadvantages of the use of these measures.

Chapter 6 outlines how a safe process can be designed utilizing the importance measures. The dynamic flowgraph methodology is used to design different layers of protection that can exist in a manufacturing plant, i.e., not only the hardware that carries out the manufacturing process but also the hardware and software that monitors the process for safety. The dynamic flowgraph methodology and the importance measures cannot only be used for design, but also for verification and validation purposes. The concept for a real-time alarm management system is introduced to assist safe operation of the process. The real-time alarm management system can be used to make risk informed decision about the safety of the process during its operation, maintenance and repair.

Chapter 7 demonstrates the concepts described in Chapter 5 and Chapter 6 with examples based on the case study described in Chapter 3.

Chapter 8 summarizes the conclusions of this work and lists recommendations for further research.

Chapter 2 Integrated Quality, Environment, Safety and Health Analysis

2.1 Introduction

There are indications that the industry programs that address quality, environment, safety and health (QESH) are organizationally interrelated and that industry has attempted to fully or partially integrate these programs. The purpose of this chapter is to examine the industry approach towards these programs, and identify the common elements that support integration of QESH.

When Paul H. O'Neil[1] became CEO of Alcoa, in June 1987, he immediately announced that his top priority was safety. He firmly believed that his company would be more profitable by focusing on safety. He stated that "To focus on safety requires comprehensive understanding of manufacturing processes, and this understanding leads to better, more productive plants". The Alcoa goal is to have an injury-free workplace. Since his appointment as president in 1987 the number of reported accidents went down significantly [12].

In 1990, the Xerox Corporation initiated the environmental leadership through quality program. Since its inception, this program has saved Xerox approximately $100 million per year [20]. The 3M Corporation has the 3P program, Pollution Prevention Pays, which resulted in $810 million savings since 1975. The 3P program has cut the illness and injury rates 50 percent since 1993 [21]. In 1988, the Unocal Corporation introduced the loss control program. Since its initiation, it has achieved a 56% reduction in recordable incidents [22]. While some companies focus on environmental improvements, others focus on safety. The following statements are in the same line of thought and further support the industry policies advocated in the previous examples [23]:

- "We recognize the importance of costing loss events as part of total safety management. Good safety is good business." Dr. J. Whiston, ICI Group Safety, Health and Environment Manager.
- "Safety is without doubt, the most crucial investment we can make" Robert E. McKee, Chairman and Managing Director, Conoco (UK) Ltd.
- "Profits and safety are not in competition. On the contrary: Safety at work is good business." Basil Butler, Managing Director, British Petroleum Co.

The integration of those programs in real industrial situations has produced beneficial results of remarkable value. Xerox improves the environment through quality. It even calls the program "Environmental Excellence through Quality" [20]. 3M's focus is on pollution prevention, but it reports that this program reduces injury and illness of people [21]. Alcoa clearly states that safety requires such a detailed understanding of the process that it identifies opportunities for improvements that increase safety, quality and profitability [12].

---

[1] Currently appointed Secretary of Treasury in the Bush Administration.

In this work it is accepted that the trend in industry is the integration of these programs. The contribution of this work within the scope of this industrial trend is to provide the means to address safety[2] and quality in an integrated manner. The philosophical foundation of this approach is the thorough and systematic understanding of the process. The approach in this work is an examination of the process in terms of its possible deviations. Controlling possible process deviations means identifying what can go wrong and how to prevent the undesirable consequences of safety or quality. In the subsequent sections in Chapter 1 we examine safety and quality in terms of how they are currently addressed. The examination will demonstrate that safety and quality have common management elements, which both depend on a thorough understanding of the process. Only if the process is understood it is possible to identify possible deviations that need to be either eliminated or controlled in order to achieve acceptable safety and quality.

2.2 Describing a process in terms of safety

The ISO/IEC Guide 51 defines safety as “freedom from unacceptable risk” [24]. Therefore, in order to manage safety it is necessary to understand what leads to unacceptable risk. This understanding can be obtained with the identification of important process parameters, their possible deviations from normal conditions, and the consequences of these deviations.

To focus on safety requires a comprehensive understanding of the manufacturing process. It is necessary to understand the process in terms of process parameters and process elements, which include the necessary hardware and software that materialize the process. These process parameters and elements need to be understood in terms of their relationships and possible interaction, and how deviations from the normal quantities, settings, or behavior can affect the safe operation of the process. To achieve a safe operating plant, it is necessary to design a process where possible deviations from normal conditions can be kept within specific limits that are dictated by what is perceived as acceptable risk.

The “design for safety” concept can be addressed in two ways. On one hand there is the use of standards, codes and guidelines and on the other hand there are detailed safety analyses. Standards, codes and guidelines mainly exist because of lessons learned from the past, usually as a result from accidents. Standards and codes deal with implementing requirements for a general process or specific applications based on existing knowledge, for example, codes and regulations for pressure vessels [25] or burner management control systems [26]. An advantage of the use of codes and standards is the limited amount of effort to achieve acceptable safety targets. The trade off of this approach is that only the minimum safety requirements are addressed which do not necessarily cover the acceptable risk of the specific process.

Safety analyses go beyond standards and codes. In other words, analysis is useful when there is a need or desire to explicitly evaluate the risk associated with the process, even after compliance with existing standards and codes.

[2] In this work the term safety encompasses people safety and health as well as environmental protection.

It is acknowledged that implementation of standards and safety analyses are two approaches that complement each other and that actually both should be used [2]. Compliance with the standards and codes achieves a level of required safety, while analyses brings safety within the limits of acceptable risk.

Safety analyses starts usually with a hazard and risk analysis. The objective of a hazard and risk analysis is to identify all hazards and their associated risk. It identifies what can go wrong and how it can be prevented or controlled. As a result of this analysis it is possible to reduce the associated risk to an acceptable level by either changing the design or adding safety measures to the design. A hazard and risk analysis can show that specific hazards are, or are not, present after code requirements have been complied with, that further safety measures are, or are not, needed, and what the possible consequence can be if the hazard causes an accident. The outcome from the hazard and risk analysis are recommendations to improve the plant design, incorporate additional safety measures, or define operation and maintenance procedures that minimize or control potential hazards. In other words, the objective of the safety analysis is to manage process parameters or elements in terms of their deviations. The next section will explain how safety is currently managed in the manufacturing industry.

2.3 Safety management

An industrial system can only be safe if all the individual elements of the system are safe and interact with each other in a safe manner. Safety, like quality or profitability, is a basic property of a system and needs to be addressed in a way that considers the individual elements, and their interaction, within the context of the system. The methods, techniques and resource allocation must be coordinated, well planned, properly justified, and able to address the entire lifecycle of the system; in synergistic and not antagonistic manner with the other basic properties of the system; in other words safety has to be has to be carefully managed.

The Department Of Energy in the US has defined five core functions for safety management that comprise the underlying process for any work activity that could potentially affect the public, the workers, and the environment. These five core functions are (see Figure 1) [27]:

1. Define the scope of work – Missions are translated into work, expectations are set, tasks are identified and prioritized, and resources are allocated.
2. Analyze the hazards – Hazards associated with the work are identified, analyzed and categorized.
3. Develop and Implement Hazard Controls – Applicable standards and requirements are identified and agreed-upon, controls to prevent/mitigate hazards are identified, the safety envelope is established, and controls are implemented
4. Perform work within controls – Readiness is confirmed and work is performed safely
5. Provide feedback and continuous improvement – Feedback information on the adequacy of controls is gathered, opportunities for improving the definition and planning of work are identified and

implemented, line and independent oversight is conducted, and if necessary, regulatory enforcement actions occur.

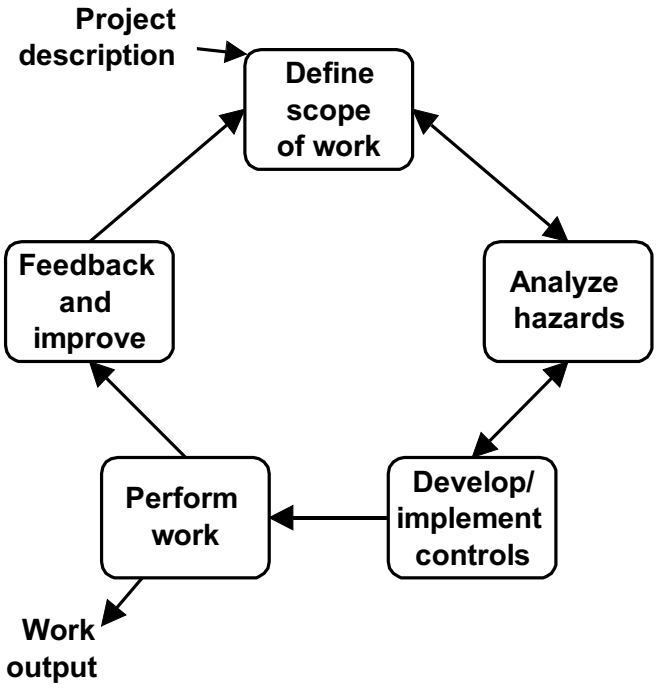

Figure 1. Safety Management Work Cycle [27]

The underlying attribute of the safety management functions is the thorough and an integrated understanding of the process. Several techniques and tools have been developed to address safety management functions within the various phases of the safety lifecycle of the system. These techniques can and are used to collect information about parameters of interest that support the five core safety management functions. Some of these techniques are outlined here.

Checklists [1,2] are very simple in nature and easy to use. They are usually compiled over years over experience and take into account history or "lessons learned". Often they result from design guidelines and good engineering practice derived from well-known and defined processes and plants or from standards and regulations. They list a known set of *hazards* or solutions against these hazards.

Preliminary hazard analysis (PHA) [1,28] is used in the early stages of the lifecycle to identify high level system functions and broad system *hazards*. Of particular importance in PHAs are interface connections between equipment and subsystems. PHAs are used to *make decisions* on accident prevention measures that can be taken to *eliminate* or *control* hazards. The results form the basis for later analysis.

A hazard and operability (HAZOP) [1] study is a technique for *identifying* and *analyzing hazards* and *operational concerns* of a system. As the name reveals it not only focuses on safety but also on operational issues. In a systematic manner every single subsystem in a plant is analyzed, considering the *intended design*, potential *deviations* from this design, and the *causes* and *consequences* of these *deviations*.

A failure mode and effect analysis (FMEA) [29] is a widely used and effective safety analysis technique. The FMEA is a bottom-up or inductive procedure. In a

systematic manner component *failure modes* and their *effects* or *consequences* are *evaluated* on system level. It is possible to *rank* components or sub-sections of the system according to their *importance* in causing a system problem. This ranking method is used to make *decisions* on what problems to address first.

The goal of fault tree analysis (FTA) [30] is to identify the *basic events* that eventually lead to an *undesired* top event. FTA can be used to investigate *causes* of *hazards* or to *identify hazards*. FTA is a deductive analysis, i.e., it follows a top down approach, this in contrast to FMEA. A fault tree only includes those faults that contribute to this top event.

Event Tree Analysis (ETA) [31] is an inductive technique that is meant to identify all possible outcomes of an *initiating event*. They are similar to fault trees but difference is that they initiate events and *examine* the possible *consequences* of these initiating events. The initiating event might be a *failure* of the system or an *external event* to the system. Event tree analysis takes into account the protections systems that are in place that *avoid* an initiating event from resulting into a real *accident*. It identifies possible accident scenarios.

New safety standards, like IEC 61508, ISA S84.01, draft IEC 61511 and draft IEC 62061 require quantitative analysis to justify the designs of plants and implemented safety systems in terms of the safety performance. Parts count analysis [32], reliability block diagrams [33], FTA [30], and Markov analysis [33] are all techniques that are being used to carry out probabilistic calculations. Markov analysis has been identified as the most flexible and capable quantitative technique to address safety for design and verification activities [34,35]. *Deviations* in the performance of designs might as well affect safety and needs also to be addressed in order to get a true understanding of safety. Rouvroye demonstrates in [34,36] that it is possible to enhance Markov analysis with uncertainty and sensitivity analysis to analyze the influence of uncertain reliability data (i.e., deviations in data) on the performance a specified design. With this method it is possible to make design changes based on probabilistic analyses.

All techniques and methods outlined above focus on physical equipment and / or processes. Modern plants are heavily dependent on programmable electronic systems where software plays an extremely important role regarding safety. Therefore, the basic safety management issues must be extended beyond hardware and physical equipment to cover software as well. The approach used in industry to address software safety management has two facets. On one hand the quality of the software process or organization that developed the software is examined. On the other hand the actual software is examined and tested for safety. The traditional techniques used to analyze software safety include an evaluation of the software development process, software requirement specification analysis, software *criticality* analysis, software module and system testing, and software fault injection testing [37].

New developments in software safety recommend not only to analyze software in isolation but also to address the software in the context of its operating environment. Garrett defined this as the error-forcing context of software [38]. An overview of software hazard analysis techniques is given in Table 1. Software hazard analysis is directly related to system hazard since it depends upon system hazard analysis for its inputs. In [95] it is stated that software hazard analysis should:

- Respond to every hazard identified in system hazard analysis;

- Ensure that the operation of the software does not interfere with the goals or operation of the system; and
- Evaluate and make recommendations to mitigate how software could hinder the goals or operation of the system

Table 1. Software Hazard Analysis Techniques Specified by Standards

Technique	Standard
Cause Consequence Diagrams	▪ Software System Safety [39] ▪ Functional Safety of Electrical / Electronic / Programmable Electronic Safety-Related Systems [18]
Code Walk-Throughs	▪ Reviewer Guidance for Computer-Controlled Devices [40] ▪ System Safety Program Requirements [44]
Common Cause Failure Analysis	▪ The Procurement of Safety Critical Software in Defence Equipment [45]
Cross Reference Listing Analysis	▪ System Safety Program Requirements [44]
Design Walk-Throughs	▪ System Safety Program Requirements [44]
Event Tree Analysis	▪ Functional Safety of Electrical / Electronic / Programmable Electronic Safety-Related Systems [18] ▪ Software Safety Plans [42]
Failure Mode, Effects, and (Criticality) Analysis	▪ Reviewer Guidance for Computer-Controlled Devices [40] ▪ Reviewer Guidance for Computer-Controlled Medical Devices Undergoing 510(k) Review [41] ▪ Functional Safety of Electrical / Electronic / Programmable Electronic Safety-Related Systems [18] ▪ Software Safety Plans [42]
(Software) Fault Tree Analysis	▪ Software System Safety [39] ▪ Reviewer Guidance for Computer-Controlled Devices [40] ▪ Functional Safety of Electrical / Electronic / Programmable Electronic Safety-Related Systems [18] ▪ Software Safety Plans [42] ▪ Software Systems Safety Handbook [43] ▪ System Safety Program Requirements [44]
Hazard and Operability Study	▪ Functional Safety of Electrical / Electronic / Programmable Electronic Safety-Related Systems [18]
Monte-Carlo Simulation	▪ Functional Safety of Electrical / Electronic / Programmable Electronic Safety-Related Systems [18]
Nuclear Safety Cross-Check Analysis	▪ Software System Safety [39] ▪ System Safety Program Requirements [44]
Petri Net Analysis	▪ Software System Safety [39] ▪ Software Systems Safety Handbook [43] ▪ System Safety Program Requirements [44]

Technique	Standard
Sneak Circuit Analysis	▪ Software Safety Plans [42]
Software/Hardware Integrated Critical Analysis	▪ System Safety Program Requirements [44]
Software Sneak (Circuit) Analysis	▪ Software System Safety [39] ▪ System Safety Program Requirements [44]

In addition to other techniques, which are not mentioned in this document, these techniques are used to collect information about parameters of interest and make decisions concerning safety. Independent of which technique is used the purpose is to gather information that helps understanding the process that takes place and analyze this process in terms of deviations and how these deviations effect safety. Using the methodologies and techniques described above, current safety analysis can be summarized as follows:

- Safety information needs to be available throughout the life of a manufacturing process. Safety decision need not only be made during design but also during operation, maintenance, repair and other phases (the lifecycle approach). There is a need to be able to design a safe process upfront and to maintain safety through the remaining lifecycle phases;
- Safety information is collected by experience over time or by analyses (see checklists). Safety analysis is carried out analyzing deviations in normal behavior of components and processes. Deviations (failure modes, or process deviations) from intended design are considered, not only their consequences but also their causes. There is a need to see the effects of failure on process level, but also what causes unsafe conditions, that is, one wants to see how failures propagate through the system and what conditions need to exist in order to arrive at an undesired condition (for example FTA vs. FMEA).
- In order to understand safety there is a need to understand the interaction between components, subsystems, and systems in terms of hardware and software (See PHA, HAZOP, FMEA, Software, etc.).
- There is a need to rank information in order to make informed decisions about safety (see FMEA, HAZOP, etc.). This ranking can be done qualitatively, semi-quantitative or quantitative. The purpose of safety analysis is to eliminate, prevent or control safety issues, which can be supported by ranking.
- Safety analysis not only focuses on hardware failures but also on operational issues (see HAZOP, FMEA). Software plays an important role in manufacturing plants and needs to be integrated in the safety analysis within the context of its operating environment.

2.4 Describing the process in terms of quality

ISO defines quality as “the totality of characteristics of an entity that bear on its ability to satisfy stated and implied needs” [46]. This is a very broad definition since a product, process, or service can have many features or characteristics that must abide by the stipulation of this clause. When it comes to a product, quality addresses not only physical characteristics of a product but also features that are indirectly related to the product, like customer service after the product has been bought, or prompt delivery. In order to manufacture a product of the desired quality it is necessary to understand what can influence these product characteristics or features before the customer negatively experiences them.

The aspects that influence the quality of a product are best addressed if the lifecycle of the product is understood. A typical lifecycle of a product is given in Figure 2 [47]. The lifecycle shows the phases from the initial market research through specification, design & development, procurement, production, storage and packaging, sales and distribution, installation and commissioning, maintenance, and eventually disposal of the product. With so many product lifecycle phases, there is no doubt that an enterprise must have a good quality system to deliver a product that actually meets at the end all of the customer’s expectations. For each phase, it is possible to define a set of procedures, which might be work instructions, or the use of tools or methods that aim at improving the quality of the final product. For all phases together, this set of procedures will eventually make up the quality manual that will be part of the quality management system of a company. The next section will explain how quality is currently managed in industry.

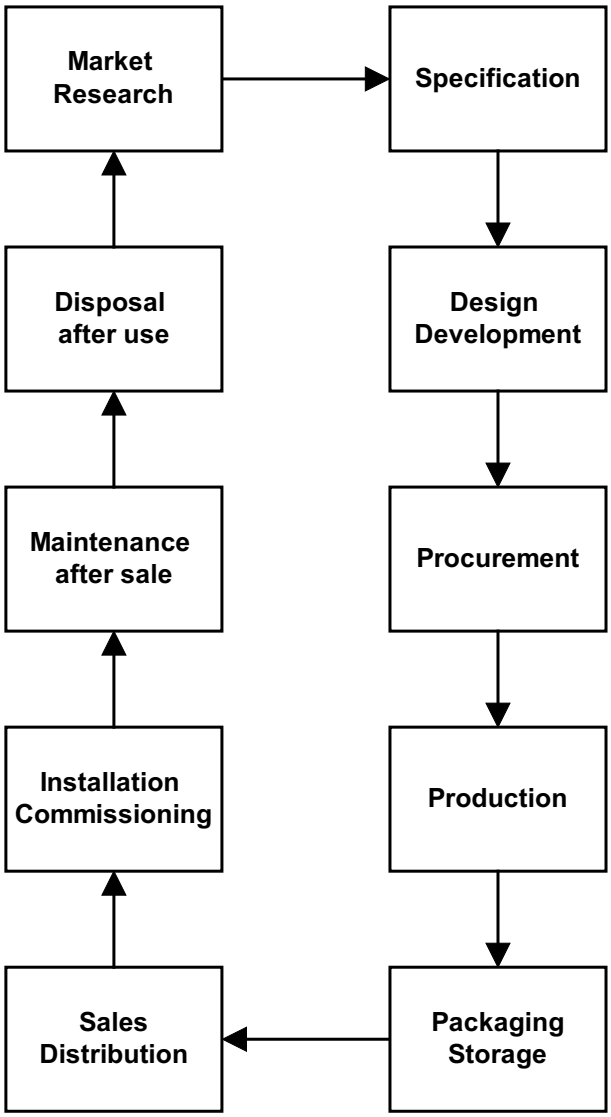

Figure 2. Product Lifecycle [47]

2.5 Quality management

Ultimately products of good quality can only be achieved during manufacturing if a company has the three fundamental quality elements under control, i.e., the organizational, product design, and manufacturing elements. The organizational and product design aspects of quality management are beyond this work, but they can influence the performance of the manufacturing process directly. Just like safety, quality requires a systems approach. To achieve a manufacturing process that can produce quality products, it is necessary to understand the process. The following paragraphs will briefly explain different quality systems that demonstrate this statement.

The family of ISO standards and guidelines, like ISO 9000 [48] or ISO 14000 [49], are one of the best-known quality systems and form the basis in many industries for implementing a quality management system [50]. They describe the minimum measures for an adequate quality management system. The ISO 9000 quality series of standards represents the essential requirements that every enterprise needs to

address to ensure the consistent production and timely delivery of its goods and services to the marketplace [51]. The system standards describe what requirements need to be met, not how they are to be met. This quality management system focuses on a framework that ensures understanding the process that takes place.

The quality management systems approach is similar to the safety management approach outlined in section 2.3, as it consists of the same basic elements. The following describe a summary of some of the elements that comprise a quality management system according to ISO 9000 [52]:

- Determine the needs and expectations of the customer;
- Determining means of preventing nonconformities and eliminating their causes;
- Determine opportunities to improve the process
- Determine and prioritize improvements;
- Assessing the results against the expected outcomes;
- Reviewing the improvement activities to determine appropriate follow-up actions.

The above-mentioned ISO series of standards only describes the framework for implementing a quality management system that has the capability of achieving consistent products and processes of quality (even if that consistency means bad quality all the time). Some examples of more detailed quality systems are the Hazard Analysis and Critical Control Point system, the Capability Maturity Model and the Maturity Index of Reliability model. These quality systems guide an organization with specific steps and methods on how to achieve quality.

HACCP is an acronym for Hazard Analysis and Critical Control Point (pronounced "hassip") and is a food program that was developed nearly 30 years ago for NASA. It ensures the quality of food products used by the astronauts in the space program [53]. Even though it has all the characteristics of a safety analysis technique it is described as a quality management system. Its purpose is to prevent hazards that could introduce potentially dangerous food-born illnesses in food by applying science-based controls that cover all aspects from raw resources through preparation to final product. HACCP is comprised of seven principles of which the most notable can be summarized as following [53]:

- Analyze hazards.
- Identify critical control points at which hazards can be controlled or eliminated.
- Establish critical limits and monitor the critical control points for these limits.
- Establish procedures to verify that the system is working properly.
- Establish effective record keeping in order to document the HACCP system

Analyzing hazards requires a thorough understanding of the process so that hazards can be identified and be eliminated or at least controlled. A major focus point is on the record keeping of hazards and their control methods. Implemented safety requirements are continuously monitored and corrective actions are taken to prevent

problems or how non-conformances (deviations) are to be prevented from reoccurring [53].

The Capability Maturity Model (CMM) is developed by the Software Engineering Institute (SEI) and is a framework that describes key elements of an effective software process. To define what a process is SEI uses the IEEE definition, i.e., a sequence of steps performed for a given purpose [54]. The CMM covers practices for planning, engineering, and managing software development and maintenance [94]. It guides software organizations that want to gain control of their processes for developing and maintaining software and to evolve toward a culture of software engineering and management excellence. The CMM does this by giving guidance in selecting process improvement strategies by determining the current process maturity. The latter is summarized below.

There are five levels of process maturity. These five levels define an ordinal scale for measuring the maturity of an organization's software process and for evaluating its software process capability. They also help an organization *prioritize* its improvements efforts. A company that is at level 1 has a software process that is characterized as ad hoc, and occasionally even chaotic. It is an immature organization. Few processes are defined, and success depends on individual efforts and heroics. Methods and techniques are applied but why and how is not understood by the organization. Only a company that is arrived at level 4 understands its software product and process. At this level an organization applies with skill techniques and tools that support process management. Tools are applied like source code *analyzers*, test coverage analyzers, *problem-tracking* packages, *data collection* and management systems. These techniques and tools all help identify deviations and give the analyst the ability to understand the effect of these deviations on product quality level.

Eindhoven University of Technology and Philips CFT have jointly developed the Maturity Index of Reliability (MIR) concept [94]. The objective of MIR is to analyze the response of a process on internal or external *deviations* [55]. A process can be a technical process, like a manufacturing process, or a business process, like a product development process. MIR analyzes how activities (for example a transformation of either information or material) respond to deviations and how this response propagates through the process in terms of business drivers like quality, reliability, or profitability. MIR assumes that a company is only able to take action if the *relevant information* on process output is available. To be able to prevent the occurrence of a problem in the future it must, first, be known *what exactly caused the problem*, and, second, a solution must be found. The level or quality of response towards the capability to *analyze* and *control deviations* (problems) is captured in five so-called MIR levels [55] (see Figure 3). A process (and thus company) that is in level 0 is uncontrolled. There is no relevant quantitative evidence of the process output of the products. There is no information feedback. Only a company that has a process at level 4 understands the origin of problems what *causes* them and what needs to be done about it is known. The level of knowledge is such that not only root causes of problems or known but also it is possible to *anticipate* and *prevent* similar problems in the future. All corresponding *feedback* or control loops are in place [55].

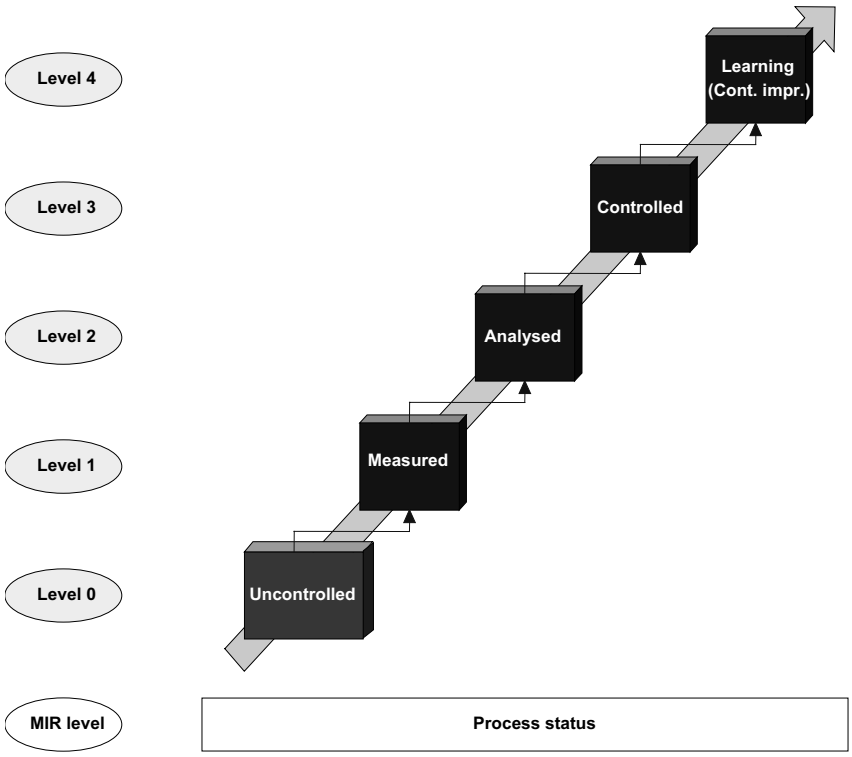

Figure 3. The MIR Model [55]

Many other quality management systems exist that are not mentioned in this work. Some of them are general in nature, other apply to specific industries and/or products. Independent of the quality management system used the purpose is to understand and gather information that helps manage a process in terms of possible deviations and how these deviations affect quality. Using the quality management systems described above as examples the objectives of quality systems can be summarized as follows:

- Quality systems are implemented to ensure consistent and timely delivery of products. Quality needs to be assured through the different lifecycle phases of the product.
- Quality information is collected by understanding the process that takes place. This is achieved by measuring quality, or more specifically by understanding how quality is sacrificed because of deviations of any kind (internal to the process or external).
- Deviations need to be identified and analyzed in terms of how they can influence quality. It is necessary to understand how deviations can propagate through the system. Therefore it is necessary to understand the interaction between the different components, sub systems and systems (hardware and software).
- There is a need to prioritize (rank) deviations and implement control or preventive measures. Not only during design but also during other lifecycle phases like manufacturing (or customer service).

2.6 Integrated management of safety and quality

Previously a process was defined as a sequence of steps performed for a given purpose [54]. A manufacturing process can be defined as a series of operations performed in the making or treatment of a product. This process needs to be understood in order to manage the process in terms of its business drivers. Typical business drivers are quality, profitability, functionality, or time [55]. Also safety can be considered as a business driver. See, for example, the following statement from Unocal in their 1999 environmental performance report [56]. In 1997 and 1998, Unocal paid $375,000 and $3,505,000, respectively, in fines and penalties to government agencies. These include approximately $1.7 million that Unocal paid in penalties in 1998 to various California regulatory agencies related to the settlement for contamination of the Guadalupe oil fields. Unocal lost almost 4 million dollars in two years by not complying with local safety regulations. The ability to manage safety will minimize losses and contribute to profitability.

An enterprise needs to manage its processes to meet the goals set for its business drivers. The output of a process can be measured on any business driver of interest [55]. Any deviations from the goals can result in a loss. An enterprise needs to be able to prevent losses or at least control them to a minimum. Losses can occur in every phase of the product lifecycle, including manufacturing. To prevent or minimize losses during manufacturing it is necessary to understand what can cause these losses. Since a process is a set of operations in the making or treatment of a product, deviations can either occur in the (basic) materials necessary to make the product or in the process equipment, i.e., the hardware and software, necessary to carry out the process.

Controlling the process means to have the ability to manage the process in terms of possible deviations that can affect the business drivers. It means not only handling any possible deviations during the design of the process but also during other lifecycle phases like operation, maintenance, or repair. Deviations can be eliminated, minimized or controlled by implementing design measures or by controlling them during the operational phases of the lifecycle. Once deviations are known they can be prioritized.

2.7 Conclusions

The purpose of this chapter is to demonstrate that management and analysis of safety and quality can be integrated. Quality addresses the elements of the product the way the customer will sense the product, while safety addresses the elements of the means to get the product in the hands of the customer. Even though they are perceived differently, quality and safety analysis have the same goal. They both try to minimize losses by improving the process to deliver a product in the most efficient and profitable way.

Safety and quality can be integrated because they have common management elements that depend on a thorough understanding of the process. In both cases the objectives are to identify problems (hazards, conditions), eliminate or control the problems using appropriate measures, and monitor the implementation of these measures to provide feedback and continuous improvement (see sections 2.3 and 2.5).

Integration of safety and quality also makes sense from the profitability point of view. Quality is a business driver, considering that inappropriate quality decreases profitability. Also safety can be seen as a business driver. See the Unocal example in section 2.6, where profitability is decreased because of the company did not comply with safety regulations. Even though a quality product can be delivered to the customer, non-compliance with safety regulations in the manufacturing process of this product can lead to huge financial penalties or even to a complete shutdown of the manufacturing process. Integration requires only one management system, which is possible since both management systems have the same elements. Focusing on both safety and quality and optimizing the process for both business drivers results in the decrease of losses and improves profitability.

Managing for safety and quality can be summarized as understanding the process and the possible process deviations. Although it is recognized in industry that integration can lead to results in terms of profitability (see section 2.1), no method currently exists to our knowledge that can actually support such integration. In this work a method is presented that is based on a thorough understanding of the process. Only if the process is understood it is possible to identify and analyze the effect of possible deviations. Deviation analysis consists of understanding the root causes of deviations and how these deviations propagate through the process. The method will further focus on the prioritization of these deviations in order to eliminate or control the deviations either during the design of the process or if this is not possible to control the deviations during the following lifecycle phases, like operation, maintenance and repair, retrofit, etc.

Chapter 3 Safety Protection Layers

3.1 Introduction

The purpose of this chapter is to describe the so-called safety protection layer philosophy (SPLP) that exists in modern industrial plants (see Figure 4) [57]. This philosophy represents the implementation of the technical framework used to manage safety associated with operating an industrial process. It is described to explain the technical safety issues that exist in a plant. This philosophy applies in general to any kind of hazardous manufacturing process and plant. As industry tries to implement this philosophy it needs to be able to carry out analyses that support the design and verification of these protection layers. The methodologies used to design and verify these protection layers must address the current state of the art in technology as used in these manufacturing facilities.

To demonstrate the safety protection layer philosophy an example of an existing manufacturing plant will be given. This plant will be used as a case study throughout this thesis. It will be used to explain the tools and theories that will be used throughout this work to support the safety analysis. This example is derived from an existing manufacturing plant [58], and therefore it is not the purpose of this work to justify the existing design of the plant and the safety system. At the end of this chapter a summary is given of the required capabilities of a method that is able to support the implementation of the different protection layers.

3.2 Safety Protection Layer Philosophy

A hazard and risk analysis forms the basis to implement the so-called safety protection layer philosophy (SPLP). These layers of protection can be divided in layers of prevention and layers of mitigation (see Figure 4). The purpose of the prevention layers is to prevent any conditions or events in the plant that can lead to an accident. The purpose of the mitigation layers is to control the consequences once an accident occurs. The focus in this work is on the prevention layers, but both will be explained in the following paragraphs.

The design of the actual process accounts for the first actual safety layer. The purpose of a hazard and risk analysis is to design an “inherent safe plant”. Kletz uses the term "inherently safe design" to describe the philosophy of eliminating or reducing the risk by careful selecting the basic process operating parameters [59]. Other elements of the process design are for example the selection of the process itself, site selection, and the use (or not use) of hazardous materials and their processing conditions. The outcome of the hazard and risk analysis is used to make decisions about design changes, but even if the plant design is optimized there is always remaining risk that can only be reduced by implementing additional layers of protection. These additional safety layers are the basic process control system, critical alarms, operator supervision, manual intervention, and a safety system. A safety system is implemented, carrying out specific identified safety functions, as a last layer of protection.

A basic process control system (BPCS) can be seen as a protective layer but its primary purpose is control and not protection or safety. A BPCS is used to keep

important process parameters between predefined boundaries and to produce products of the desired quality. A BPCS controls the process by executing complex logic provided by the operator. A BPCS responds to alarm levels like Low and High set points. The operator is there to supervise the operations of the BPCS. The BPCS controls the process in an automated way. Operators can intervene in the process and control the process manually at any desired moment. This layer can therefore be seen as an independent protection layer. Operators continuously monitor the process while in the process control room. They can intervene in the process because of an undesired event, like an equipment failure, or because there is a desire to change the operation of a process, e.g., because of startup, shutdown or higher or lower production demands.

Critical alarm levels are available in the process to notify the operator audibly and/or visually. An operator highly depends on the information provided by the BCPS. Wrong sensor data can lead to wrong interpretation of the information available to the operator. A BPCS is designed with a high reliability and availability in mind and is seldom the source of an initiating event that can cause a plant upset. Still, a BPCS failure can occur, and then this failure can lead to the initiating event that eventually can place a demand on the safety system. Other demands on the safety system are related to human errors, the environment or the safety system itself. These safety systems should be implemented as isolated protection layers. This means that they should operate in an automated way and without interference of operators. The process industry refers to safety systems as a safety instrumented system (SIS). Other terms that are common in the process industry include emergency shutdown system, safety shutdown system, or safety interlock system. Nowadays these safety systems are based on programmable electronic technology.

A safety system monitors the industrial process continuously, just like a BPCS. However, a safety system only intervenes in the process when certain predetermined conditions are violated. For example, when the measurement of a temperature or pressure set point is too high. These safety systems play an important role in the SPLP and thus in the risk management of modern process plants. They form a last layer of defense against a possible accident. There performance in terms of safety is so important that industry requires independent certification parties to test and certify the functional safety behavior of these systems [19].

The layers described so far All acted as prevention layers. They try to prevent upset conditions that can lead to accidents. Should all those prevention layers fail to function when required then there are still layers of mitigation present that will help mitigate the consequences of a possible accident. The first mitigation layer is the physical protection layer. The physical protection layer is sometimes divided into the active and passive protection layer. The active protection layer includes devices like rupture disks, sprinkler systems, and explosion- or firewalls. An example of a passive protection layer is a containment dike, which should be able to contain all possible hazardous material spills from a plant or process.

An emergency response layer is always available should a real accident occur. Emergency response layers can be on plant level, depending on the size of the plant, and on community level. They include first aid, fire brigades, and other rescue services. Just like an airport has its own fire brigade, also industrial plant sites can have their own fire brigade. The emergency response layer is the last layer of mitigation available.

Besides these protection layers there are of course many other issues that influence the safety aspects of an industrial plant. Under all circumstances events or conditions should be prevented that can jeopardize the safety of people and environment. The SPLP is the most modern way of dealing with safety in practice. The actual implementation and maintenance of the SPLP brings out other important issues, whose contribution to a safer plant should not be underestimated. These issues are plant management, training of personnel, maintenance and repair, quality assurance, compliance with national and international standards, periodic process hazard analysis, and so on.

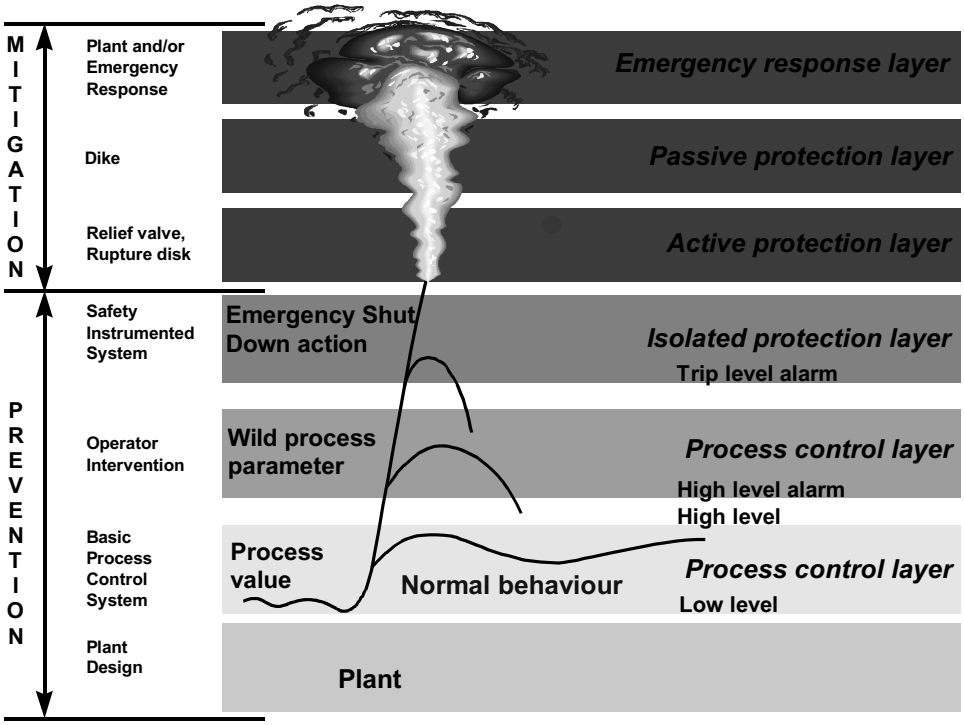

Figure 4. Safety Protection Layer Philosophy

The following sections will introduce the readers to an example that is used throughout this thesis. This example represents a typical safety case that exists in industry. It is based on a manufacturing process that currently exists in industry. The focus will be on the prevention layers, and in particular design of the plant and the safety instrumented system.

3.3 Example: PETN manufacturing plant

The purpose of the manufacturing plant is to mix nitrate acid with pentaerythritol to create in a safe manner pentaerythritol tetranitrate (PETN) of the desired quality. PETN is produced by the nitration of pentaerythritol in a batch process. The main process equipment, necessary to facilitate this mixing process, is shown in Figure 5. In the chemical plant there are actually two nitrators next to each other. Both nitrators can be operational at the same time.

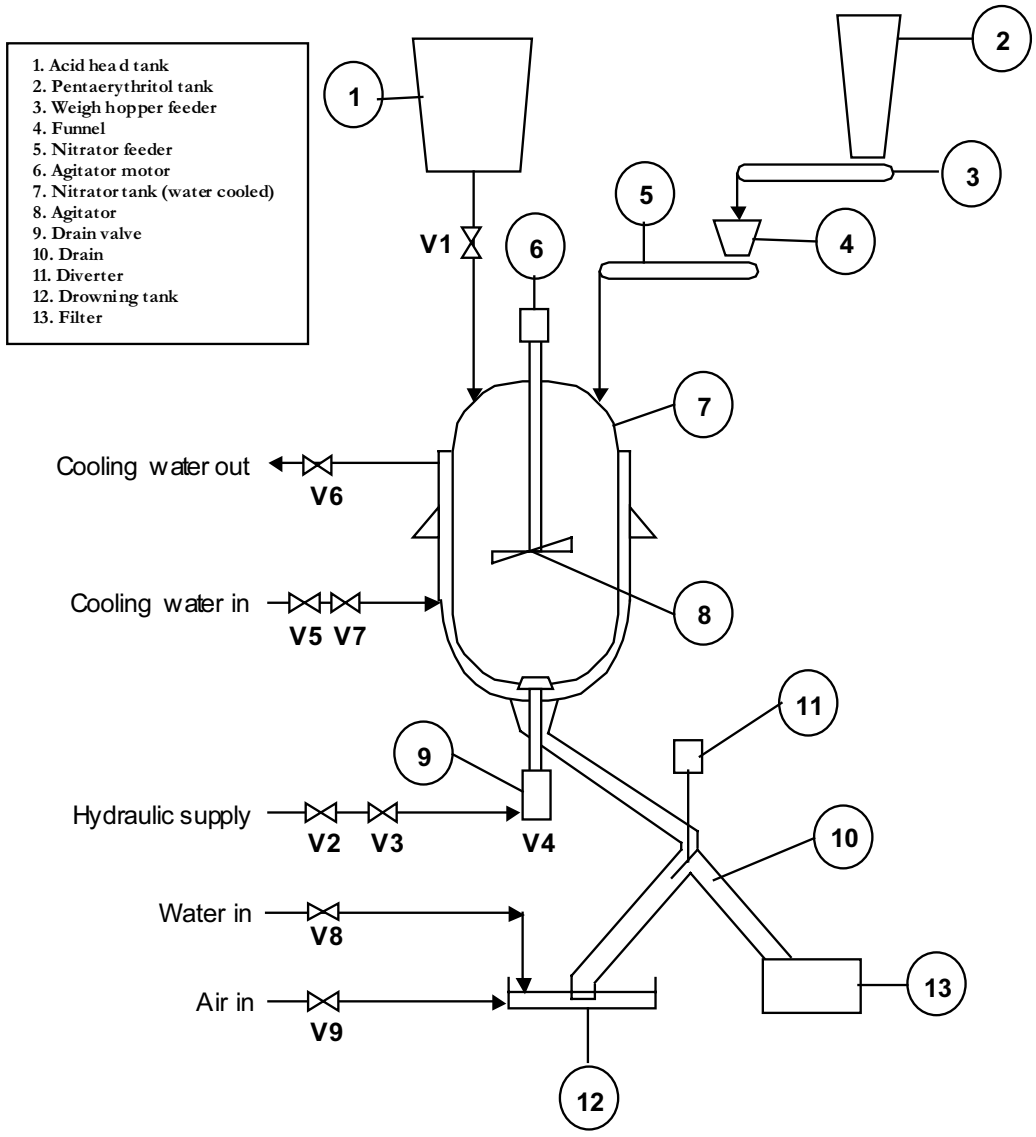

Figure 5. Schematic diagram of nitrator [58]

The nitrating acid and PE is mixed in the nitrator tank (7). Predetermined quantities of acid are stored in the acid head tank (1), and released when necessary into the nitrator tank. This release of acid is nearly instantaneous. The PE is stored in the PE tank (2) and released onto the weigh hopper (3,4) until the desired quantity is determined. In a controlled manner the PE is then fed, via the nitrator feeder (5), into the nitrator tank and mixed with the acid. The feeder rate is controlled in such a fashion so that the contents of the vessel are maintained at a predetermined temperature. An exothermic reaction takes place must be controlled. A too high rise in temperature must be prevented as it results in the decomposition of the PETN in flammable and toxic fumes. If the exothermic reaction is not controlled, there is a risk of fire, with potential exposure to the finished explosive PETN in the plant. There fore, if the temperature in the nitrator tank arises above the critical 35 °C, the content of the nitrator tank is dumped into the drowning tank. Because the temperature is a critical parameter, the nitrator tank is also cooled with water that is circulated around the exterior surface of the vessel. The contents of the vessel are continuously stirred with an agitator (6), and after a preset time, and only when the temperature has

reached the desirable level, the contents are discharged through a single hydraulically operated drain valve (9). The valve opens in a de-energized-to-open mode and releases the PETN to a drain (10). A diverter (11) in the drain is used either to guide the PETN to the drowning tank (12) or to the filter (13). The drowning tank contains a large volume of water. When material is released into the drowning tank, compressed air is fed into the bottom of the tank to agitate the contents and stimulate mixing to keep the material temperature below 35 °C.

Under no circumstances should PETN be dumped, as part of an emergency shutdown routine, while the diverter is positioned towards the filter. This will expose possible flammable and toxic fumes to other areas in the plant leading to an additional hazardous situation. In order to dump the material in the drowning tank it first has to be checked whether the diverter is in the correct position, i.e., towards the drowning tank and not the filter. It is possible that the diverter is in the filter position as the second nitrator might be dumping successfully mixed PETN to the filter. In case of an emergency dump situation, it might be necessary to reset the diverter to the default position, even if this means loss of the successfully mixed PETN.

Besides the design of the manufacturing equipment, two additional layers of protection exist in the plant. First of all there is the basic process control system (BPCS). Second, there is the programmable electronic safety system, in the process industry also referred to as safety instrumented systems. The BPCS controls the batch process and other processes in the plant. The safety system carries out specific designed safety functions. As the name implies the purpose of the BPCS is control and not safety. The BPCS is mainly quality related (it monitors the process to assure the production of the product) but if designed correctly it adds additional safety to the overall process. The focus in the thesis is on the design of the process itself and the programmable electronic safety system. The design and verification of the logic solver and the application logic (software) of the safety system is included, as the software plays a major role in the implementation of safety. The principle theories and techniques presented in this work are general in nature and can easily be applied to the BPCS as well.

3.3.1 Overview instrumentation and logic solvers

An overview of the BPCS instrumentation and process equipment of the batch process is given in Figure 6 and Table 2. The BPCS receives information throughout the plant regarding process parameters such as temperature, pressure, positions, speeds, and weights. The BPCS receives field signals through sensors, performs control logic and updates the field through the actuators. The BPCS itself can trigger alarms signals. Not all signals are critical and lead to an immediate shutdown of the process. Some signals will issue a warning, which can be overruled, if visual inspection determines an instrumentation failure, for example, low water level in the drowning tank. If the level in the tank is correct but an instrument failure issues a low water level alarm then this alarm can be overridden by operator verification after visual inspection.

Hydraulic pressure is necessary to close the drain valve at the bottom of the nitrator. The hydraulic pressure at the drain valve is only maintained if the two hydraulic valves are open and if hydraulic pressure is available. If any of the two valves is closed, the hydraulic pressure that closes the drain valve will be lost and the

drain valve will be de-energized to open and release the contents of the nitrator tank to the drain.

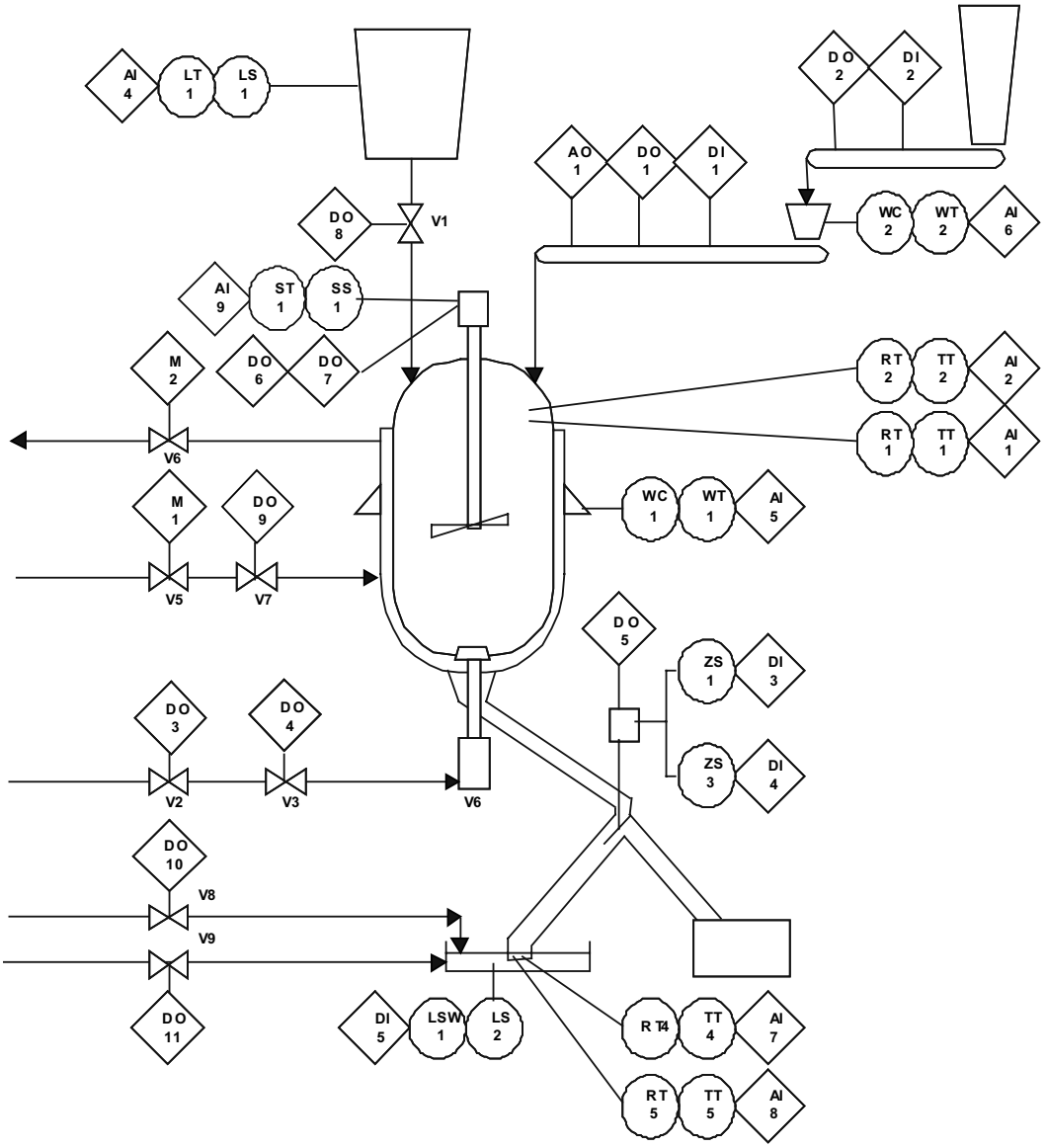

Figure 6. Diagram of Nitrator with BPCS Instrumentation and Signals

Table 2. BPCS Instrumentation and Related Equipment

Symbol	Description	Function	Signal source	Signal destination
LS1	Level sensor 1	Level, acid tank	Acid tank	Level transmitter
LS2	Level sensor 2	Level, drowning tank	Drowning tank	Level switch
LSW1	Level switch 1	Level, drowning tank	Level sensor	Digital Input
LT1	Level transmitter 1	Level, acid tank	Level sensor	Analog input
M1	Manual 1	Position, valve 5	Operator	Valve 5
M2	Manual 2	Position, valve 6	Operator	Valve 6
RT1	Resistance thermometer 1	Temperature, Nitrator tank	Nitrator tank	Temperature transmitter
RT2	Resistance thermometer 2	Temperature, Nitrator tank	Nitrator tank	Temperature transmitter
RT4	Resistance thermometer 4	Temperature, drowning tank	Drowning tank	Temperature transmitter
RT5	Resistance thermometer 5	Temperature, drowning tank	Drowning tank	Temperature transmitter
SS1	Speed sensor 1	Speed, agitator	Agitator motor	Speed transmitter
ST1	Speed transmitter 1	Speed, agitator	Speed sensor	Analog input
TT1	Temperature transmitter 1	Temperature, Nitrator tank	Temperature sensor	Analog input
TT2	Temperature transmitter 2	Temperature, Nitrator tank	Temperature sensor	Analog input
TT4	Temperature transmitter 4	Temperature, drowning tank	Temperature sensor	Analog input
TT5	Temperature transmitter 5	Temperature, drowning tank	Temperature sensor	Analog input
WC1	Weigh cell 1	Weight, Nitrator tank	Nitrator tank	Weigh transmitter
WC2	Weigh cell 2	Weight, funnel	Weigh funnel	Weigh transmitter
WT1	Weigh transmitter 1	Weight, Nitrator	Weigh sensor	Analog input
WT2	Weigh transmitter 2	Weight, funnel	Weigh sensor	Analog input
ZS1	Position switch 1	Position, diverter	Position sensor	Digital Input
ZS3	Position switch 3	Position, diverter	Position sensor	Digital Input

Table 3. Process control system signals

Signal type	Associated function
AI1 = Analog input 1	Temperature in nitrator tank, which is not allowed to reach 35 ºC
AI2 = Analog input 2	Temperature in nitrator tank, which is not allowed to reach 35 ºC
AI4 = Analog input 4	Level acid head tank. The level in the acid head tank determines the amount of acid that will be released to the nitrator tank. A predetermined quantity will be released when the nitrator tank is empty.
AI5 = Analog input 5	Weight nitrator of nitrator tank.
AI6 = Analog input 6	Weight of PE to be added to the nitrator tank.
AI7 = Analog input 7	Temperature drowning tank. This temperature is monitored if the contents of the nitrator is dumped into the drowning tank and is not allowed to reach 35 ºC.
AI8 = Analog input 8	Temperature drowning tank. This temperature is monitored if the contents of the nitrator is dumped into the drowning tank and is not allowed to reach 35 ºC.
AI9 = Analog input 9	Speed agitator. Failure to mix will result in local concentrations of PE that may give rise to localized rapid evolution of heat and a localized or bulk temperature rise.
AO1 = Analog output 1	Speed nitrator feeder. Determines the speed of the nitrator feeder and thus the rate with which the PE is fed into the nitrator. The speed should not be too high to prevent local high PE concentrations
DI1 = Digital input 1	Nitrator feeder. Determines whether the nitrator feeder is switched on.
DI2 = Digital input 2	Speed of weigh hopper. Determines whether the weigh hopper feeder is in a on or off position. The weigh hopper feeder speed is constant.
DI3 = Digital input 3	Position diverter Drowning Tank. Reads the position of the diverter whether it is towards the drowning tank.
DI4 = Digital input 4	Position diverter Filter. Reads the position of the diverter whether it is towards the Filter.
DI5 = Digital input 5	Level drowning tank. If the level of the drowning tank is low then an alarm will sound.
DO1 = Digital output 1	Speed nitrator feeder. Switches the nitrator feeder on or off.
DO2 = Digital output 2	Speed weigh hopper. Switches the weigh hopper feeder on or off.
DO3 = Digital output 3	Positions valve 2. Opens or closes valve 2. The process control system opens valve 2 if the product is ready to be processed in the filter.
DO4 = Digital output 4	Positions valve 3. Opens or closes valve 3. The process control system opens valve 3 if the product is ready to be processed further in the filter.
DO5 = Digital output 5	Positions diverter. Switches diverter to the drowning tank or the filter. The diverter is switched to the filter if the content of the nitrator is ready for further processing. The diverter is switched to the drowning tank if the temperature in the nitrator exceeds 35 ºC.
DO6 = Digital output 6	Speed agitator. Switches speed agitator to low.

Signal type	Associated function
DO7 = Digital output 7	Speed agitator. Switches speed agitator to high.
DO8 = Digital output 8	Positions valve 1. Opens or closes valve 1. Valve 1 is opened if the level in the acid head tank is correct and the next batch to produce PETN is started.
DO9 = Digital output 9	Positions valve 7. Opens or closes valve 7. Valve 7 controls the flow of cooling water during normal operation and is opened when the batch process is started.
DO10 = Digital output 10	Positions valve 8. Opens or closes valve 8. If the level in the drowning tank is to low then valve 8 is opened to add water.
DO11 = Digital output 11	Positions valve 9. Opens or closes valve 9. Valve 9 is opened when the contents of the nitrator tank are dumped to the drowning tank. In this way, the water in the drowning tank is supplied with air to stimulate the mixing of the dumped material with the water in the drowning tank.

3.3.2 Safety System

The safety system serves as last layer of protection against accident scenarios that can arise from the operation of the process. When the safety system fails it is not possible anymore to prevent in a controlled manner an accident. To prevent accidents the safety system carries out specific identified safety functions. It reads inputs from the field (process parameters) via sensors. The programmable electronic logic solver uses these inputs to execute the application software designed by the process safety engineers. If necessary the application logic will actuate field devices like valves. A correctly designed safety system is supposed to work autonomously, i.e., it works independent of operators, the BPCS or other support functions.

The main safety function executed by the safety system implemented at the manufacturing plant is focused on the temperature in the nitrator tank. The safety system is supposed to dump the mixed material in the drowning tank by opening the drain valve if the temperature exceeds the limit. If necessary the safety system has to switch the divider to the drowning tank before dumping the material.

The actual application logic necessary to solve the above presented safety function is programmed in the logic solver. A high level version of the logic solver is presented in Figure 7. It consists of three distinct modules, i.e., the input module, the main processor module, and the output module. The input module has several input channels that read signals from the field. The input module communicates with the main processor module via bus communication. The main processor module carries out the safety logic using application software that is off-line created by an operator. The operator can upload the application software via a human-machine interface software into the main processor module. The main processor communicates with the output module via bus communication. The output board has several output channels that actuate field devices.

The safety function carried out by the safety system requires three input signals and one output signal. The required signals are listed in Table 4. One temperature sensor signals the temperature in the tank. The safety system also retrieves the signals from the diverter, but signals the BPCS if there is a need to switch the diverter. The only safety function output signal is towards the valve that controls hydraulic pressure supply to the drain valve.

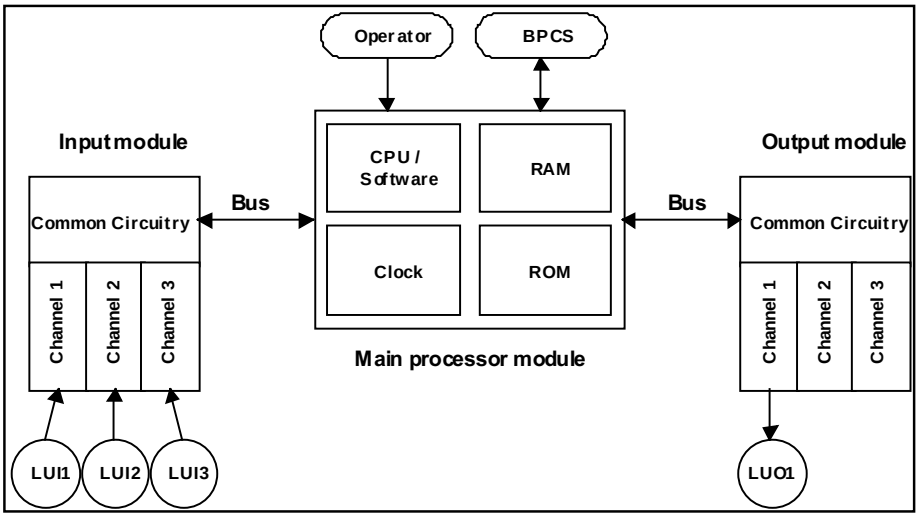

Figure 7. Logic Solver (excluding field devices and related equipment)

Table 4. Description of safety system input and output signals

Signal	Description	Possible states
LUI1	Logic Unit Input signal 1: Temperature switch in the tank.	[Ok, Too high]
LUI2	Logic Unit Input signal 2: Diverter drowning tank switch.	[Drowning Tank, Off]
LUI3	Logic Unit Input signal 3: Diverter filter switch	[Filter, Off]
LUO1	Logic Unit output signal 1: Drain valve	[Open, Close]

3.4 Requirements for the analysis method

If a process like this needs to be designed, or if the design needs to be verified by an independent party, then a method needs to be available that supports the analysis of the different layers and the interaction between the layers. The method also needs to support the integration of safety and quality as described in Chapter 1. From the information derived from Chapter 1 and Chapter 3, the following requirements are described:

- The method should take into account all possible parameters of interest that can exist in a process plant including hardware, software, physical parameters, human parameters and other parameters of interest.
- The method should take into account all desired and undesired deviations of these parameters.
- The method should take into account the functional interaction between the above parameters.
- The method should take into account the dynamic interaction between the above parameters.

- The method should be able to analyze root causes of deviations.
- The method should be able to analyze how deviations can propagate through the process.
- The method should be able to prioritize these deviations so that informed decisions about safety and quality can be made.
- The method should take into account the total life of the process, not only the design but also the operation, maintenance, and repair of the process.

In Chapter 4 the dynamic flowgraph methodology is introduced to support the method as a tool that is capable to carry out most of the above identified requirements. Chapter 5 introduces the importance measures to further support the analysis in terms of prioritization.

3.5 Conclusions

This chapter introduced the reader to the technical framework that is used in industry to manage safety in modern manufacturing or processing plants. The chapter described a typical manufacturing plant the way they currently exist in industry. The example contains all the complexity to demonstrate the technical knowledge required for designing or verifying a process for safety. As safety requires a total systems approach it is necessary to understand the interaction between any parameter within one layer of protection and between the different safety layers. In terms of safety, the interaction between the layers of protection should be interference free, i.e., a change, desired or undesired, of a parameter should not result in an unsafe process condition. This chapter summarized the requirements for a safety method that can analyze the complex technical aspects that exist in modern manufacturing plants in terms of deviations.

Chapter 4 Dynamic Flowgraph Methodology

4.1 Introduction

The purpose of this chapter is to introduce the reader to the dynamic flowgraph methodology (DFM). DFM is used to model and analyze a (manufacturing) process. The results of a DFM analysis serve as basis for the method outlined in this thesis to carry out process hazard analysis and process safety management. This chapter explains the basic concepts of the dynamic flowgraph methodology. It will also explain the background of DFM, what a DFM model is, how a DFM model is created, and how it can be used to analyze a system. The chapter will end with an overview of the advantages and disadvantages that DFM has compared to traditional safety analysis tools.

4.2 Background

The dynamic flowgraph methodology arose from the logic flowgraph methodology, which was developed in the early eighties [60]. The logic flowgraph methodology (LFM) originated as a new tool to be useful in reliability and risk analysis applications. In the beginning, LFM was mainly used for applications in the nuclear industry [61,62,63]. Later, other application areas were investigated, i.e., aerospace systems [64] and aerospace embedded systems [65]. Over the years, LFM had proven to be effective, as the underlying methodology in process failure diagnosis, and decision support systems. Research and further efforts for the improvement of the LFM concept resulted in a modeling and analysis approach eventually named the dynamic flowgraph methodology (DFM) [66,67]. The major improvement that DFM added to the LFM concept was the capability to analyze, besides logical behavior elements, also time-dependent aspects within a system.

The DFM approach is very general in nature and can model the logical and dynamic behavior of complex systems, including such elements as hardware, software and human actions. DFM models the relationships between important process parameters because of cause-and-effect and timing functions inherent to the system. If this system is a programmable electronic system (PES), i.e., a system where mechanical devices and physical parameters are controlled and operated by software, then both the physical system, as well as the software controlling the system can be taken into account by the DFM system model. As such, DFM is a useful methodology capable to analyze and test hardware and/or software related systems [66,68,69].

4.3 The DFM model

A DFM model of a system is presented as a directed digraph representing the logical and dynamic behavior of the system in terms of important system parameters (physical, software, human interaction, or any other parameter). A digraph model explicitly identifies the cause-and-effect and timing relationships that exist between key parameters and system states that are suited to describe the system behavior [67]. The DFM system model extends a normal digraph model because it represents an integration of three networks, i.e., a “causality network”, a “conditioning network”,

and a “time-transition network". DFM uses a set of basic modeling elements to represent the system parameters and their relationships in terms of these three networks. The possible modeling elements are (see Figure 8):

1. Variable and condition nodes;
2. Causality and condition edges; and
3. Transfer and transition boxes and their associated decision tables.

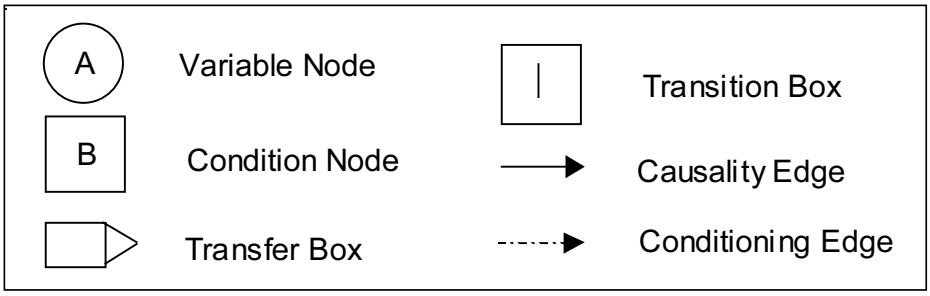

Figure 8. DFM Modeling Elements

The nodes of the DFM system model represent important system components, parameters, or variables and their normal and abnormal functional conditions. Nodes are discretized into a finite number of states that represent the parameter best. This discretization can represent much more than just success or failure, or on/off situations. They can represent, for example, a temperature range, or possible representative values of a software variable, e.g., an integer or a real variable. The transfer and transition boxes represent the relationship that exist between parameters. Each box has an associated decision table that is used to incorporate a multi-state representation of the cause-and-effect and timing relationships that can exist among the connecting parameters. The two different edges are only used to visually represent the kind of relationships that exists between parameters, i.e., a cause-and-effect or conditioning relationship.

For example, the nitrator section, as described in section 3.2, consists of a drain valve that is operated by hydraulic power. The DFM model of the drain valve and the hydraulic supply is presented in Figure 9. This small DFM model consists of seven variable nodes (DO3, SV2, HP1, DO4, HP2, HP3, and PDV), three conditioning nodes (CV2, CV3, and CDV), and four transition boxes (T). The explanation of the different nodes is given in Table 5. The discretization of the different nodes is given in Table 6 through Table 12. The DFM model is created, starting with the actual physical position of the drain valve (PDV), which can be open or close. From this starting point, the question is then asked what situations lead to the possible positions of the drain valve. In this case, the position of the drain valve is determined by the actual condition of the drain valve and the availability of hydraulic pressure. The model is worked out further for each process variable taking into account the desired level of detail. The model is developed to a point where no further information is needed or further information is not available.

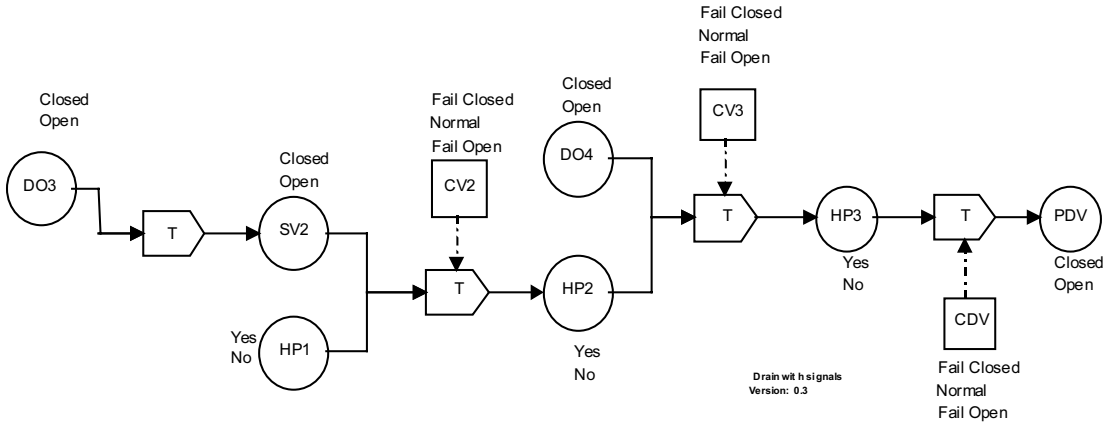

Figure 9. DFM model drain valve

Table 5. Overview nodes

Parameter	Description	Type
DO3	Digital Output 3	Variable node
DO4	Digital Output 4	Variable node
SV2	Signal Valve 2	Variable node
HP1	Hydraulic pressure position 1	Variable node
HP2	Hydraulic pressure position 2	Variable node
HP3	Hydraulic pressure position 3	Variable node
PDV	Position Drain Valve	Variable node
CV2	Condition valve 2	Condition node
CV3	Condition valve 3	Condition node
CDV	Condition Drain Valve	Condition node

Table 6. Discretization of DO3

States	Description
Open	Open valve
Close	Close valve

Table 7. Discretization of DO4

States	Description
Open	Open valve
Close	Close valve

Table 8. Discretization of CDV

States	Description
1	Drain valve stuck closed
2	Drain valve operates normally
3	Drain valve stuck open

Table 9. Discretization of CV2 and CV3

States	Description
1	Valve stuck closed
2	Valve operates normally
3	Valve stuck open

Table 10. Discretization of HP1, HP2, and HP3

States	Description
Yes	Hydraulic pressure available 1
No	Hydraulic pressure not available 1

Table 11. Discretization of PDV

States	Description
Closed	Drain valve closed
Open	Drain valve open

Table 12. Discretization of SV2

States	Description
Closed	Signal valve 2 is closed
Open	Signal valve 2 is open

The transfer and transition boxes represent the relationship that exists between the different parameters. Each box has an associated decision table that is used to incorporate a multi-state representation of the cause-and-effect and timing relationships that can exists among the connecting parameters. The DFM model in Figure 9 has four transition boxes. The actual position of the drain valve, open or close, depends on the condition of the drain valve (i.e., failed or not failed) and whether there is hydraulic pressure or not. The decision table that represents this relationship is presented in Table 13. The drain valve is closed, if the condition of the drain valve is normal and if hydraulic pressure is available, or if the drain valve is

failed closed (i.e., the condition of the drain valve is stuck closed). The drain valve is open, if the condition of the valve is normal and there is no hydraulic pressure or if the drain valve is failed open.

Whether hydraulic pressure exists to close the drain valve depends on the condition and position of valve 2 and 3, and whether hydraulic pressure before these valves exists. This relationship is captured in Table 14 and Table 15. Hydraulic pressure to close the drain valve only exists when hydraulic pressure is available before valve 3, the condition of valve 3 is normal and the position of valve 3 is open. The same relationship exists between the hydraulic pressure before and after valve 2. When valve 2 and 3 operate normal, it is possible to open and close them with a signal from the BPCS (SV2 and SV3). The operation of valve 2 and 3 is thus controlled by software. The application software of the BPCS is not further considered in this thesis. If it would be considered it would be possible to extend the DFM model with the relationship between the hardware and the BPCS application software. When the SIS is modeled it will be demonstrate how hardware and software structures are modeled.

Table 13. Transition table (HP3, CDV, PDV)

Input		Output
HP3	CDV	PDV
Yes	Normal	Close
No	Normal	Open
-	Close	Close
-	Open	Open

Table 14. Transition table (HP1, SV2, CV2, HP2)

Input			Output
HP1	SV2	CV2	HP2
-	Close	Normal	No
Yes	Open	Normal	Yes
Yes	-	Open	Yes
No	-	-	No
-	-	Close	No

Table 15. Transition table (DO4, HP2, CV3, HP3)

Input			Output
DO4	HP2	CV3	HP3
Close	-	Normal	No
Open	Yes	Normal	Yes
-	Yes	Open	Yes
-	-	Close	No
	No	-	No

To demonstrate the use of a time transition box an additional DFM model of a drowning tank is presented (see Figure 10). It is necessary to model the drowning tank with a time transition box as it takes a certain time to fill the tank with water. This time depends on factors like:

- The in-flow rate of water;
- The outflow rate of water;
- The volume of the water tank;
- The condition (i.e., is there leakage or not) of the tank.

In this case, a simplified model is used where the level of the tank depends on the inflow rate and the current level of water in the tank. The level of the water tank is discretized into three levels, i.e., low, normal, high. The modeler has to determine how much time it takes to go from one level to the next level in the tank. This time is referred to as the time step. If more then one time transition box is used the time step is the smallest time constant in the system. The level of water in the drowning tank depends on whether water is going into the drowning tank and what the level of water was in the previous time step. Further explanation on time management with DFM is given in section 4.4.4.

The different parameters involved in Figure 10 are listed in Table 16. The discretization of the parameters is presented in Table 17 through Table 20. The decision table associated with transition box T1 and time transition box T2 is presented respectively in Table 21 and Table 22. In this way, it is possible to model any relationship that exists between parameters of interest.

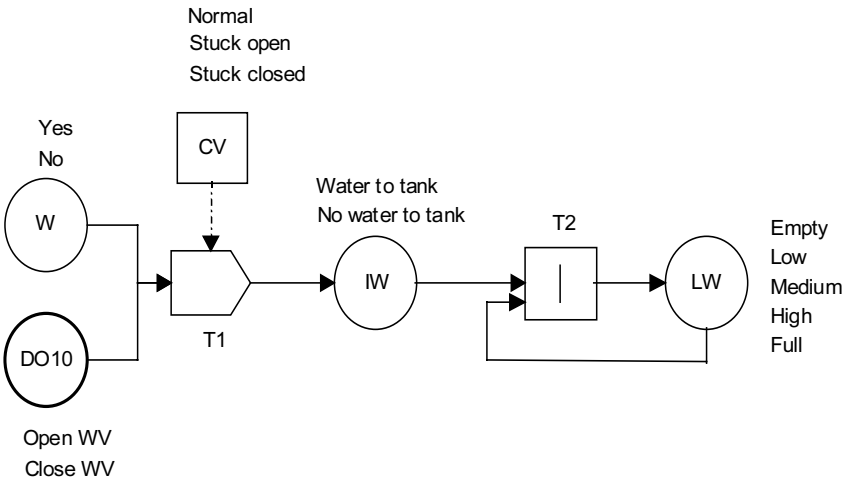

Figure 10. DFM model water tank example

Table 16. Parameters DFM model

Parameter	Description
CV	Condition water supply valve
IW	Intermediate water
LW	Level water
W	Water
DO10	Digital Output Signal 10

Table 17. Discretization of CV

States	Description
Open	Water supply valve failed open
Normal	Water supply valve operates normally
Closed	Water supply valve failed closed

Table 18. Discretization of IW

States	Description
No	No water flow to tank
Yes	Water flow to tank

Table 19. Discretization of LW

States	Description
Low	Level drowning tank low
Normal	Level drowning tank normally
High	Level drowning tank high

Table 20. Discretization of W

States	Description
No	No water available
Yes	Water available

Table 21. Transition table (DO10, W, CWV, IW)

Input			Output
DO10	W	CWV	IW
Open	Yes	Normal	Yes
Closed	Yes	Normal	No
-	No	-	No
-	-	Closed	No
-	Yes	Open	Yes

Table 22. Time transition table (IW, LW, LW)

Input		Output
IW	LW	LW
Yes	Low	Normal
Yes	Normal	High
Yes	High	High
No	Low	Low
No	Normal	Normal
No	High	High

It doesn't matter whether the DFM model parameters represent process variables, hardware conditions, signals or human action or interference. As long as the analyst is capable of capturing the relationship in a decision table, it is possible to model this relationship with DFM. For reliability and safety analysis, DFM meant a major step forward because it was possible to analyze in one model hardware, software, environmental, and human interaction. Although in this example the

application software of the BPCS is not further modeled, the influence of software states (the input and output signals) on the behavior of the batch process is clearly present. The accuracy or quality, in case of complex physical relationships, depends on the level of discretization of the values of the system parameters. The more detail is put into the discretization of the parameters the more detail can be put in the decision tables. On the other hand the more complex the model is made in terms of detail the easier it is for a modeler to make mistakes. Developing a very detailed DFM model requires good verification and validation procedures.

4.4 Modeling and analyses

Once the DFM model is available it is possible to analyze the model. The DFM model is created independently of the analyses of interest. This means that the model is a very comprehensive model. With a DFM model it is possible to carry out a multitude of analyses of interest. The DFM model contains every system feature that determines the possible desirable and undesirable behavior of the system. This is one of the key features that distinguish DFM from other analysis methods, which usually are only focused on modeling undesirable behavior.

In the analysis phase two different approaches can be utilized, i.e., deductive analysis and inductive analysis. Event sequences can be traced backward from effects to causes, or forward from causes to effects. The DFM analysis engine contains sophisticated software algorithms that have been developed to support automated deductive and inductive analysis. Being able to combine inductive and deductive analysis in one methodology makes DFM a very powerful tool to analyze systems within the context of design verification, failure analysis and/or automatic test sequence and vector generation.

4.4.1 Deductive analysis

A deductive analysis of a DFM system model starts with the identification of a particular system condition of interest, depending on the objective of the analysis. This system condition, or top event, can contain system states that represent failure, success or a combination of both. The objective of DFM's deductive analysis is to find the root causes of the top event of interest, just as traditional Fault Tree Analysis identifies the basic events causing the predefined top event. A DFM top event is expressed in terms of the state(s) of one or more process variable nodes. To find the root causes of the top event the model is analyzed by backtracking through the network of nodes, edges, transfer and transition boxes of the DFM system model, using a specially developed analytical software algorithm. This automated backtracking algorithm contains the procedure to work backward in a cause-and-effect flow to identify the paths and conditions by which the top event may be originated. Because of this backtracking algorithm, the analysis identifies what the causes or conditions are at the root of the top event. These causes are expressed in terms of combinations of process variable and condition states and are similar to cut sets in fault tree analysis.

In this thesis the main focus is on deductive analysis. When deductive analysis is used, DFM generates so-called prime implicants [69]. A prime implicant consists of a set of variables of interest that are in a certain state at a certain time that causes a predetermined system state. Prime implicants are similar to minimal cut sets known

from fault tree analysis. They differ in the sense that they can contain normal or non-failed states, failed states, or any other state of interest. Each variable state is also associated with a time, stating when this prime implicant variable needs to be in this state. Each variable with its properties is called a literal. The following is an example of a prime implicant:

```
Prime Implicant #XYZ
At time -3,   CNF = High    (Stuck high)        AND (Literal)
At time -3,    BW = Wide    (Wide)              AND (Literal)
At time -3,   CAM = Normal  (Operates normally) AND (Literal)
At time -3,  PETN = 1 %     (1 %)               AND (Literal)
At time -3,    TT = Low-low (Low-low)           AND (Literal)
At time -2,   CAM = Normal  (Operates normally) AND (Literal)
At time -2,  PETN = 2 %     (2 %)               AND (Literal)
At time -1,    M1 = Open    (Manual Open V1)    AND (Literal)
At time -1,   DO6 = Off     (No speed)          AND (Literal)
At time -1,   DO7 = Off     (Speed off)         AND (Literal)
At time -1,   CAM = Normal  (Operates normally)     (Literal)
```

There are three properties of interest for each literal, i.e., the time step “-3”, the variable name “CNF”, the state of the variable “High” (meaning “Stuck High”). The “AND” at the end of the line represents the Boolean relationship that exists between the literals. Each literal needs to be true in order for the prime implicant to be true and thus for the top event to happen. The number of prime implicants depends on the complexity, size, and required level of detail of the system to be modeled. Because of the time dependency, the number of literals is also strongly correlated with the total analysis time.

As it can be seen from this example, a prime implicant can contain a wealth of information. Prime implicant #XYZ is a mixture of software signals, hardware states and process parameters. This makes it much more valuable then the results from any other existing reliability or safety methodologies like, e.g., FMEA or FTA. The prime implicants are a very useful resource for risk management and they should be explored for that purpose. The prime implicants form the basis of the risk management tool proposed in the thesis. Later in this thesis importance measures are introduced that can be used to automate the analysis of the prime implicants for the purpose of risk management, as the number of prime implicants can be enormous.

4.4.2 Inductive analysis

Inductive analysis follows a bottom-up approach by introducing a set of component states and analyzes how this particular set of interest propagates through the system and what the effect will be on a system state level of interest. Inductive analysis follows the principles of fault injection and is useful to examine the consequences of hazards on system level. Once an initial set of conditions is defined, inductive analysis is used to trace forward in the system the events that can occur from this starting condition. The initial conditions and boundary conditions can be defined to represent desired and undesired states. Starting from a combination of desired states, an inductive analysis can be used to verify whether the system meets its

design requirements. Starting with undesired states, inductive analysis can be used to verify the safety behavior of the system. To demonstrate the usefulness of DFM as a lifecycle management tool, inductive analysis will be used for design verification.

4.4.3 Consistency rules

The DFM software tool has the capability to model or apply dynamic consistency rules. Consistency rules are very useful to model specific situations of interest as they can model a situation or condition more accurately. They are defined in terms of allowable variations of parameter values across different time steps. There are two types of rules:

1. The state of a parameter cannot change in a certain direction between two time steps.
2. A parameter cannot change by more than a certain amount of states between time steps

Rules of the first type can be defined from the analyst's knowledge about the dynamic constraints of the system or from modeling assumptions. For example, once a valve has failed stuck open, it stays in the open position. Another example is the concentration of PETN in the nitrator tank. It can only stay constant or increase over time. For certain parameters it is not possible to change direction.

Rules of the second type come from the rules of the system. For instance, rules of the second type can state the position of the valve cannot vary by more than two states in one time steps, as it takes a finite amount of time for the valve to open or close. The consistency rule would limit the analysis to the amount of steps the parameters can change within a time step.

Actually, there is an additional rule that needs to be taken into account. The analyst applies this rule while building the model. This rule dictates that several parameters must vary in a specific way between time steps. For example, a valve and its flow rate must vary in a proportional manner, as required by physical law. It is not specified as an additional consistency rule as it needs to be taken into account by the decision table associated with the time transition box.

4.4.4 Time management

The capability of the time-transition box makes DFM a very powerful tool. Depending on the complexity of the system subject to the analysis it might be though that more then one time constant applies. Take for example the PETN manufacturing process. There are different sections in the plant that have different time constants. The plant could be divided into the following sections taking into account the different time constants of:

- The feeder section;
- The mixing process;
- The drowning tank;
- The basic process control system;
- The safety system.

The feeder section might have a time constant of several minutes, while the mixing process might take a few hours. Filling the drowning tank with water might take a day, while the basic process control system and the safety system might have a time constant from a couple of milliseconds to a few seconds. The problem with the time constant is that the DFM tool currently can only analyze a DFM model using one time constant. This means that if one model is created for the total batch process, the smallest time constant must be chosen that applies to all sections. This would mean that if one would make a very comprehensive model of a manufacturing plant including a programmable electronic system that the smallest time constant could be in the millisecond range, while the longest time constant would be for example a day. In terms of deductive analysis this would mean that one has to go back in time millions of time steps in order to see the influence of all parameters of the plant. Although theoretically not impossible, practically it cannot be done with the current calculation capacity of the software tool and computer systems. The challenge is in the analysis phase. The number of prime implicant would be too exhaustive to analyze.

There is an easy solution to overcome this problem. Each section with its own time constant can be modeled and analyzed individually. The only hurdle to overcome is to manually connect the different models in terms or results. If it is possible to create only one model then also only one top event is necessary to analyze all the aspects of the model. If the model is split in sub models, for example, one model for the plant and one for the safety system, then it can be that the results of the analyses of one top event are required to define one or more new top events for a sub section. In the worst case the results of the analysis of a sub section might be used as feedback into the first sub section. Unfortunately this all needs to be accomplished manually. Future versions of DFM will do also this automatically.

4.5 DFM model of the PETN manufacturing process

A complete DFM model of the equipment under control of the manufacturing process is presented in Figure 11. The model includes the following sections

- Supply of material;
- Mixing of material;
- Cooling system and process of the mixing tank;
- Drain valve operation along with the hydraulic supply system used for the operation of the valve;
- Drain with diverter;
- Drowning tank.

Although the actual characteristics of the chemical reaction in the tank are beyond the scope of this paper, a simple model has been created to determine the temperature in the tank taking into account the relevant parameters. The model assumes that the temperature is influenced by:

- The mixture of acid and PE, i.e., the concentration of PETN;
- The speed of the agitator;
- The flow of cooling water around the tank;

- The mixing time;
- The position of the drain valve.

The chemical reaction is assumed to take place over a certain period and is modeled with a time transition box. There are other time constants in the system, e.g., the filling of the drowning tank or the filling of the weigh funnel with PE. It is also assumed that the time constant of the chemical reaction are smaller then other time constant of this process and that the latter are derived from the time constant of the chemical reaction. A ratio of 1 to 2 is assumed. This means that if the smallest possible temperature change in the tank takes place in 1 time step than a change in the water level or the weight in the funnel takes 2 time steps.

The temperature in the tank depends, of course, on the above-mentioned parameters and the temperature in the tank at the previous time step. It is impractical to display the decision table associated with this time transition box in this paper. It includes five input parameters (PDV, NCW, SA, PETN, TT) and one output parameter (TT). The discretization of the different parameters resulted in a decision table with 76 rows.

Table 23. Discretization of PDV

States	Description
Closed	Drain valve closed
Open	Drain valve open

Table 24. Discretization of NCW

States	Description
Pos	Positive flow through tank
No	No flow through tank

Table 25. Discretization of SA

States	Description
Zero	Agitator motor speed 0
Low	Agitator motor speed low
High	Agitator speed high

Table 26. Discretization of PETN

States	Description
1	Concentration Pentaerythritol Tetranitrate 0
2	Concentration Pentaerythritol Tetranitrate 1
3	Concentration Pentaerythritol Tetranitrate 2
4	Concentration Pentaerythritol Tetranitrate 3
5	Concentration Pentaerythritol Tetranitrate 4
6	Concentration Pentaerythritol Tetranitrate 5
7	Concentration Pentaerythritol Tetranitrate 6
8	Concentration Pentaerythritol Tetranitrate >6

Table 27. Discretization of TT

States	Description
Low-low	Temperature tank low-low
Low	Temperature tank low
Medium	Temperature tank medium
High	Temperature tank high
High-high	Temperature tank high-high

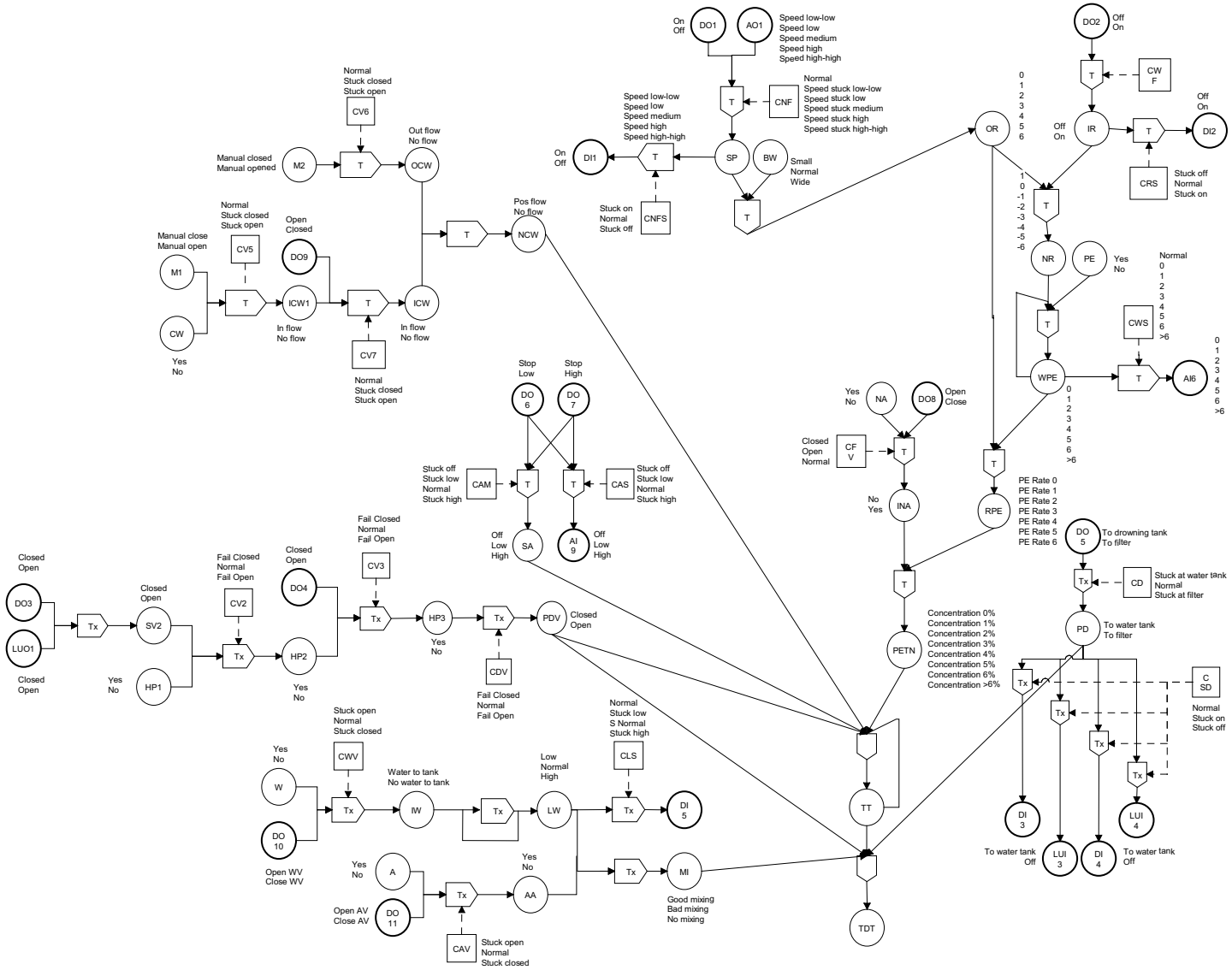

Figure 11. DFM model of the nitrator section

The temperature in the tank depends on five different parameters, which are each discretized into several states. The possible states of these five parameters are determined by the interaction of hardware, software and human action. The model is now ready to be analyzed. The analyst can investigate many different top events of interest, e.g., what can actual cause a high-high temperature in the tank.

4.6 DFM model safety system

The DFM model of the safety system is shown in Figure 7. The objective of this system is to collect information from the field, interpret this information, and decide what action to feed back to the field. This model is again created taking into account the required or necessary information flow and not only the possible failure behavior of the safety system. The undesired behavior of the system has been modeled taking into account the failure mode requirements of the IEC 61508 standard [18], see Table 44. The level of detail in this model has been chosen in a way that it reflects identifiable functional blocks that, if they fail, will fail the complete safety systems or one of the safety functions carried out by the safety system. The model of the safety system is created without any online diagnostic features.

This model is still on a high level and does not address the lowest possible individual component failures, but at first instance it allows the verification of the system structure for safety issues before the design is worked out and verified in detail. The DFM model includes the complete functional behavior of the safety system, including the interaction between hardware and software. The application software design is as well modeled by DFM. In addition, the interaction with the BPCS and Operator is modeled in a simplistic way. It is assumed that the BPCS can fail and exchange the wrong information. The operator can change the critical values and thus enter the wrong values, for example setting the limit higher then the supposed 35 °C (High-High).

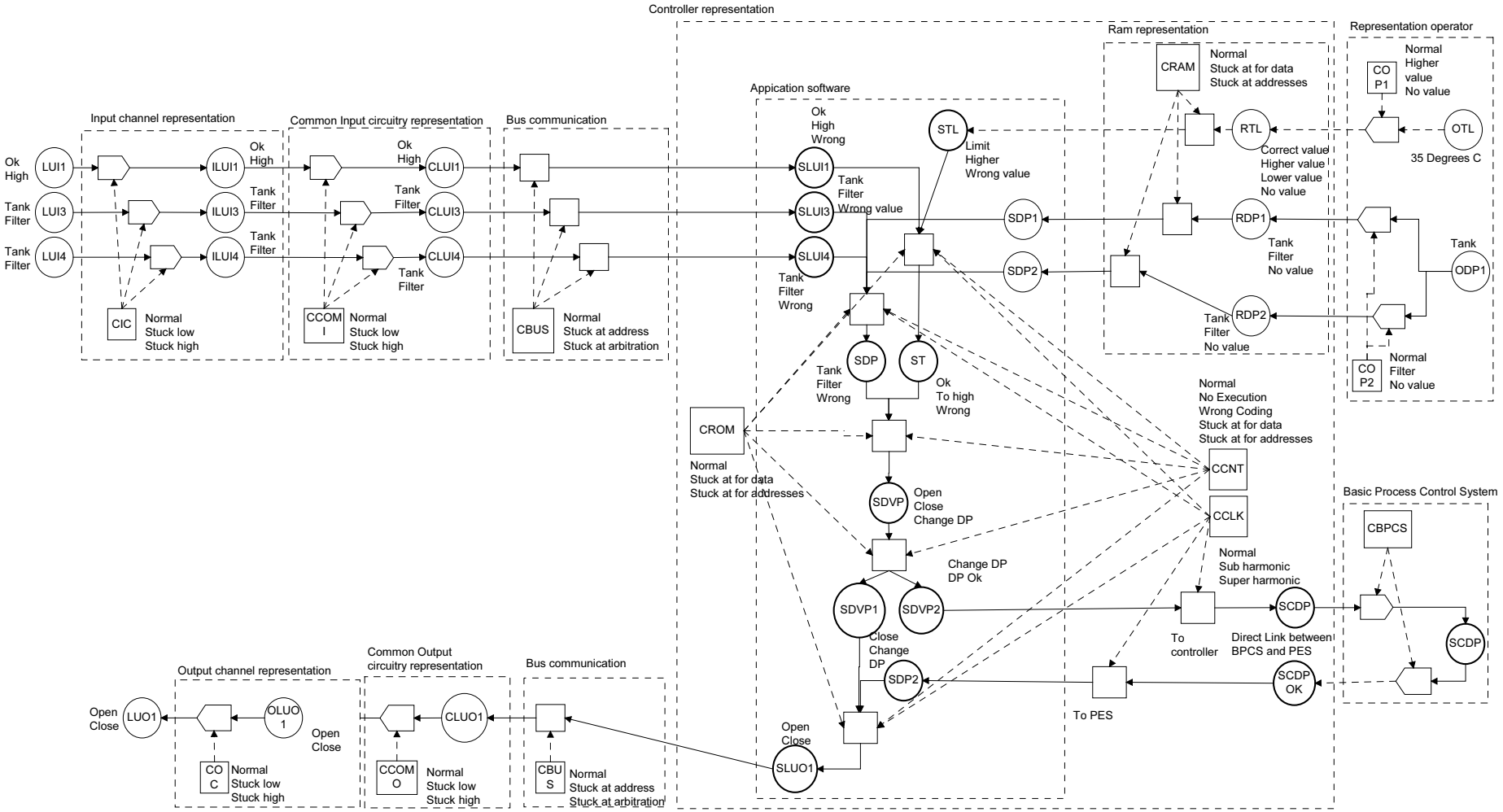

Figure 12. DFM model safety system (hardware and software) [70]

The DFM model of the safety system consists of the following sections:

- Input / Output Channels
- Common Input / Output circuitry
- Bus communication
- Controller consisting of
  - Application software
  - CPU
  - Clock
  - RAM memory
  - ROM memory
- Interface with the operator
- Interface with the basic process control system.

This model shows one of the capabilities of DFM that are worth pointing out. This model contains, hardware as well as software, and is capable of modeling the interaction between the hardware and software. As this model is eventually integrated with the DFM model of the process, the model incorporates the interaction between the physical field parameter and representation of this parameter in the software. The application software is modeled in the context of its operating environment. An example of one of the software routines is the routine that compares the measured temperature in a tank with the limit programmed by the operator and then decides whether the temperature is at the programmed limit or not. The software routine looks as follows:

```
If (SLUI < STL) THEN ST = "OK" ELSE ST = "TO_HIGH"
```

The SLUI1 variable represents a temperature that is either OK or TO_HIGH. The variable STL is the temperature limit programmed by the operator and can be at the required LIMIT or if the operator made a mistake at a HIGHER_LIMIT. A lower limit is not assumed as failure as this would be easily found during testing or normal operation. The process would shutdown even though the high temperature limit would not be achieved.

The software routine sets the output variable ST. The output variable ST has only two possible states, OK or TO_HIGH. The software routine can only be executed if the controller (CCNT), the clock (CCLK) and the ROM memory (CROM) function normally. It is assumed that any failure of these components will lead to the OK state for the ST variable, which is the worst-case assumption because the safety function basically thinks that the temperature is low enough while this might not be the case. The actual relationship, which exists between the parameters SLUI1, STL, ST, CNTL, CCLK and CROM, is represented by a mapping of the possible combination of states in a decision table (see Table 28).

Even though the example only focuses on the software routine what needs to be pointed out is that temperature representation in the software depends on the

actual mixing process that takes place in the nitrator tank. The complete process is modeled and the actual value of the software variable SLUI1 depends on the process that takes place and the equipment that is involved in measure and communicating the value up to the point where it turns into the software variable SLUI1. A lot of equipment needs to be operating correctly before the software value is set. When it comes to hardware this value depends on the temperature sensor, the temperature transmitter, the input channel, the common circuitry, bus communication, and any other equipment that makes the loop work.

The same counts for the variable STL. This software variable not only depends on the correct operation of hardware but also on what the operator programmed this variable. The operator needs to be extra careful with this variable, as it is safety critical. Special procedures should exist to program this variable.

Table 28. Decision table

INPUT					OUTPUT
SLUI1	STL	CROM	CCLK	CCNT	ST
OK	LIMIT	NORMAL	NORMAL	NORMAL	OK
OK	HIGHER_LIMIT	NORMAL	NORMAL	NORMAL	OK
TO_HIGH	LIMIT	NORMAL	NORMAL	NORMAL	LIMIT
TO_HIGH	HIGHER_LIMIT	NORMAL	NORMAL	NORMAL	OK
-[3]	-	-	-	NO EXEC	OK
-	-	-	-	WRONG CODING	OK
-	-	-	-	STUCK DATA	OK
-	-	-	-	STUCK ADDRESS	OK
-	-	-	SUB HARMONIC	-	OK
-	-	-	SUPER HARMONIC	-	OK
-	-	STUCK DATA	-	-	OK
-	-	STUCK ADDRESS	-	-	OK
SLUI1 = Software variable Logic Unit Input 1, STL= Software variable Temperature Limit, CROM = Condition ROM memory, CCLK = Condition Clock, CCNT = Condition Clock, ST = Software variable Temperature					

4.7 Advantages of DFM

DFM is introduced and used in this thesis because it has certain advantages over conventional safety and reliability methods. DFM has features that distinguish this

[3] The "-" represents "No matter what the state is". The other input variables, and not this one, determine the state of the output variable(s).

modeling approach from many other techniques and tools like failure mode and effect analysis, fault tree analysis, event tree analysis, or hazard and operability analysis. The following characteristics are pointed out:

- With DFM it is possible to create one model that can capture the complete functional behavior of a system. A process model is created and not a failure model as is the case with traditional safety analysis techniques like FTA or FMEA. This means that a model can be created that is independent of the analysis to be carried out. In Chapter 7 it will be demonstrated that it is exactly this capability that allows us to find failures in the design of a process that do not only represent traditional failure modes of hardware components, but also software design failures. Because a DFM model captures the complete process behavior it can be analyzed from any viewpoint of interest.
- A DFM model and its associated analysis can be used to support verification and validation activities. It can be used to verify the correctness of the design, to carry out failure analysis and to define test cases.
- With DFM it is possible to analyze the interaction between unusual parameters like hardware, software, human interaction or any other variables of interest. As long as the behavior between parameters can be captured it can be modeled with DFM.
- With DFM it is possible to analyze the behavior of software in the context of its the operating (hardware / process) environment. This allows the analyst to find prime implicants that do not contain traditional hardware failures, which means that there is an error in the software design.
- With DFM it is possible to model functional as well as dynamic behavior. It can take into account timing issues that affect the interaction between parameters of interest.
- The DFM methodology can be applied throughout any phase of the lifecycle of a system. DFM models can be created in a hierarchical manner, starting form a high level top design down to a low level detailed design.
- DFM is incorporated into an automated tool that integrates the capabilities of techniques FMEA, FTA, and HAZOP in one approach. DFM has the capability to backtrack into a system or to trace forward into a system. This allows us to find root causes for specified events of interest or to see the effect of conditions or events on an output parameter of interest.
- The software tool supports the creation and analyses of the DFM model [71]. The possible analyses require a deductive or inductive approach. This requires backward tracking or forward tracing through the system. The DFM software tool automates this process, which makes it possible to investigate thousands of combinations of events. This makes it possible to analyze systems that otherwise would be too complex to comprehend or to be carried out by a human analyst.

- The DFM model is easy to communicate to management. The graphical design interface can create models that reflect physically the layout of the manufacturing plant or process, and the safety system. Engineers and management can easily understand the results of the deductive analysis if the parameters chosen reflect the components and variables the way they are known by the designers of the manufacturing plant, safety system or application software.

4.8 Disadvantages of DFM

DFM has some disadvantages that all modeling techniques deal with. For example, the accuracy of the analysis depends on the level of detail used to create a model and the quality of the analysis depends on the expertise of the analyst. As with all methodologies or techniques DFM has some weaknesses also:

- The amount and size of the prime implicants can easily become unmanageable. The amount and size depends on two aspects. On one hand there is the level of detail that is put into a model. The more detail the larger, in number of literals, the prime implicants will be. On the other hand there is the option to make a dynamic model. The larger the time period used to analyze the model the more prime implicants and the larger the prime implicants in number of literals. Whether the resulting prime implicants are unmanageable depends on the available time and resources. A solution to make the analysis of the prime implicants more manageable is given in Chapter 5 by means of importance measures that can help prioritize the resulting prime implicants of a DFM analysis.
- The development of a DFM model requires time and resources. Creating a DFM model for a complex system is a team effort and requires experts in different industry fields. With DFM a lot of time and resources are spent creating and analyzing the DFM model. It is not clear whether an analysis like this will be economically beneficial. This needs to be further researched.
- The dynamic capabilities of DFM are limited. With DFM it is possible to model, to a certain extend, the dynamic behavior of a system. A DFM model is not a real-time model. With the transition box it is possible to take into account a time lag or time transition. This time delay represents the required time period between a change in the input variables and the update of the output variables. At this point in time it is possible to use only one time scale. Even though in theory it is possible to choose an extremely small time constant, and thus simulate almost a real-time environment, in practice this will result in tremendous modeling and analysis efforts. For example, if the smallest time constant in a system is 1 second, and the largest time constant in the same system is 1 day, then one would have to go back in time 86400 time steps in order to see any change in variables that are associated with transition box representing a 1 day time constant. This will result in too many prime implicants with too much undesired detail to analyze. This can be solved if the DFM models can be separated in models with time constants in the same order.

The software tool used to create the models should incorporate this feature and should be suited for combining analysis of different models. Right now this needs to be done by hand.

- A DFM model is deterministic. Only known unexpected behavior, for example an operator who changes a software valuable into a wrong value, is modeled. Real stochastic aspects are not taken into account.

4.9 Conclusions

This chapter described the DFM methodology, which will be used throughout this work to carry out process hazard analyses on complex systems. The DFM methodology is implemented in a software tool that uses a graphical user interface to model the parameters of a system and their interaction. The tool can be used to carry out deductive and inductive analysis of interest. The main advantage of DFM is that a process model is created that captures the complete desired and undesired behavior of the system. This process model captures all the technical aspect of the system, including hardware, software, physical parameters, environmental parameters, or human interaction when desired. This is an advantage as a process model is created and not a traditional failure model. The main disadvantage is that the output generated by DFM can be to enormous to be analyzed by hand. Therefore, the next chapter will introduce importance measures that can prioritize the output and help the analyst focus on what is important in terms of safety.

Chapter 5 Importance Measures

5.1 Introduction

This chapter introduces existing and newly developed importance measures that are used to support hazard identification and safety management in terms of analyses and prioritization. The chapter starts with an introduction to importance measures. The theory of these importance measures is described and their use is demonstrated by examples. The chapter will end with a discussion on the use of the importance measures.

5.2 Introduction to importance measures

An important problem in safety and risk management is the evaluation of the relative importance of components and parameters influencing the performance of a system. Importance evaluation can be used to support decisions related to modifying or improving the system. In the light of limited resources and other constraints, it is practically impossible to implement all possible design modifications that may improve the system operation. Priority should be given to components or parameters of higher importance. The assignment of such priority to prime implicants is the topic of this chapter.

In [72], a differentiation is made between ranking and categorization of items. Ranking is defined as arranging items in increasing or decreasing importance, while categorization deals with the allocation of these items into groups, according to some preset guidelines or measures. The focus in this thesis will be on ranking or prioritization rules as they can be defined and explained in general. Categorization will not be ignored, but will be explained by example as the rules to categorize depend on the specifics of the system, industry, and experience of the analyst.

Depending on the complexity of the system and the level of modeling detail, the number of prime implicants generated can reach unmanageable levels. Prioritizing the prime implicants according to some preset metrics and identifying important subsets becomes a practical necessity. The approach followed is based on prioritization according to a particular measure of interest or importance. Prioritization is based on the output of a DFM deductive analysis. A typical output file is presented in Figure 13.

```
For the top event:

At time 0  ,    TT = High-high (High-high)                  AND
At time -1 ,  PETN = 6%    (6 %)


There are 1292 prime implicants

Prime Implicant #1
     At time -2 ,   WPE = 1     (1 unit)                    AND
     At time -2 ,   DO1 = On    (Speed on)                  AND
     At time -2 ,   AO1 = High  (Speed high)                AND
     At time -2 ,   CNF = Normal (Operates normally)        AND
     At time -2 ,    BW = Wide  (Wide)                      AND
     At time -2 ,   DO7 = Off   (Speed off)                 AND
     At time -2 ,   CAM = Normal (Operates normally)        AND
     At time -2 ,  PETN = 1%    (1 %)                       AND
     At time -2 ,    TT = Low-low (Low-low)                 AND
     At time -1 ,   DO6 = Off   (No speed)                  AND
     At time -1 ,   DO7 = Off   (Speed off)                 AND
     At time -1 ,   CAM = Normal (Operates normally)

Prime Implicant #2
     At time -2 ,   WPE = 1     (1 unit)                    AND
     At time -2 ,   DO1 = On    (Speed on)                  AND
     At time -2 ,   AO1 = High  (Speed high)                AND
     At time -2 ,   CNF = Normal (Operates normally)        AND
     At time -2 ,    BW = Wide  (Wide)                      AND
     At time -2 ,   CAM = Off   (Stuck off)                 AND
     At time -2 ,  PETN = 1%    (1 %)                       AND
     At time -2 ,    TT = Low-low (Low-low)                 AND
     At time -1 ,   DO6 = Off   (No speed)                  AND
     At time -1 ,   DO7 = Off   (Speed off)                 AND
     At time -1 ,   CAM = Normal (Operates normally)
```

Figure 13. Example DFM output file

The output file needs to be examined as it contains the information from which the importance criteria need to be derived. Examining the output file helps understand that importance measures can be derived at:

- System level;
- Prime implicant level; or
- Literal level.

At the system level, information is available about the top event that occurs and the number of prime implicants that are derived for this top event. Information at the system level, the only information available about the top event would be the probability of occurrence of this top event. More detailed information is available on prime implicant level. Each prime implicant has a probability of occurrence and consists of a number of literals. These literals themselves contain the most valuable information about the system. A literal consists of a time step, a variable, and a variable state. The literal itself has also a probability of occurrence.

With this information in mind it is possible to develop probabilistic and non-probabilistic importance measures. The probabilistic importance measures are based on the probability of occurrence. The probabilistic measures are calculated indices,

mostly based on probabilities of the occurrence of the events associated with the prime implicants. The probabilistic importance measures discussed in this document are mainly derived from concepts and methods of risk ranking already developed by others. The following probabilistic importance measures are used:

- Probability of the Prime Implicant
- Risk Reduction Worth and Fussell-Vesely
- Risk Achievement Worth

The non-probabilistic measures reflect descriptive attributes of the prime implicants and literals. The following non-probabilistic importance measures are defined covering all the available information from the output file:

- Number of prime implicants;
- Contents of prime implicants;
- Number of literals;
- Number of variables or states;
- Time dependence of prime implicants;
- Correlation of variables or variable states.

5.3 Probabilistic importance measures

Quantification has always been of interest in the field of safety and reliability because it makes objective comparison of systems or designs possible. With DFM, probabilistic importance measures can be calculated if the probability of occurrence of the top event, the prime implicants and the literals are available.

When the probabilities of individual literals are available, caution needs to be exercised when calculating the probability of the prime implicant and of the top event. This calculation is relatively simple and easy to automate in a computer program if the probabilities of the literals are independent. In practice, this is seldom the case and calculations must be carried out by hand taking into account the possible dependency of the different variables. Appendix A, gives an overview of the computer tool developed and used to calculate the importance measures.

The following gives an overview of possible probabilistic importance measures that can be defined for literals and prime implicants. The probabilities used in these examples are determined using expert opinion and are not based on any existing figures. It is not the purpose of this thesis to prove the accuracy of the calculation but to demonstrate the usefulness of probabilistic importance measures. Further research needs to be carried out that demonstrates how probability calculations for prime implicants and literals need to be addressed, how it can be applied in practical situations, and how to automate the calculations using the DFM software tool. To calculate the probabilities the following assumptions have been made:

- The probability of a prime implicant is based on a Boolean AND relationship between the literals of that prime implicant;
- The probability of the top event is calculated by using the Boolean OR relationship between the prime implicants;

- As data are not always available the probabilities of individual literals are selected using engineering experience and judgment.

The following sections address the following probabilistic importance measures:

- Probability of the Prime Implicant;
- Risk Reduction Worth;
- Risk Achievement Worth;

5.3.1 Measure: Probability of the Prime implicant

The purpose of this measure is to be able to select those prime implicants that are of significant importance because of their high probability of occurrence. The importance of a single prime implicant to the top event gives valuable information about the prime implicant contributing the most to causing the top event. The prime implicant importance can be defined as the ratio between the probability of occurrence of a prime implicant and the probability of the top event. An alternative is the ratio between the prime implicant and the prime implicant with the highest probability of occurrence, which is useful as it normalizes the importance measure.

$$I_{PI_i} = \frac{P(PI_i)}{P(TE)} \text{ or } I_{PI_i} = \frac{P(PI_i)}{\max(P(PI))} \quad \text{(PI = Prime Implicant, TE = Top Event)}$$

This importance measure is particular useful when decisions need to be made on which prime implicants need to be examined first. Ranking can take place from high to low and rules can be defined that, for example, require for example analyzing the top 10% of the prime implicants that contribute the most to the top event.

5.3.2 Measure: Risk Reduction Worth

The purpose of this measure is to give an indication of the importance of a literal in relationship to reducing the probability of the top event. The basic idea of the Risk Reduction Worth (RRW) [72] and the similar Fussell-Vesely [73,74] importance measure is that a literal, without obviously being critical, can contribute significantly to the top event by its presence in one or more fault tree cut sets [28]. The risk reduction can be calculated by dividing the probability of the top event with the probability of the literal set to its true value, by the probability of the top event with the literal set to zero. The latter represents a situation as if it can never occur or in terms of failure analysis as if it is extremely reliable.

$$RW_{L_i} = \frac{P(TE)}{P(TE_{P(L_i)=0})} \quad \text{(TE = Top Event, L = literal)}$$

The Fussell-Vesely importance measure is calculated by dividing the fraction of minimum cut sets that contain the basic events by the probability of the top event. That is the probability of the top event minus the probability of the top event when the probability of occurrence of the basic event of interest is set to zero and this divided by the probability of occurrence of the top event. There is a direct relationship between Risk Reduction Worth and Fussell-Vesely:

$$FV_{L_i} = \frac{P(TE) - P(TE)_{P(L_i)=0}}{P(TE)} = 1 - \frac{P(TE)_{P(L_i)=0}}{P(TE)} = 1 - \frac{1}{RW_{L_i}}$$

The RRW and Fussell-Vesely measures are useful in identifying opportunities for improving the reliability of components that reduce the risk the most. The RW measure uses the same calculation concept as RRW or Fussell-Vesely. The RW measure gives an indication of how much the probability of the top event would be reduced if the probability of the specific literal would equal to zero (i.e., the prime implicant will not exist). It gives an indication of the importance of an individual literal. With DFM, the top event can be any condition of interest, desired, undesired, or a combination. These two importance measures should only be used with top events that represent undesired conditions as it does not makes sense to reduce the probability of an “desired state”.

5.3.3 Measure: Risk Achievement Worth

The purpose of the risk achievement worth (RAW) importance measure is to give an indication of the importance of a literal in relationship to increasing the probability of the top event [72]. The RAW measure represents the opposite of the RRW measure. Literally, the RAW measure represents how worth it is to keep the current level of reliability for a basic event. It is useful when it needs to be examined how important a component like a valve or pump is when this component temporarily is unavailable, for example, because it either failed or periodic maintenance is carried out.

The RAW importance measure gives an indication of how much the probability of the top event goes up if the literal always occurs (i.e., the probability of the Literal is one). The measure can be calculated dividing the probability of the top event with the probability of the literal set to one by the probability of the top event with the literal set to its true value.

$$AW_{L_i} = \frac{P(TE)_{P(L_i)=1}}{P(TE)}$$

With DFM, the top event can be any condition of interest, desired, undesired, or a combination. The RAW importance measure should only be used with top events that represent undesired conditions as it does not makes sense to reduce the probability of an “desired state”.

5.3.4 More probabilistic importance measures

The importance measures presented in the previous two sections have been presented as they are derived from well-known and probably are the most used importance measures. The PSA Application Guide specifically identifies Fussell-Vesely, Risk Reduction Worth, and Risk Achievement Worth as appropriate measures to use [75]. Several other probabilistic importance measures exist in the literature. As mentioned earlier several concerns have been raised about probabilistic importance measures. As research and development goes on, new importance measures are introduced or old ones are improved. For the completeness of this thesis, other probabilistic importance measures that can be found in literature are introduced and briefly explained.

In [75] the criticality importance measure is mentioned that considers the fact that it is more difficult to improve the more reliable components than to improve the less reliable components

Barlow and Proschan introduced a measure for systems whose components fail sequentially in time [76]. The Barlow-Proschan measure considers the sequence of event failures that causes the system to fail in time and is a function of the past behavior rather than a point in time [28].

Another well-known importance measure is the Birnbaum or reliability importance [72]. It is completely dependent on the structure of the system model and is independent of the current probability of the basic event. The Birnbaum measure is calculated as the difference between the probability of the top event with probability of the basic event set to 1 and the probability of the top event with the probability of the basic event set to 0.

A generalized risk importance measure has been introduced by Schmidt et al in [77] and Cheok et al in [72]. This importance measure allows any valid probability of the basic event and does not restrict it to the extreme zero and one. The generalized importance measure is defined for all probabilities of the basic event. It is possible to draw the generalized importance measures as function of the basic event probability. This relationship is called risk curve or risk impact curve and in [72] it has been demonstrated that these curves can give complete different perspectives on the risk importance of basic events compared to using extreme measures like RRW, RAW or Fussell-Vesely. This approach has been proven to give useful guidance on the importance of change in reliability and the impact on risk. It will identify basic events that when changed a little bit will have a significant impact on risk and basic events that when changed significantly will have only a relatively small impact on risk. This importance measure is very promising as it handles the uncertainty associated with data. If it is not sure whether the data can be trusted and the basic event is of little importance because of this measure then the use data does not play an important role anyway. On the other hand if the basic event turns out to be important but the data is uncertain then the focus can be on getting more quality data to improve the analysis.

In [78], Borgonovo and Apostolakis introduced a newly developed importance measure called Differential Importance Measure (DIM). DIM gives the analyst information about the importance of proposed changes that effect component properties and multiple events. An important feature of DIM is that it is additive, i.e., the DIM of groups of basic events or parameters is the sum of the individual DIMs. Like the generalized importance measure this measure has the advantage over RRW, FV and RAW that it does not depend on extreme calculations but rather focuses on small deviations from the original probability.

5.4 Non-probabilistic importance measures

If possible ranking should be based on probabilistic importance measures. Quantification of importance measure is preferred as it makes objective comparison of designs and systems possible. The probability of the literals and their existence in the prime implicants determine the importance of the literal in relation to the top event. On the other hand, it is also realized that probability data is not always available, or that it is hard to obtain, for individual literals. It is for this reason that a

set of non-probabilistic importance measures has been developed that can help an analyst improve the system. Unlike the probabilistic importance measures, the non-probabilistic importance should be used collectively. Analysis based on non-probabilistic evaluation should utilize a combination or all non-probabilistic importance measures. When used in combination with each other they can stimulate discussions among knowledgeable people regarding system improvements.

Non-probabilistic importance measures are based on the amount or interpretation of the contents of prime implicants and literals. The ability to interpret prime implicants requires qualification of each variable and its associated states. The following sections will give an overview of the non-probabilistic measures.

5.4.1 Measure: Number of prime implicants

On a system level, it is of interest to look at the number of prime implicants derived for a particular top event. Different designs or manufacturing concepts might give different total numbers of prime implicants. A design or concept that produces less prime implicants might be preferred over a design that produces more prime implicants. The more prime implicants there are the more difficult it is to analyze and manage all these prime implicants. Fewer prime implicants should not be interpreted as necessarily a lower probability of occurrence for the top event and thus a better system. In order to favor one design over another, the number of prime implicants should be significantly different, e.g., at least a factor of 5 or 10 decrease in prime implicants.

The advantage of having a design with fewer prime implicants is not that it necessary decreases the probability of occurrence of the top event but that the decision to analyze the number of prime implicants is much easier. If a design can be created that has instead of 2000 prime implicants only 500 prime implicants it might be decided to analyze every single prime implicant. If time and resources are allocated to address the prime implicants it is possible to identify further design changes, or to implement protection measures in the form of safety layers. If there are too many prime implicants that need to be analyzed then decisions need to be made on how to do this. Other importance measures will be introduced that can support further analysis by focusing on those prime implicants that are of interest.

The disadvantage of this measure is that in practice, a “better” design does not necessarily produce less prime implicants. DFM derives all conditions that can cause the top event of interest. If a design is changed and based on redundant instead of single safety systems, it is likely that the number of prime implicants is more for a particular top event. The design is safer and the difference can only be found by examining the contents of the prime implicants. The following importance measures can support further analysis. The importance measures can be used in combination with importance measure to make further system improvements.

5.4.2 Measure: Contents of prime implicants

The purpose of this measure is to use the contents of a prime implicant as a reason to prioritize. It is possible to prioritize the prime implicants if their contents can be interpreted. To achieve this, the interpretation is based on the previous qualification of variables and variable states. The qualification is derived from what the variables and variable states represent in the context of the process or system. A

differentiation is made between the qualification of variables and the qualification of variable states. The qualifications of variables typically describes the nature of the variable, e.g., the following qualifications are useful for the PETN manufacturing process:

- Process:
  A variable describing process conditions like pressure, temperature or speed.
- Software:
  A variable that represents software, e.g., an integer or real variable, or a software input or output signal.
- Human action:
  A variable that represents the interaction of a person with the system. For example, a valve that can only be manually operated or a set point for a value to be programmed by an operator in the memory of a control system.
- Conditional variable:
  A variable that can influence a physical process because of its condition. For example, a valve that is stuck closed or a temperature sensor stuck at 30 °C.

The above examples are based on what the variables represent in a system. The qualification of variable states is based on the condition of this variable. For example, a state can be classified as “normal”, “failed”, or “undesired”. There are no fixed or prescribed qualifications. The analyst can adjust the qualification to specific needs, or depending on the system or industry.

For conditional variables, as defined by DFM, qualification is easily done. If a valve is stuck closed, the state is classified as “failed”. If a valve operates normally, the classification would be “normal”. For process and other variables this cannot always be done, as it is not always clear whether a process condition is desired or undesired. A temperature of 100 °C in a tank might be undesired during startup of a plant but desired during normal operation. A drain valve can be open or close, both normal situations. However, if the drain valve stays closed when it needs to open or opens when it needs to close then these states are undesired. The classification actually depends on the context the variable is in at that time. These kinds of situations (i.e., combinations of seemingly normal operational states that result in unwanted behavior) are so hard to find. Only when it is obvious that a certain condition is unwanted it is possible to classify it as “undesired”. For example, in the presented case study the temperature in the tank is not allowed to exceed 35 °C, at any time. Even though it as a possible process condition, a higher temperature would be undesired because of the possibility of an explosion.

DFM has demonstrated itself as a tool that can examine or find these kinds of situations that are desired or undesired depending on the context of the system. Garrett and Apostolakis were the first to demonstrate this, by using DFM to examine the influence of the operating environment, i.e., the context, on a system. They used DFM to examine the (hardware) operational environment on mission critical software and how it can make the supposedly correct software fail [79]. In this document, it will be demonstrated how the importance measures can support this type of analysis as it is possible to set up rules in the database that can look for suspicious situations.

Once the variables and states are classified, it is possible to collect valuable statistical information about the contents of prime implicants. For each prime implicant the literals are interpreted and the number of occurrences of the possible qualifications counted. Once the qualifications are counted it is possible to prioritize and look, for example, for prime implicants with zero failed states, with only software states, or with one failed stated or one or more software states. This importance measure supports analyzing the prime implicants for one of the most beneficial features of DFM, as deductive analysis can result in prime implicants with no hardware / human failures. This means that the software caused the failure, i.e.. a design error that apparently was not found during software testing. Any prioritization, representing an analyst's interest, can be applied in this way.

5.4.3 Measure: Number of literals in a prime implicant

Emphasis can be placed on eliminating prime implicants with a small number of literals. Prime implicants can then be examined on measures like "no single prime implicant", i.e., containing only one literal. If the probability of a variable being in a state is small and independent, the occurrences of single events should be more often then those of doublets or triplets. The number of acceptable literals or analyzing a certain percentage of the total prime implicants can be set as a rule. For example, examine all prime implicants with less then 10 literals, or examine the top 20% of all literals with the least literals. A typical application could be to rank prime implicants with two or three literals. These prime implicants can then be examined for common cause failures. Using this measure, rules can be established that eliminate prime implicants and focus the analysis on a set of prime implicants that are of interest because of the selection measures.

5.4.4 Measure: Number of variables or states

A very simple but effective importance measures is the ranking based on the number of occurrences of a variable or variable state. The number of times a variable or variable state is present gives an indication of the contribution of this variable or variable state to the top event. The advantage of this importance measure is that it gives the analyst an indication of where to focus further analysis, i.e., which variables or states to examine more closely.

There is also a significant difference between ranking for variables and variable states. If a variable shows up on top of the list then it should be questioned which variable state is causing this. If the stuck open state of a shutdown valve shows up on top of the list then it can be decided to use a shutdown valve that uses a fail-safe design.

5.4.5 Measure: Relation between variables or variable states

The purpose of this measure is to identify whether there is a relation between the occurrence of certain variables or variable states within the set of prime implicant that belong to a top event. This measure is useful as it might not always be possible to eliminate the main variable or variable state, i.e., the variable or variable state that occurs the most. It might be possible though to eliminate the one that occurs often in combination with this one. This would also make it possible to eliminate prime

implicants. This importance measure is particularly useful if the variable or variable state that occurs the most often is one that is actually desired.

For example, the normal condition of a pump might be desired but under certain circumstances this pump needs to be pumping and should not be in the off position. If the pump is working correctly other problems might exist that cause this pump to be off. It could be that the control software in the basic process control system gives an off signal. By examining the relation between the normal state of the pump and all other variable states it is possible to find other conditions like software variables that can cause the pump signal to be off. This measure requires careful interpretation of the analyst. The analyst can apply this measure if the results of other measures don't make sense or just need further examination.

It is possible to make a matrix that presents the relation between each existing variable or variable state. Table 29 gives an example of a possible relation matrix.

Table 29. Relation between variables

	Variable A	Variable B	Variable C
Variable A	-	X	Y
Variable B	X	-	Z
Variable C	Y	Z	-

5.4.6 Measure: Time dependence of prime implicants

The purpose of this measure is to identify prime implicants with a time dependent relationship of interest. Each literal has a time step associated with it, i.e., this literal needs to occur at this particular time compared to the time of occurrence of the top event. It is possible to rank prime implicants based on literals occurring, for example, within one time step, or in between a specified number time steps. This information can be used in combination with or to specify repair and maintenance procedures. It would be useful to eliminate the investigation of prime implicants were the different literals occur with too much time in between. If periodic maintenance is carried out on specific time intervals then it can be assumed that certain undesired states, like valve stuck closed, would be detected during this scheduled maintenance. Prime implicants with literals occurring after this period can then be eliminated from the analysis, narrowing down the number prime implicants for the analyst to consider.

This approach can also be used to specify periodic tests. Prime implicants can be prioritized depending on the time step that a literal needs to occur. If all prime implicants are identified that occur above a certain time step then it is possible to examine these prime implicants and specify functional tests that need to be carried out during the maintenance activity.

5.5 Use of Importance Measures

The benefit of using importance measures is the ability to use them as input or as basis of decision-making processes. This is the reason why they have been developed and can be used with the DFM program. The use of importance measures depends on the specific system subject to the analysis, the experience of the analyst

and company, national or international guidelines like design procedures or applicable standards and laws. Different systems or analyses require a different use or application of importance measures.

The importance measures need to be applied with care to prevent any misinterpretation. The interpretation truly depends on the top event that is defined for the deductive analysis. For example, a manufacturing process might have an ideal operating temperature of 50 to 60 °C in order to produce qualitative good products. The same process might have a maximum temperature of 70 °C in order to maintain safety. Both situation can be defined as top events in DFM and will produce applicable prime implicants. The interpretation of the different importance measures will be different as one focuses on safety while the other focuses on quality aspects. The analyst has to make sure to relate the results to the actual meaning of top event.

The probabilistic as well as the non-probabilistic importance measures have their pros and cons when used. Both of them need to be applied and used with caution. One of the problems identified in literature with prioritization is that the ranking is always based on individual contributions and not combination of contributions [80]. The probabilistic DIM importance measure and the non-probabilistic measure introduced to investigate the relation between variables or variable states seem to be a welcome solution.

The probabilistic measures are the preferred importance measures as they can individually measure the importance of a variable or parameters in terms of the top event in terms of probabilities. The probabilistic importance measures have been well established over the years, but questions arise about their useful applicability and correct use in terms of reliability data. Their usefulness is questioned because the probabilistic measures work with extreme values, i.e., putting the probability of an event either to 0 or to 1. In practice this is an unrealistic approach. To successfully apply probabilistic importance measures it is necessary to have probabilities of literals, prime implicants, and top events available and understand the relationships between the possible combinations of literals. Prime Implicants can contain many different literals that are not, per definition, independent events. Arriving at the right reliability data for these events is a challenge and therefore the practical use of probabilistic importance measures can be questioned. A positive development in the field of probabilistic measures is the introduction of measures that can deal with uncertainty in probability data. The generalized risk importance and the DIM measure seem to be promising solutions as they can analyze the effects of small changes parameter data. It would be useful to research how they can be used in conjunction with the importance measures approach developed for DFM.

The non-probabilistic importance measures give information about the contribution of an individual variable state towards the top event. The non-probabilistic importance measures cannot be used in isolation. For example, just because a variable state occurs many times it is not clear yet whether this variable state is truly important. It is the contribution in terms of probability that determines whether a variable state is truly important. It is not possible to apply a single non-probabilistic importance measure that will solve all problems for a particular system. It is always preferred to use a combination of importance measures. Such analysis will give good insight to a system and can initiate intelligent discussion on system improvements between analysts.

5.6 Conclusions

This chapter described importance measures that can be used in conjunction with the prime implicants that are derived from the DFM analysis. Probabilistic and non-probabilistic importance measures are introduced. The probabilistic importance criteria are the preferred measures as they can measure the importance of literals and prime implicants in an unbiased manner. They can be used to prioritize the prime implicants using cut-off measures.

The non-probabilistic importance measures have been developed, as probability data will not always be available or difficult to obtain. Analysis based on non-probabilistic evaluation should always utilize a combination of non-probabilistic importance measures. When used in combination with each other they can stimulate discussions among knowledgeable people regarding system improvements. The importance measures give the analyst guidance on which prime implicants are most important to analyze and can therefore be used as a prioritization method.

Chapter 6 Safety Lifecycle Application

6.1 Introduction

The purpose of this chapter is to introduce DFM and the importance measures as a lifecycle safety management method. This means that DFM and the importance measures will not only be used to design safe industrial processes but also to operate the industrial process in a safe manner.

Figure 14 gives an overview of the steps to be taken when designing a process plant for safety. In short, these design steps focus on identifying process hazards, implement appropriate safety measures against these hazards, and assure that the implementation of the safety measures is reliable. There is a direct relationship between the process, the safety functions that protects the process, the safety system that carries out the safety functions, and the diagnostic systems that monitor the correct operation of the safety functions. DFM will be used to integrate the design of the process with the safety system that carries out the safety functions to ultimately determine the diagnostic functions. This makes it possible to design the safety functions and the diagnostic systems in the context of the manufacturing process that is protected. To accomplish this task it is required to

- Support the design of the (manufacturing) process;
- Identify safety interlock and instrumented functions;
- Support the design of the hardware and application software of the safety system; and
- Identify diagnostic systems.

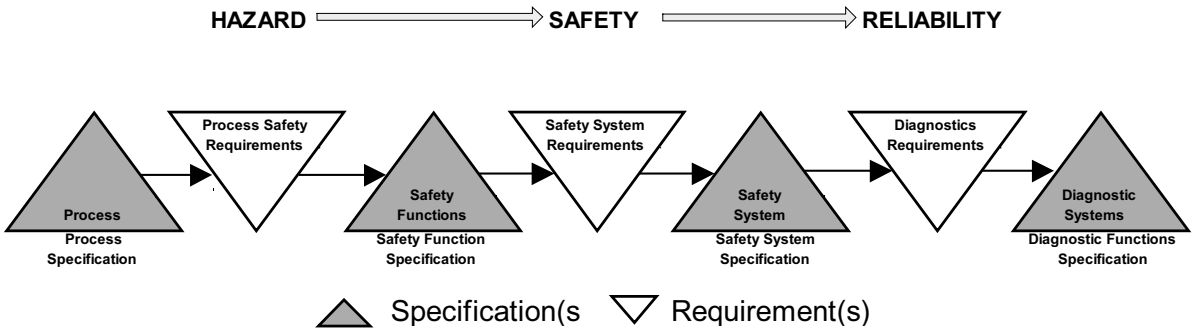

Figure 14. Process Safety Design

As it is not enough to only design a safe process, also the concept of a real-time alarm management system is introduced that supports the safe operation of a manufacturing process. This alarm system is intended to efficiently manage abnormal or undesired conditions that can occur during plant operation. The operator can use the alarm system to make informed decisions about the safe operation of the plant. The following sections will elaborate on the use of DFM and the importance measures to design and operate a manufacturing plant in a safe manner.

6.2 Process design

The design of a manufacturing process depends on some fundamental decisions that relate among others to aspects like the manufacturing concept used, or the objectives to be met in terms of production capacity. Once these decisions have been made the actual design work of the manufacturing process can start. This design should be made as inherent safe as possible. The initial design of the process includes the basic process equipment and control system, as this is the minimum equipment necessary to carry out the process. DFM and the importance measures are used to analyze and support the design in terms of safety.

The design process is captured in Figure 15. A DFM model is created for the initial design of the process. One or more top events will be specified for this DFM model to carry out deductive analysis. These top events represent the objective of the analysis, i.e., process conditions of interest in terms of safety. The deductive analysis results in a number of prime implicants that are analyzed using the importance measures specified in Chapter 5. The selected importance measures represent the design rules of the company or the independent party. Once the design rules are applied the question needs to be asked whether the current design is acceptable or not. If there are any conditions in the plant that do not meet the design rules then the answer is no. The design needs to be changed taking into account the prime implicants that result from application of the design rules. Once the design changes have been made a new DFM model is created taking into account these changes. The analysis process is repeated until the design of the manufacturing process meets the measures or rules used by the analyst.

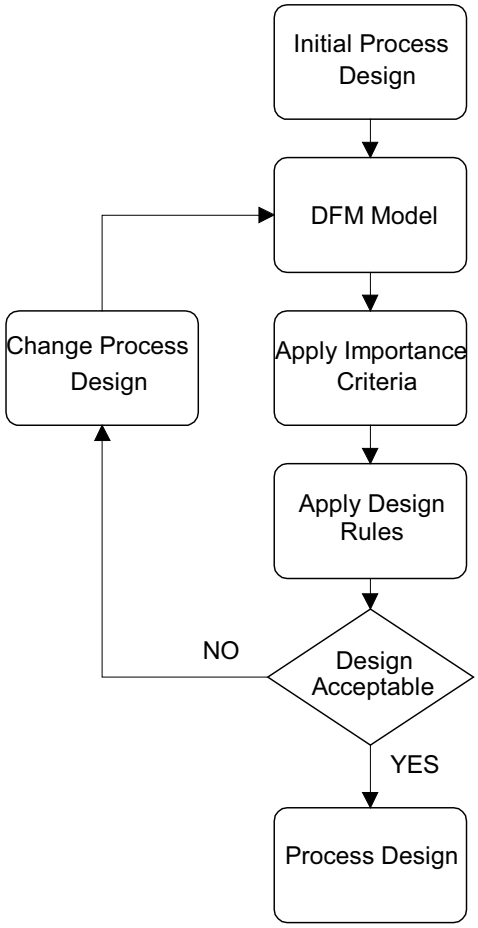

Figure 15. Iterative Process Design

6.3 Identification of safety functions

For most industrial manufacturing processes it will not be possible to remove all hazards and therefore it will be necessary to identify appropriate safety functions to further protect the manufacturing process. This section will explain what a safety function is, the kinds of safety functions that exist and what information can be derived from DFM and the importance measures to implement safety functions.

A safety function can be defined as a function to be implemented by a safety-related system that is intended to achieve or maintain a safe state for the equipment under control, in respect of a specific hazardous event [18]. The purpose of a safety function is to continuously diagnose the status or condition of the process. The safety functions monitors the process for any conditions that can exist in the plant that can upset the process and that requires immediate action. The process carried out by the safety function consists of gathering information, interpreting this information and make a decision on the kind of action to take (see Figure 16).

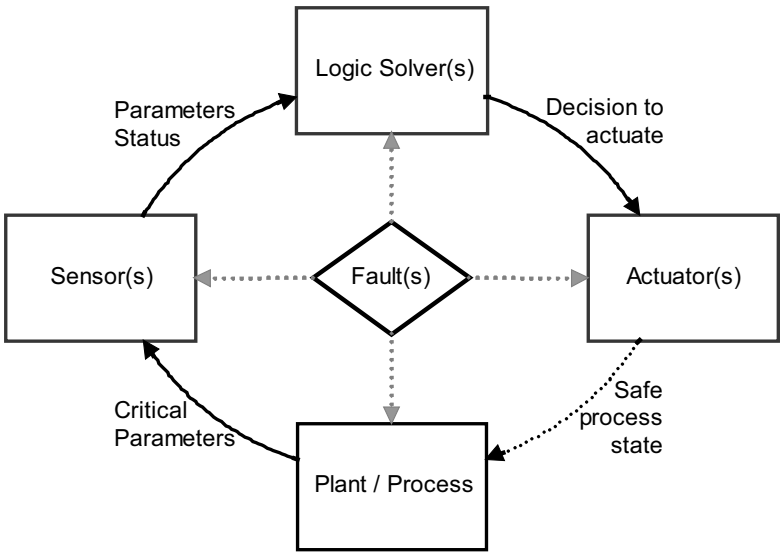

Figure 16. Safety system diagnosing the status of the process

A safety function is defined by the following five characteristics:

1. The sensing element;
2. The logic solving element;
3. The actuation element;
4. The timing element; and
5. The reliability element.

The sensing element measures a physical quantity of interest, e.g., pressures, temperatures, speeds, or particles. The logic element solves some kind of logic concerning the measured physical quantity. Usually this is rules based, e.g., `if x > y then z`. The actuation element representing the action to be taken after the logic is executed. In terms of safety this usually means one of two things, either do nothing, or take action by, for example, closing a valve, starting an electrical motor, or sounding an alarm.

The timing element is required to specify the time period in which a safety function needs to be fully executed. This time depends on the process safety time, i.e., the time during which it is possible to detect undesired conditions and correct or handle them in a way that the process will reach a safe state. Figure 17 gives an overview of the different timing aspects that need to be taken into account when designing a plant and identifying the required response times for the safety system.

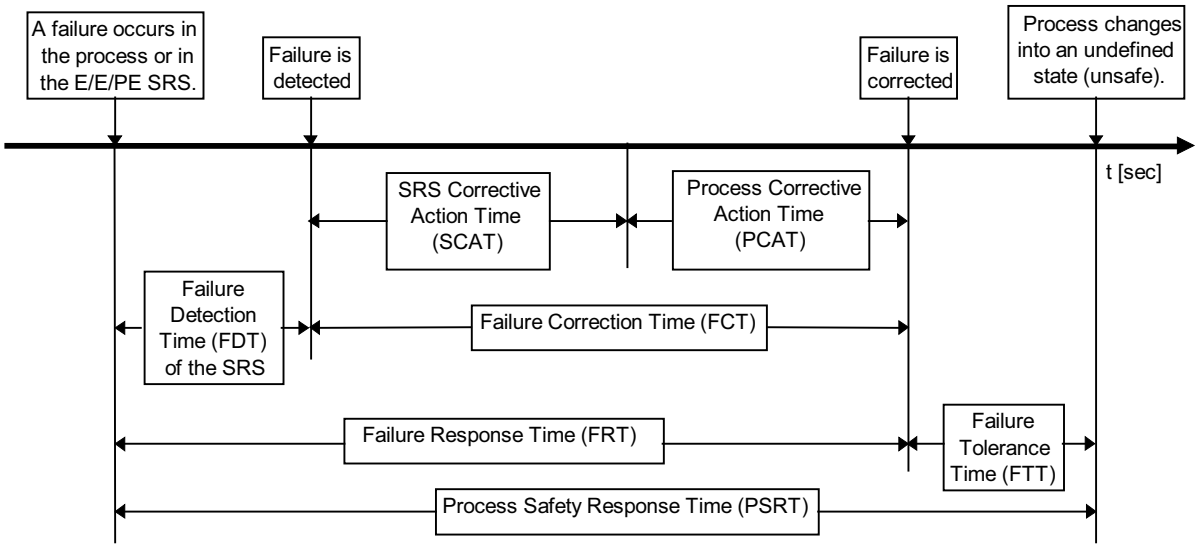

Figure 17. Process Safety Response Time[4]

The reliability element determines the acceptable probability of dangerous failure on demand. The probability of failure on demand is the probability that a safety system is not capable of responding to a process demand. It is not capable, because of an internal failure of any kind.

Safety functions are derived from the final process design. DFM and the importance measures are used to support the identification of the safety functions. The prime implicants resulting from the final process design contain the necessary information to help define the input, logic solving, and actuation element. DFM cannot be used to determine the process safety time and thus the required safety response time of the safety function. But, during the design of the safety system, DFM can be used to determine the response time of the safety function. This time can then be used to verify by other means whether it is suitable or not. Determining the required reliability of the safety functions requires quantification of the risk involved with the process and a definition of the acceptable risk. This is beyond the scope of this work.

6.4 Design safety system

At this point the process has been designed and safety functions have been identified. The next task is to design the safety system that carries out the safety function. An example a programmable electronic safety-related system is given in Figure 18. Industry pays a lot of attention to these safety systems as they serve as a last layer of defense in the protection layer philosophy (see Figure 4). If a safety system fails to respond to a process demand because of an internal random, systematic, or common cause failure then an accident can usually not be prevented. Because these safety systems play such a critical role, and because they are based on programmable electronic technology (and thus software) they are subject to independent third party functional safety certification. A safety system is defined functionally safe if random, systematic and common cause failures do not lead to malfunctioning of the system and do not result in injury or death of people, spills to

[4] This figure illustrates the principle. Time intervals strongly depend on the process and do not reflect actual duration(s).

the environment, and loss of equipment or production [81]. Designers of safety systems as well as the independent parties need to verify the design for functional safe behavior. This requires detailed information about the hardware as well as the software behavior of safety systems.

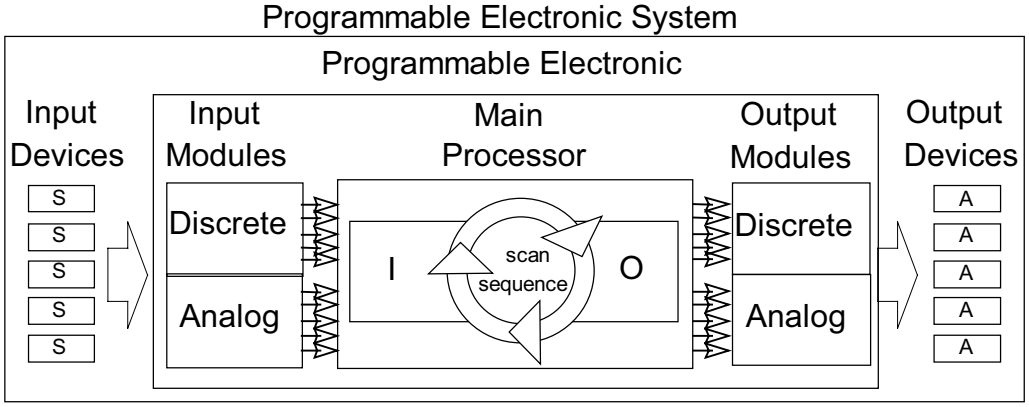

Figure 18. Programmable Electronic Safety-Related System [82]

In the certification industry it is required to use of the V-model to assure that functional safety is incorporated from the start of the design [83]. The V-model is a process model that guides the developers along the design process and helps them identify the necessary test specifications and product documentation (see Figure 19). In this way it is not only possible to arrive at a safe design but also to achieve the required traceability of the requirements. It supports the required verification activities to be carried out by an independent party.

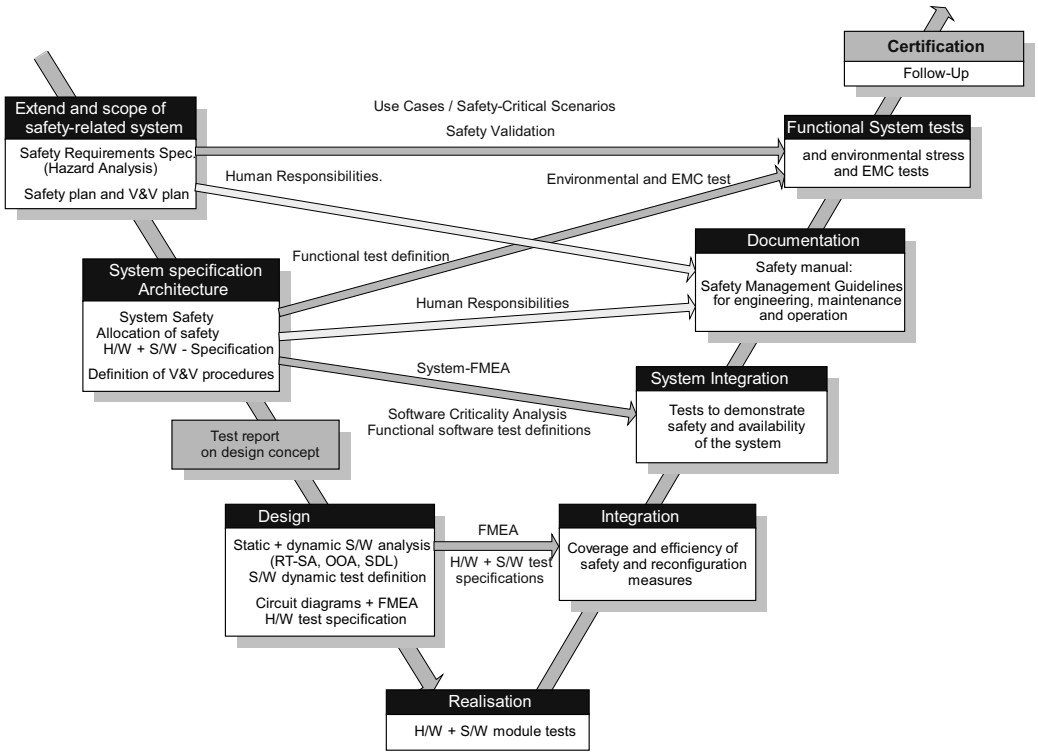

Figure 19. V-model for safety system design [83]

The left side of the V-model represents the activities necessary to design the safety system. The right side of the V-model represents the necessary activities to test and verify the accomplished design. The design starts from the safety requirements specification as derived from the safety function requirements. The design starts on a high system level. After the high system level has been approved, the actual hardware and software is designed in more detail and actually realized. From the safety requirements specification, the high-level system design, and the detailed hardware and software design tests are generated that will be used to verify the intended behavior of the safety system. As these tests are generated on every level of the system design, and are recorded throughout testing phases on the right side of the V-model the required traceability is established that can be used for verification activities by an independent party.

DFM and the importance measures can be used in the different phases of the V-model to design the safety system, test it qualitatively, and define test cases that can be used once the design is realized in hardware and software. In this capacity it can serve as an important design and verification tool. When it comes to software, DFM can be used to model the implemented software code and verify the correct behavior or to generate test vectors for software that already exists. An advantage is that the DFM model of the safety system can be integrated into the DFM model of the manufacturing plant. After safety analysis using the DFM model of the safety system, additional safety analysis can be carried out using the integrated model. This ensures that the safety analysis address the performance of the design of the safety system in the context of the process, see Figure 14.

6.4.1 Safety system design and analysis

The design of the safety system is based on the safety functions identified in the previous section and on design requirements that are derived from design guidelines, like (inter-) national safety standards. The design guidelines determine, for example, the redundancy and voting aspects of the safety function, the use of diverse means to carry out the safety function, or the kind of failure modes that the safety system must be able to withstand. The actual design of the safety system needs to take into account all specified requirements and operate as desired within the context of the manufacturing process.

Safety system designs are usually modular, i.e., different modules are designed that support different parts of the safety system. For example, typical modules are the input and output modules, the processor modules, or application software. The design and analysis of the safety system can be top down, starting with a high level design on building block level. This high level design captures the safety philosophy in terms of voting and redundancy, but not the detailed functional and failure behavior of the safety system. To truly understand the behavior of this safety system it is possible to design and analyze each building block in detail.

Once the design of the building blocks are as intended they can be integrated and the behavior of the safety system can be analyzed as one system. In first instance the design of the safety system is analyzed independent of the process. Once this design of the safety system is approved it needs to be incorporated into the process design and the process and safety system need to be analyzed as one system. This integration makes it possible to analyze the safety system within the context of the manufacturing process.

6.4.2 Application software design and analysis

Software does not have the same failure behavior like hardware. It does not fail like hardware does, e.g., because of aging. Software always executes the way it was implemented, and therefore can only consist of systematic failures[5] [79]. This in contrast to hardware where also tolerance and degradation failures can exits. Systematic failures can be made because of specification errors or human errors. For example, programming an “AND” gate while an “OR” gate was required, or programming the “+” while a “-“ was intended. Even if the design is the way it was intended, this is not always the way the customers uses the software. The main question for software is twofold. First of all is it designed to do what it is supposed to do and second, does it not do what it is not supposed to do [84]. Software code needs to be implemented to operate safely under all (desired and undesired) conditions.

It is possible to model the individual software and test the behavior of individual routines by specifying top events that simulate the intended output of that software routine. In this manner it is possible to test every individual software routine and integrate these routines where necessary to build the complete software logic. Each individual building block can be tested either by deductive or inductive analysis. The results of deductive analysis can be analyzed using the importance measures and look for situation that don’t make sense. Inductive analysis becomes more difficult as the application logic grows, as it is not clear what kinds of test cases are required.

Eventually the complete implementation of the application software can be incorporated in the DFM model of the hardware design of the safety system (see Figure 20) and the design of the manufacturing process. The software design and safety system design can be integrated into the process. This approach actually models the possible operational environment of the application software in the safety system. Independent of whether the software was actually designed for this environment it can now be tested to this environment. This makes it possible to analyze the software using the concept of an error-forcing context. If the software leads to an unexpected state without an apparent failure in the design then the software was actually not designed to handle that certain situation or environment (the “context”).

[5] Design failures are often classified as systemic or systematic failures

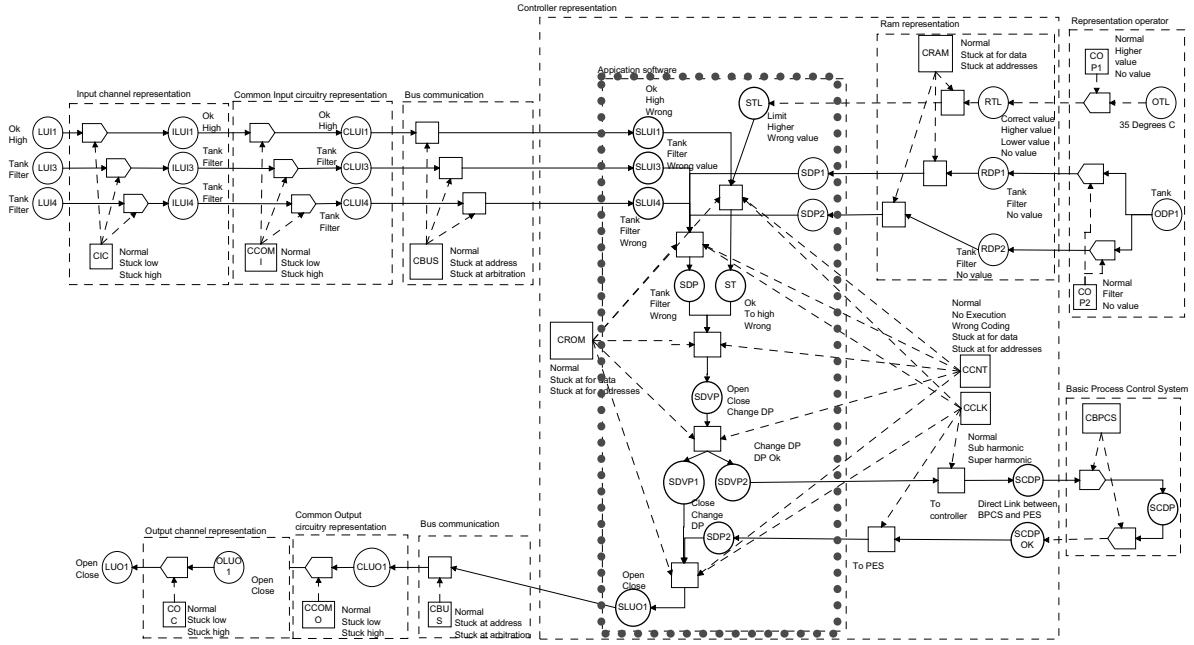

Figure 20. DFM model with application software

A top event is defined for the DFM model that integrates the process with the safety system, including the application software. The resulting prime implicants need to be analyzed for situation that clearly indicate that something must be wrong with the intended software design. Typical prime implicants that would indicate a failure in the software are those that don't contain hardware failures, or those that only consist of normal process conditions, or software variables. The importance measure to be used in this case is the one that can examine the content of the prime implicants. An analyst needs to be able to interpret the results when this prime implicant is applied. This prime implicant can add a lot of intelligence to the analysis but only if also only applied with intelligence. Chapter 7 will demonstrate examples of prime implicants that clearly show that there is a problem with the software design.

6.4.3 Analysis of existing software

Sometimes, the software implemented in the safety system can be too complex to model by hand or information about the software is just not available, e.g., in the case of existing software (commercial of the shelf software). In these situations, it is possible to model the actual software as a black box and use DFM to generate test vectors for the input parameters to the software.

These test vectors are based on all possible input variables and condition nodes of the DFM model. These states in the DFM model are not developed further because they represent the lowest possible detail, e.g., a failure mode or a process variable. With today's computational power it is possible to define all possible test vectors and run the actual software with all these test vectors. The test vector consists of all variables and their possible states. For the safety system as presented in section 4.6 this would mean that the following variables would be included, see Table 30.

Table 30. Variables and # of states comprising test vectors for example system

Variable	# Of states	Variable	# Of states
LUI1	2	CCNT	5
LUI2	2	CCLK	3
LUI3	2	CRAM	3
CIC	3	COP1	3
CCOMI	3	OTL	1
CBUS	3	ODP1	1
CROM	3	COP2	3

The multi-state representation of the input parameters to the software is the basis for the test vectors. Each test vector in the example consists of 17 different variables of which most can be in more then one state. In total there will be 262.440 different test vectors. Although this sounds a lot with today's computational power this should not be a problem. The actual software is executed using these test vectors and the output of the software is fed back into the DFM model (see Figure 21). In this way, it is possible to find conditions that would produce undesired output results. These conditions do not necessarily need to represent hardware failure conditions. Special test vectors can be created that only represent normal process variables. All variables that represent conditional variables can be set to "operating without failure" and then it is possible to test the software truly on design failures, as it should be able to handle possible process condition.

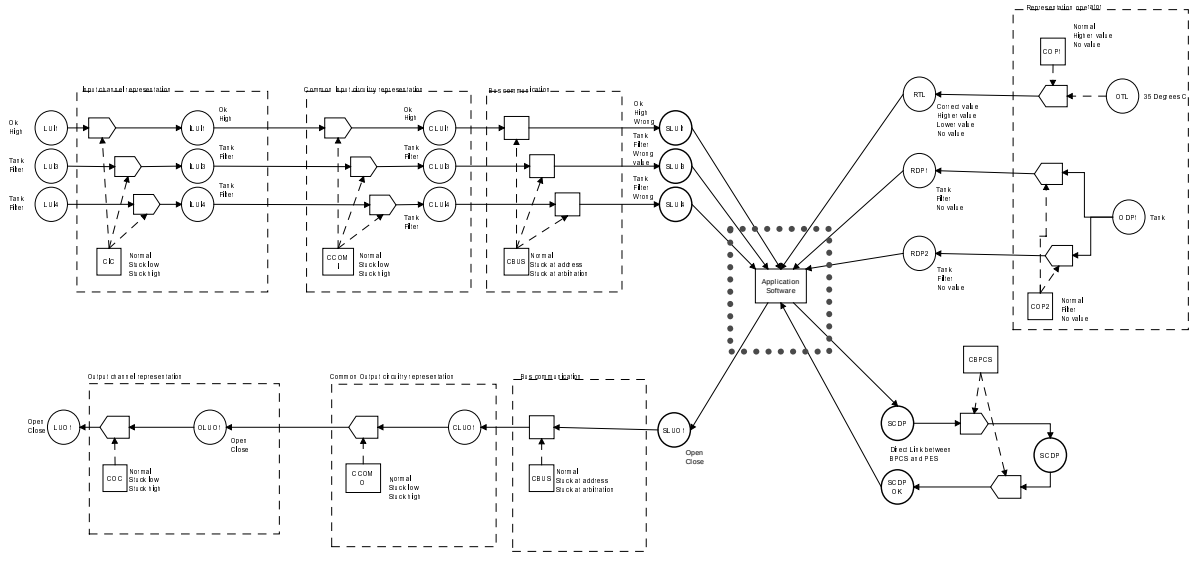

Figure 21. Testing existing software with DFM

Utilizing test vectors like this and verifying desired and undesired outputs turns DFM into an automated fault injection tool. Fault injection is particularly useful in finding situations that were not "thought about" during the design specification phase,

also referred to as the error-forcing context. Chapter 7 will demonstrate test vectors that allow us to find errors in existing software.

6.5 Identification diagnostic functions

At this point in time the safety system that carries out the safety functions has been designed and analyzed. It has been integrated into the process and the correct functionality has been verified in the context of the process environment. The next step in the safety design process is to assure that the implemented safety functions are actually carried out by the safety system. The worst-case scenario is to have safety functions with internal failures that are not able to respond to a process demand. The safety functions need to be monitored for correct operation.

This can be accomplished by adding an additional layer to the safety system that continuously diagnosis the status of the safety functions (see Figure 22). A diagnostic layer is necessary, as failures cannot only occur in the process (which are protected by the safety functions), but also in the hardware and software that carries out the safety functions. Diagnostic systems need to be implemented that reveal safety function failures, when they occur. This makes it possible to immediately repair the safety function before an upset condition of the safeguarded process occurs, or if necessary to put the process into a safe state if protection cannot be guaranteed. Diagnostics can improves the reliability of the safety functions and thus the safety of the process. The diagnostic layer can therefore be seen as an additional protection layer in the safety protection layer philosophy.

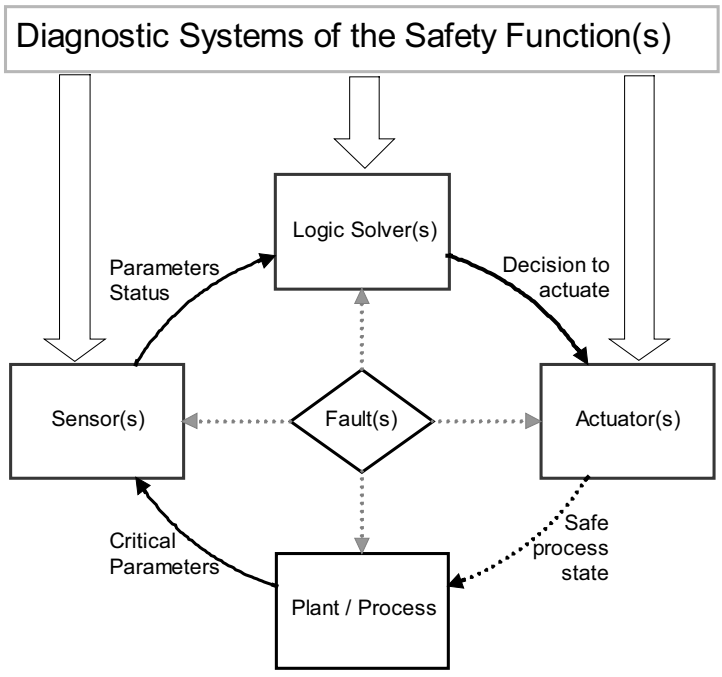

Figure 22. Diagnostic systems diagnosing the status of the safety function

Most of the safety systems are general-purpose safety system that can be programmed for the process it is supposed to safeguard. The diagnostic systems of these general-purpose safety systems are mainly concentrated in the programmable electronic main processor (see Figure 18). The general-purpose safety systems have a disadvantage as currently the built-in diagnostics are designed independent of the

process that the safety system needs to protect. The manufacturer of the general-purpose safety system only builds in diagnostics for known safety system failures, but this might not be sufficient to detect failures in the safety function that protects the process.

Indirectly there is a relationship between the process, via the safety functions of the safety system, and the required diagnostic systems of the safety system. This is demonstrated in Figure 14 and Figure 22. The behavior of the process determines the safety functions that need to be carried out by the safety system. If the safety functions are known it is possible to build in the diagnostics functions that the safety system requires to diagnose the status of these safety functions. The diagnostic systems have to monitor the performance of the safety system within the context of its operating environment, i.e., the process. Therefore, the safety system cannot rely on the general purpose diagnostics that exist in the different element of the safety system. The diagnostics need to be tailored to the specifics of the manufacturing process (the operating environment).

With DFM it is possible to identify the required diagnostic systems of the safety functions with the context of the operating environment. When and integrated DFM model is created, i.e., a DFM model of the process including the safety system and the application logic, it is possible to derive the required diagnostic systems using the deductive analysis approach. A top event can be specified that focuses on the incorrect behavior of the safety function.

6.6 Verification activities

Verification is confirmation by examination and provision of objective evidence that requirements have been fulfilled [18]. It plays especially an important role when it comes to programmable electronic safety systems, as industry requires verification by an independent third party. DFM and the importance measures are used as to define verification activities in the format of checklists or fault injection tests. Checklists can be used to verify that all requirements have been met and tests can be defined to demonstrate that a system performs as intended (as specified by the specifications). Independent parties can accept or reject the design of the plant or safety system using the results of the verification activities.

In order to verify the design of the process and the safety system the independent party needs to make a completely independent DFM model and carry out independent analysis. Depending on the outcome of the verification the independent party can accept the final design, accept it with revisions, or reject the design. Should the design be rejected then the design team can use the prime implicants found by the independent party as guidance to make design changes.

An independent party can create its own DFM model of the actual design of the plant and define the top events of interest. The resulting prime implicants can be used as a checklist to verify whether the plant owner implemented to correct safety function. The plant owner submits a list of safety functions that are implemented in safety systems, and the independent party verifies this list using the checklist. Only if the safety functions identified by the independent party are implemented one hundred percent the process design is acceptable.

When it comes to the safety system an independent party can make its own DFM model and analyze it applying design rules utilizing the importance measures.

The resulting prime implicants can be used to test the system against failure behavior, as it serves as fault injection testing. In this manner a multitude of analysis can be carried out that can verify the correct behavior of the safety system. Eventually the safety system needs to be integrated into the DFM model of the process to demonstrate that the safety system operates as intended under all possible process conditions.

The DFM model of the integrated process with the safety system can be used to verify the required diagnostic coverage. The resulting prime implicants can serve again as a checklist and as fault injection test cases. Every prime implicant can be used to verify whether the actual implemented system can detect these failures. If there is doubt about the implemented diagnostic then the prime implicant can serve as a test vector. In this way the independent party can verify whether a diagnostic is truly implemented or whether the implemented diagnostic is of the desired quality. The independent party can use the checklist to determine the actual coverage of dangerous failures. If equal weight is given to all required diagnostic systems then the total diagnostic coverage can be calculated as follows:

$$DC = \frac{\#\text{Of Diagnosed Prime Implicants}}{\text{Total \# of Prime Implicants}}$$

If all the prime implicants were covered by a diagnostic system then the diagnostic coverage of the safety function would be 100%.

6.7 Real-time Alarm Management System

The previous sections explained how DFM and the importance measures are used to support a new design or verify an existing design. The purpose of this section is to demonstrate how the final design of a plant and the final results of a DFM analysis can be used to build a real-time alarm management system. It supports risk and safety management during the operational phase of the lifecycle and possible maintenance, repair or retrofit. The objective of this section is to introduce the fundamental elements of the real-time alarm management system (RAMS) intended for monitoring deviations in the operation of industrial processes at manufacturing plants.

6.7.1 Introduction to the Real-time Alarm Management System

It is estimated that alone in the US petrochemical industry abnormal situations result in a loss of more than $20 billion dollars per year [85]. These costs are associated with events like large disasters, operational interruptions, unscheduled shutdowns, equipment failure, and quality problems. The purpose of the RAMS tool is to support the operator who deals with undesired events or abnormal situations that cause plant operations to deviate from the normal acceptable operating range.

The main task that operators perform at manufacturing plants is to control the process in terms of productivity and with attention to issues related to the protection of personnel, quality, the environment, and safety and health (QESH). Understanding the normal and abnormal behavior of the manufacturing process in all of its operating modes becomes an increasingly more important task. Operators are mainly trained to understand the process in terms of productivity, i.e., converting raw materials into semi finished or finished products in an efficient and cost-effective manner. Knowing

what to do in undesired situations, or recognizing process behavior, which can lead to situations jeopardizing the health and safety of people, the environment, or the quality of a product, is a task that is largely underestimated and often unknown.

When off-normal situations occur, operators rely on process signals (field information) and automated alarm signals to judge the situation based on their training and operating experience. These off-normal situations or conditions generate voluminous information that needs to be analyzed and interpreted by qualified personnel. It is the responsibility of the operator to determine the seriousness of a situation and initiate the appropriate response, if necessary. This diagnostic and decision-making process is based on composite, and typically complex information, comprised of alarms of various levels and the values of related process variables; it is carried out by operators of diversified background, training, and mental model of the system [86].

Yang explains why a decision-making process like this has several disadvantages [86]. First, the availability of experts that fully understand the process may depend on conditions such as work shift, vacations, and personality and personnel related issues. Second, operator training is primarily focused on how to handle standard productivity and quality situations and procedures; operators are less prepared to handle rare or unusual situations. For example, events like equipment failure, wrong software set points due to human error, or multiple simultaneous alarms can easily compound the difficulty of human decision-making, considering the stress level of the operator during undesired and uncertain situations. In situations of industrial accidents, typically, many alarms occur simultaneously and the process generates an overflow of information. Unless properly trained and supported by appropriate means operators may be driven to the wrong rather than the right decisions. Third, the operator's logic model of the process can be incorrect. Especially when manufacturing plants are complex and the amount of information available to the operator increases and changes quickly, it is difficult to solely rely on human interpretation and decision-making processes.

As a fourth reason should be added that operators tend to ignore frequent alarms or can handle only a number of alarms at the same time. If a certain abnormal condition occurs frequently, operators become familiar with the situation and the alarm signals are continuously ignored or totally switched off, i.e., the visible and audible features of the alarm are disabled. Repeatedly this is done to meet production goals or simply because the operator knows how to handle the situation and prevent a serious deviation. Often, the operator understands the seriousness of an alarm but this does not mean that an alarm should be judged the same at all times. It is the unusual circumstances where the operator does not understand the seriousness or does not know how to judge or prioritize events that can lead to accidents or loss of production.

At all times during the operation of a plant an operator must be able to understand the current and possible future behavior of the plant. Based on the current situation and the possibility of future events an operator must be able to make operational decisions. Making decisions on the significance of events, their importance and ranking is a task that is currently not systematically being carried out. Time constraints, production goals, and human limitations do not always allow operators to discriminate critical from non-critical events and take the correct course of action. Therefore, it is important to develop knowledge based and risk informed systems to aid the operators in their function within manufacturing plants.

Although the focus in the thesis is on safety the alarm management system can be used to improve safety as well as productivity of the plant. Productivity is addressed by improving the uptime of the process - or otherwise stated by addressing situations that can result in downtime. Safety is addressed by preventing undesired consequences relating to the protection of people, the environment, and capital equipment.

In order to realize an alarm system that can address productivity and safety it is necessary to incorporate elements of diagnosis, interpretation and prediction. An alarm system that performs these three elements can infer probable causes of system malfunctions and interpret off-normal situations. It can also guide the operator in the course of action as it allows more rapid and detailed analyses of problems in a timely manner and reduces errors in human judgment.

The alarm system needs to incorporate a diagnostic function to infer probable causes of undesired system behavior. A diagnosis is based on situation descriptions, behavior characteristics, or knowledge about the system. A common way to carry out a diagnosis is rules based. Each rule consists of one or more premises (or a condition, a cause, or a symptom) and a conclusion (or an action, or consequence). An example of a diagnostic rule is:

```
DIAGNOSTIC RULE
IF      a) cooling water is not available AND
        b) agitator is not running AND
        c) concentration of PETN is high
THEN the temperature in the tank will go high
```

The alarm system needs to consist of a knowledge base of diagnostic rules that represents all possible combinations of conditions or events that can lead to undesired plant behavior. How this knowledge base is derived for the alarm management system is explained in section 6.7.3.

In real life, not all the information that is required to make a diagnosis will always be available in a manner that it can be used for diagnosis. For example, data that represents possible conditions in a plant can be noisy, conflicting, incomplete, unreliable or even incorrect. This means that in practice it is necessary to make an interpretation of the situation, which deals with understanding the validity or reliability of the collected diagnostic data. Interpretation deals with real data rather then the symbolic representation of the situation, as is the case in a diagnosis. Judgment needs to be made whether the collected diagnostic data is usable. Although, the diagnostic rules might be clear and straightforward, an interpretation system needs to be built that collects the actual required data from the field. A more detailed overview of the interpretation system is given in section 6.7.4.

Once the diagnostic data is interpreted, a prediction needs to be made on how likely it is that the current plant condition leads to an unacceptable deviation in QESH. The prediction element gives an indication of the likelihood of the given consequences. During operation the conditions in the manufacturing process continuously change and thus so do the conditions in the diagnostic rules. Although the consequences do not change, the remaining likelihood's of the consequences

continuously change. A more detailed overview of the prediction element is given in section 6.7.5.

The alarm system should support the operator by continuously diagnosing the condition of the process and give advice on how to operate the process in terms of parameters that influence QESH. If the operator is aware of the current process condition by interpreting the condition of the plant, and gets advice on how this condition can lead to QESH deviations, it is possible to make prompt decisions to prevent these deviations. An operator must have some sense of prioritization to make or support decisions.

The proposed alarm management system informs the plant operator about the current risks involved in operating the plant, which makes it possible to prioritize, based on this risk. It is possible to express risk qualitatively or quantitatively, and events can be ranked from high risk to low risk, allowing the operator to prioritize and use this prioritization to make risk informed decisions on the next course of action.

6.7.2 The RAMS architecture

The alarm management system carries out the elements of diagnosis, interpretation, and prediction by utilizing dynamic and static building blocks (see Figure 23). The dynamic building blocks carry out three different functions, which are collection of field information, diagnosis and presentation of the results. They are dynamic because the information flow is continuously changing. The static building blocks represent information that is not supposed to change frequently. It can change however due to modifications made to the layout or implementation of the manufacturing process. How to deal with modifications or retrofit to the plant is not further addressed in this document.

The following paragraphs will explain the implementation of the diagnostic system, the interpretation system and the prediction system. The field data collector is not explained in detail as it deals with the technical implementation of collecting data. The field data collector represents sensing elements (like pressure transmitters, temperature sensors, on/off switches, or analog to digital converters) that send their information to a central processing unit. This processing unit prepares the information in the format so that it is useful in the real-time field database.

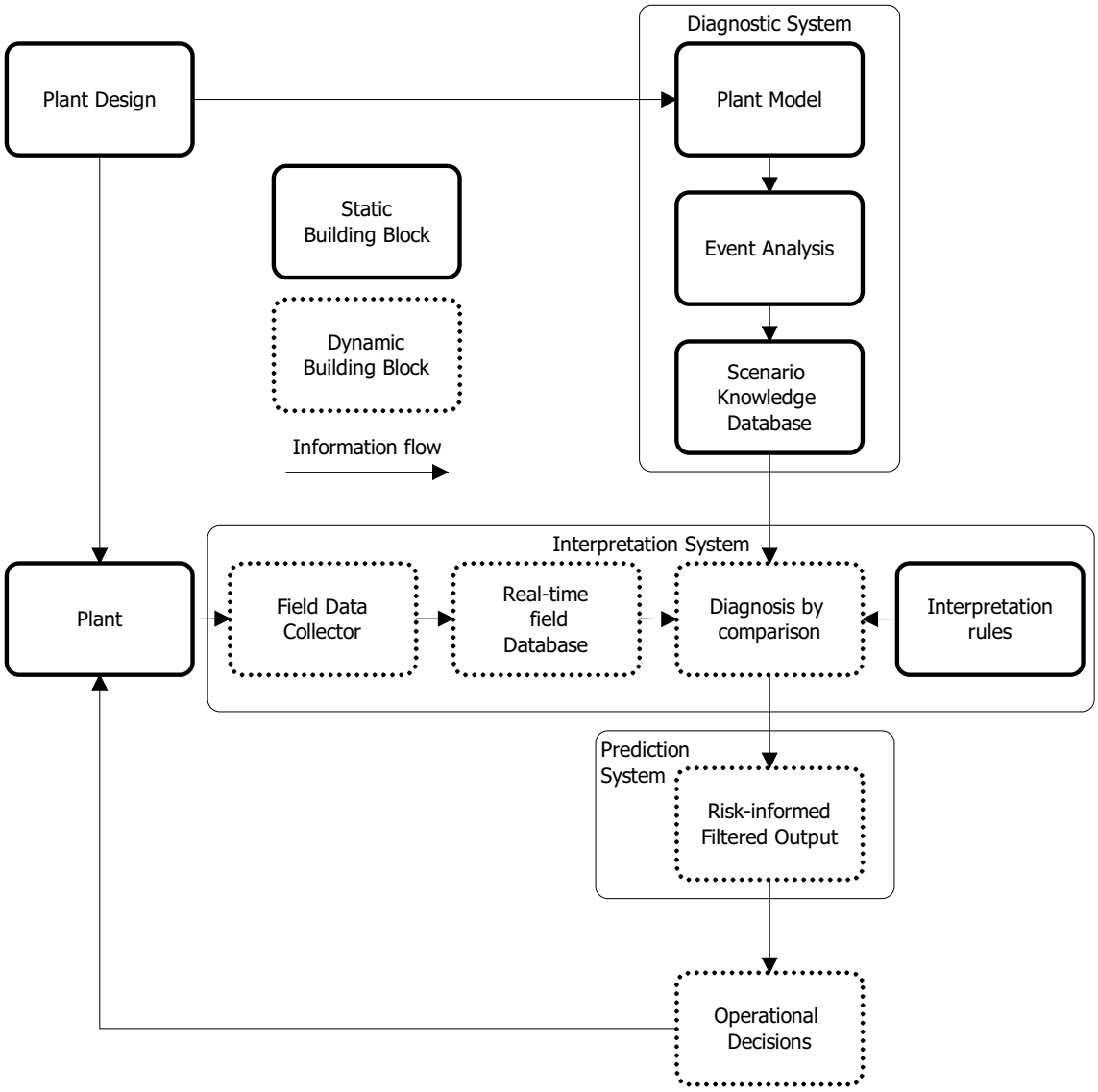

Figure 23. Architecture of the Real-time Alarm Management System

6.7.3 The Diagnostic System

To objective of the diagnostic system is to establish a knowledge base of diagnostic rules. The knowledge base consists of all possible scenarios that can lead to undesired deviations. A scenario is a combination of events that, if all events occur, lead to an unacceptable process deviation. One scenario represents one diagnostic rule. In order to derive all possible scenarios it is necessary to do an event analysis on a model of the plant that can be used to identify these scenarios. Any technique or methodology that can do event analysis and that can derive the possible scenarios can be used to establish the knowledge base. In this thesis, DFM is used to carry out this task. The advantage of DFM is that it is possible to make one comprehensive model of the manufacturing process that includes all possible desired and undesired behavior of the system.

For each specified top event, the deductive capability of the DFM software tool is utilized to derive all possible prime implicants and their associated literals. Each

prime implicant represents a scenario for that particular top event to happen. These prime implicants and the associated literals form the basis for the scenario knowledge base. The scenario knowledge base represents a rules based diagnostic system. The existence of literals plays a major role in the mechanics of the alarm management system. The status of each literal, i.e., whether the literal exists or not, needs to be verified with the actual operational process. The real-time information that is collected from the process needs to focus on every unique literal in the knowledge base.

The DFM model should be derived from the final design of the plant, i.e., the equipment carrying out the manufacturing process. This design is optimized via an iterative process using DFM and the importance measures until the design is acceptable (see Figure 15). When the plant design is finalized it needs to take into account the alarm management system that will actually collect all the diagnostic information in an automated manner. If for example, during one of the iterative design stages, a prime implicant appears that contains a literal that represents a manual operated valve then it will be better to replace this valve by a valve with an automated valve positioner. This to make certain that the real-time alarm management system does not need to depend on human action to update the real-time database.

Although in theory possible, the final design used to establish the alarm management system should not include the (programmable) control and safety instrumentation and logic controllers. The control and/or safety equipment is part of the interpretation system (see section 6.7.4) and is specifically designed to control the process and to collect the necessary diagnostic data. The instrumentation used to collect the necessary diagnostic data will depend on the required information in the scenario knowledge database. If the control and safety equipment is included in the model that forms the basis for the diagnostic system, then additional instrumentation is necessary to collect much more detailed diagnostic information. The diagnostic information would then include also information on the control and safety instrumentation and logic controllers including software variable settings. Including the control and safety system in the final design to establish the diagnostic knowledge base is beyond the scope of this thesis.

6.7.4 The Interpretation System

Although, the diagnostic rules might be clear and straightforward, an interpretation system needs to be built that collects the actual required data from the field, interprets the validity of this data, and makes the actual diagnosis. Interpretation deals with real data rather then the symbolic representation of the situation, as is the case in a diagnosis. The interpretation system, as presented in Figure 23, continuously collects valid or reliable diagnostic data from the field, maintains the real-time field database, and compares the scenario knowledge base with the real-time field database to diagnose the current condition of the plant. The validity or reliability of the collected diagnostic data is determined by the interpretation rules.

The field data collector represents sensing elements (like pressure transmitters, temperature sensors, on/off switches, or analog to digital converters) that send their information to a central processing unit. This processing unit prepares the information in the format so that it is useful in the real-time field database. The field data collector is based on the required info in the scenario knowledge base. The level of detail

concerning the collected data depends on the level of discretization chosen for the variables when the DFM model was created. See for example Table 31, which demonstrates that the required level of discretization puts demands on the required field data collector. Temperature sensor 1 can be a simple on/off switch based sensor while temperature sensor 2 needs to be an analog sensor that can measure the temperature over a certain range.

Table 31. Example discretization

	Temperature Sensor 1	Temperature Sensor 2
States	T1 <50 °C	0 < T2 ≤ 25
	T1 ≥ 50 °C	25 < T2 ≤ 50
		50 < T2 ≤ 75
		75 < T2 ≤ 100

The field data collector stores the collected information in the real-time field database. This database is continuously updated as the process changes. The database is a collection of variables and their possible states, as they exist in the scenario knowledge base. There are only three pieces of information required in this database. These are the variable name, the possible states of the variable, and whether these states are currently present in the plant. If this information is available it is possible to make a diagnosis by comparison.

The information that is stored in the real-time field database is compared to the information in the scenario knowledge base. As this comparison is done real-time, it will lead to a continuous update of the current operational condition of the plant. It acts as an automatic and continuous diagnosis of the plant. There are two issues related to making the diagnosis. First of all there is the issue of trustworthy data. This is an issue that needs to be addressed by the interpretation system and will be described in the following sections. Second, there is the issue that a top event will only occur if all literals of a prime implicant exist or in other words if all events in a scenario occur. It is the objective of the alarm system to prevent the top event from occurring. How this information can be used from preventing the top event to happen is the topic of section 6.7.5.

It is only possible to make the correct diagnosis if the collected field data can be trusted. The question is "How can one validate that the collected data necessary for diagnosis is useful?" There are three ways to judge the confidence one has in the collected field data. These three confidence methods are:

1. Utilize reliability metrics; OR
2. Utilize the online diagnostic capability of programmable electronic systems; OR
3. Utilize physical rules and analytical models.

First of all there is the reliability of the interpretation system itself. One is more confident about the collected field data if the interpretation system is highly reliable. It is possible to express each literal that needs to be measured in the field with reliability metrics. Physical equipment is necessary to collect the field information and therefore it is possible to express the reliability in terms of probability of failure of this

equipment. If for each literal in a prime implicant the probability of failure to collect is known it is possible to calculate the probability of failing to interpret the diagnosis for this prime implicant. See for example Table 32.

Table 32. Probability of Failure Literals

Prime Implicant ##	Probability of Failure
Literal 1	$P_{Literal\,1}$
Literal 2	$P_{Literal\,2}$
Literal 3	$P_{Literal\,3}$

The probability that the interpretation for this prime implicant fails can be expressed as:

$$P_{Prime\ Implicant\ \#\#} = 1 - \left(1 - P_{Literal\,1}\right) \cdot \left(1 - P_{Literal\,2}\right) \cdot \left(1 - P_{Literal\,3}\right)$$

Or in general terms for a prime implicant:

$$P_{Prime\ Implicant\ k} = 1 - \prod_{i=1}^{n} \left(1 - P_{Literal\ i}\right)$$

Where *i* is the current literal and *n* the total number of literals in prime implicant *k*. The alarm system fails if the interpretation of one of the prime implicants fails. This can be expressed as follows:

$$P_{Interpretation} = 1 - \prod_{k=1}^{m} \left(1 - P_{Prime\ Implicant\ k}\right)$$

Where *k* is the current prime implicant and *m* the total number of prime implicants in the scenario knowledge base.

The probability of failure serves as a confidence metric. The more reliable the interpretation system, the more confident one is in believing the collected diagnostic data. Section A.1.1 explains a framework that can be used to calculate the probability of failure. It is based on the concept of Safety Integrity Level (SIL) calculations, which is used in the safety community to express the level of confidence one has in a system. Section A.1.3 explains in detail what a SIL level is, how SIL levels can be calculated and what the influence of design parameters, like for the architecture, hardware, software or online diagnostics capabilities is on the SIL. Depending on the criticality of the manufacturing process it is possible to setup rules that determine the level of reliability required for each field diagnostic measurement. Either a measurement is accepted based on the fact that the calculated reliability metric exceeds the specified limit or if the measurement reliability is below the specified reliability metric and the measurement shows up in a prime implicant that deems to be critical one will have to verify the validity of the measurement using additional means. This can be expressed in the following interpretation rule that needs to be executed for each literal.

INTEPRETATION RULE

IF reliability metric for literal X is above specified limit

THEN accept interpretation

ELSE look for additional supporting evidence.

The first provided solution to look for supporting evidence is based on the possible technical capabilities of programmable electronic systems. Programmable electronic systems lend themselves to incorporate online diagnostic. The online diagnostics are self-checks that are continuously executed to monitor the correct functionality of the programmable electronic system. For example, a watchdog timer can monitor whether a microprocessor stopped executing its program. An output feedback loop can monitor whether an output is frozen in a stuck at 0 or 1 position. An interpretation system that is based on programmable electronic technology can utilize these online diagnostic capabilities to its advantage. It is possible to establish rules that include the diagnostic capability and support the interpretation only if the diagnostic system does not support any errors. An example of an interpretation rule like this is presented next.

INTERPRETATION RULE

IF a) cooling water is not available AND

b) agitator is not running AND

c) concentration of explosive material is high AND

d) online diagnostics of interpretation system reports no errors

THEN the temperature in the tank is high

If the diagnostics system reports an error then it is possible to ignore this diagnostic reading, as it is likely that there is an error in reading the required field data. Other procedures can be in place to handle the failed diagnostic reading.

Supporting evidence can also be gathered by applying physical rules using analytical models. Whether field data is correct can be verified by looking for additional indicators that support the measurement reading. For example, the previous diagnostic rule requires the condition "no cooling water" to be true. It is possible to verify this condition by looking at other supporting evidence. See for example the next, very simple, interpretation rule that verifies whether cooling water is available:

```
INTERPRETATION RULE
IF     a) cooling water tank full AND
       b) supply valve 1 is open AND
       c) cooling water control valve 1 is open AND
       d) no leaks are reported
THEN cooling water is available
```

This rule can be used to validate the cooling water condition. If the cooling water reading presents one of the conditions in a diagnostics rule then the interpretation rule would support the correctness or incorrectness of the condition. The rules that are derived in this manner are highly depended on the physical aspects of the manufacturing process and can be derived by using the analytical models, including DFM models.

The interpretation system can consists of a knowledge base with rules that are based on the reliability of the interpretation system, the physical rules of the process, or a combination of both. This interpretation knowledge base is then integrated with the diagnostic knowledge base. Only if the interpretation rules support the diagnostic conditions or events it is possible to make a prediction of the diagnosis. In this thesis the trustworthiness is based on the reliability of the interpretation system.

6.7.5 The Prediction System

The described alarm system so far consists of a knowledge base of diagnostic rules, which represent scenarios that can lead to deviations in QESH and an interpretation system that collects field diagnostic data and validates the confidence we have in this data. The next step in the alarm system is to manage the collected information and make predictions on the future behavior of the plant.

The scenario knowledge base can easily consist of thousands of diagnostic rules that need to be monitored by the operator. The operator needs to be able to make operational decisions based among others on the information that can be retrieved from the alarm management system. The overwhelming amount of information that is available needs to be presented in a way that gives the operator a sense of priority. An effective alarm system would give operators guidance on how “close” the top event is about to happen. By monitoring the existence of the prime implicants’ literals in the field, as they exist in the knowledge base, it is possible to make this judgment.

The knowledge base can hold thousands of prime implicants. Each of these prime implicants will contain one or more literals. If a prime implicant is comprised of ten literals and seven of these literals actually exist in the plant then only three more literals need to occur for this prime implicant to be true and thus for the top event to happen. If we only look at the remaining literals to occur then this prime implicant could be far more significant to an operator then another prime implicant with ten literals where only four of these literals exist in the manufacturing plant. Therefore, by monitoring the existence of literals, and their roles in prime implicants, it is possible to give operators guidance on how to operate their processes in the manufacturing plant. If it is possible to tell the operator which prime implicants are “close” to happen

and which ones or not likely to happen then the operator knows why and where to put the focus when operating the plant.

The definition of “close” is not just as easy as counting the remaining literals that need to occur, although this is one way of doing it. It is impossible to judge whether three remaining literals are more important than seven remaining literals, purely based on the number of remaining literals. In practice, a knowledge base will consist of prime implicants that have a wide variety of numbers of literals. Therefore, comparing prime implicants of different size purely on the number of literals that do not exist in the field will not always make sense and will not give the operator an answer to what is truly important to focus on. If prime implicant A consists of ten literals and prime implicant B consists of thirty literals and both prime implicants have five literals left that need to occur in order for the prime implicant to exist then there is no way that an operator can make a sound judgment based on this information and decide which of these prime implicants is more important.

This problem can be addressed by examining the contents of prime implicants. If an operator has the ability to examine the contents of the prime implicants during operation then there the following valuable information can be derived:

1. The operator will see what has happened already, as a number of the literals will actually exist in the field.
2. The operator will see what still needs to happen in order for the top event to occur, as a number of literals do not exist yet.

This is extremely valuable information when operating a manufacturing plant. If it is possible to calculate the probability of occurrence of the remaining literals to occur then it is possible to prioritize the prime implicants based on this probability. By ranking the prime implicants from a high-to-low probability, the operator will know where to focus the attention when operating the plant. Calculating the remaining probability of occurrence allows the operator to compare prime implicants, independent of their size and content. Quantification based on probability would be an ideal solution as it allows objective comparison.

If quantification is not an option then the operator can base the decision-making process on the contents of the prime implicants. The operator needs to examine the actual contents of the prime implicants and prioritize the prime implicants based on what the remaining literals represent. This is less objective than the probabilistic approach. A typical prime implicant will consist of literals that represent the state of process parameters and the state or condition of equipment. The operator can decide to give more priority to prime implicants where the remaining literals are only process conditions, or only represent equipment conditions, or any combination of interest. The following section will go more into the details of probabilistic and non-probabilistic prioritization.

6.7.6 Probabilistic prioritization

The probabilistic method of prioritizing prime implicants by importance is based on the concept of residual probability. The residual probability can be calculated if the probability of occurrence of the literals is known. The residual probability calculates the probability of occurrence of the remaining literals, i.e., those literals that do not exist yet in the field. This is one way of representing how “close” something is about to happen that influences QESH. A prime implicant has a certain probability of

occurrence but only exists if all literals of the prime implicant are true. As the existence of literals will change frequently during the operation of the plant, the residual probability will have to be updated real-time to present an accurate picture of the condition of the plant.

If the residual probability is available for all prime implicants in the knowledge base, it is possible to rank prime implicants based on their residual probability. The lower the residual probability the less important the prime implicant is in terms of the specified top event. The higher the residual probability the more likely it is that the prime implicant will occur. It would be possible to establish rules that an operator can use to decide when to pay attention to a prime implicant or when to take action that influences the operation of the plant. For example, one rule could be to examine all prime implicants that exceed a certain limit of probability. Another rule could be to always examine the top 10% in terms of residual probabilities of all prime implicants. The alarm system would give an alarm as soon as one ore more prime implicants would meet the specified rules.

Those prime implicants that are of importance to the operator give the operator extreme valuable information. First, the operator will have a sense of urgency if the residual probability is suddenly higher then normal during operation. Even though nothing has happened yet in terms of QESH, the operator knows to be extra alert. Second, the operator can examine the prime implicant(s) that caused the alarm. Not only will the operator know what needs to happen next (and what the likelihood is that it will happen) but also what has happened already. If what has happened represents, for example, an equipment failure then the operator can put priority on repairing this equipment and thus reduce the residual probability. On the other hand, the operator also knows what needs to be prevented. The operator has a clear understanding of the condition of the plant and can use this information to operate the plant more effectively.

6.7.7 Non-probabilistic prioritization

Because of lack of reliable data quantification might not be an option. It is still possible to use the operator needs to understand the true meaning of the contents of a prime implicants in order to be able to make decisions on the course of action when operating the plant. The operator needs to examine the actual contents of the prime implicants and prioritize the prime implicants based on what the remaining literals represent. Non-probabilistic prioritization is based on the amount of remaining literals or the interpretation of the true meaning of the remaining literals. The alarm management system can use the non-probabilistic prioritization techniques as explained in section 5.4.

As qualification is based on any kind of information of interest, it will be possible to apply any prioritization of interest. Using the non-probabilistic importance measures the alarm system can be based on one measure or a combination of measures. For example, an alarm can be given if the percentage of remaining literals in a prime implicant reaches less than 10%. Or, if one of the remaining literals would represent an equipment failure while the other remaining literals represent process variables. The actual rules for prioritization that need to be set up depend on the particular manufacturing process, the operating guidelines and the experience with the process.

6.7.8 Uncertainty in RAMS

Many aspects determine the quality of the real-time alarm system. First of all, there are many steps involved in establishing the real-time alarm systems, see Figure 23. In each of these steps, it is possible to introduce mistakes that can influence the effectiveness of the alarm system and introduce uncertainty in the output of the system. In order to minimize uncertainty with the alarm system it will be necessary to put the development of the alarm system under an effective quality assurance system that emphasis testing, verification and validation activities. Uncertainty is mainly associated with:

- The model of the plant;
- Event analysis which determines the knowledge base;
- The ability to collect field data;
- Speed with which field data can be collected and processed;
- The risk-based prioritization;
- The quality of the decisions;

The level of detail and the quality of the model of the plant has probably the most impact on the effectiveness of the alarm system. Not enough detail will certainly lead to an alarm system that will not be able to handle all undesired situations simply because it will not be possible to capture all details that can lead to deviations in QESH. To much detail will make the model to complex in terms of collecting data from the field. The more detail is put in the more sophisticated the data collection needs to be. Uncertainty is also introduced, as the model as well as the actual plant will be based on the final plant design.

The model will be used to derive the scenarios that can lead to deviations in QESH. These scenarios built the knowledge base. Only if all scenarios are identified that can influence QESH the knowledge base will be complete. Missing an important top event will definitely influence the performance of the alarm system.

The level of detail of the model, the event analysis, and the resulting number of scenarios determine the size and the amount of data that needs to be collected from the field. The ideal situation would be to collect the necessary information to feed the knowledge base in an automated manner, i.e., using a programmable electronic system. If it is not possible to collect data using a programmable electronic system, it will be necessary to carry out the task by hand. Updating the real-time database now depends on how well the operator performs this task. The ability of the system used to collect data is another factor that will determine the quality of the alarm system.

The strength of the alarms system is that it is real-time. If data is not updated regularly then this will influence the performance of the alarm system. Even though this should not be a problem for programmable electronic systems, as they should easily be able to update information within seconds, the limiting factor over here is the data that needs to be updated manually. For example, valves that are operated manually.

If the alarm system is up and running the results need to be presented based on risk-based prioritization. If probabilistic prioritization is used then the quality of the results depends on the quality of the data and the probability calculations. If non-

probabilistic results are used then the results depend on the operator's judgment and how well the prioritization is applied.

6.8 Conclusions

This chapter applied the importance measures throughout the lifecycle of a plant. Consecutively the importance measures are used to aid the design of the manufacturing process, the identification of safety functions, the design of the safety system, and the identification of the diagnostic systems. The process and safety system are in first instance designed and verified independent of each. Once this design is approved the design of the safety system is integrated into the design of the process plant. This allows the analyst to validate the design of the safety system in the context of the operating plant. From this integrated model diagnostic functions are derived that monitor the correct behavior of the safety functions. A real-time alarm management system is introduced to guide the operator to make risk informed decision that support safety during the operation of the process. The alarm management system is based on the concept of residual probability.

Chapter 7 Illustration Using The Practical Example

7.1 Introduction

The purpose of this chapter is to demonstrate, by examples, the theories presented in Chapter 5 and Chapter 6. The usefulness of the importance measures and the real-time alarm management system will be demonstrated using the process and safety system described in Chapter 3. A description of DFM is given in Chapter 4. An overview of the importance measures is given in Chapter 5. It is not the purpose to present a complete safety case but rather to demonstrate how the importance measures can be used to prioritize important safety information.

7.2 Example deductive analysis drain valve

The following is an example of a deductive analysis of the drain valve section of the PETN manufacturing process. The drain valve is presented in Figure 24 and the corresponding DFM model is presented in Figure 25. A complete description of the operation of the drain valve is give in Section 4.3.

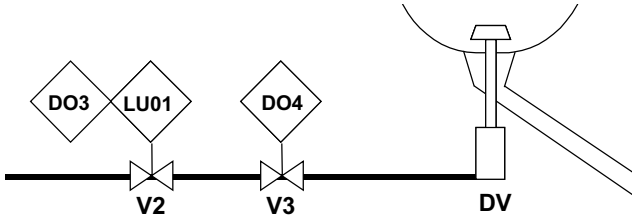

Figure 24. Hydraulic operated drain valve

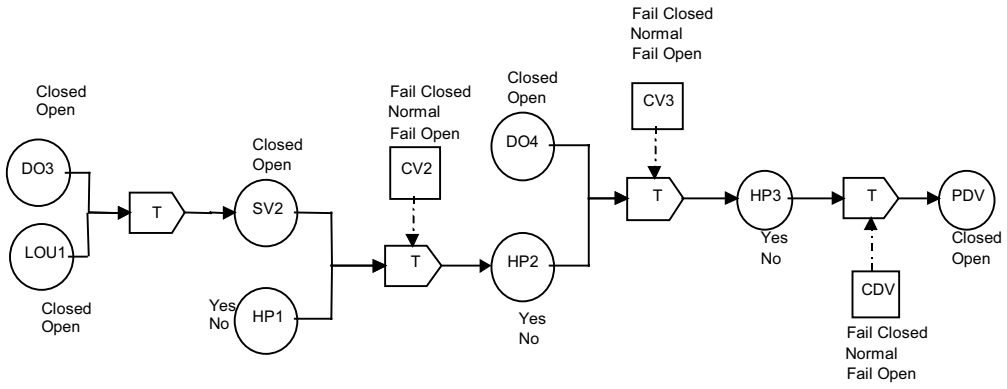

Figure 25. DFM model drain valve

The drain valve needs to be opened, if the temperature in the nitrator tank reaches an undesired limit, in order to dump the hazardous material into the drowning tank. From a safety point of view it is of interest to investigate under which

conditions the drain valve cannot open[6]. The following top event reflects this situation:

```
Top Event
At time 0  ,   PDV = Closed (DV closed)
```

This top event results in the following 5 prime implicants:

```
Prime Implicant #1
At time 0  ,   DO4 = Open  (Open valve)              AND
At time 0  ,   HP1 = Yes   (Available)               AND
At time 0  ,   DO3 = Open  (Open valve)              AND
At time 0  ,  LUO1 = Open  (Open valve)              AND
At time 0  ,   CV2 = Normal (Operates normally)      AND
At time 0  ,   CV3 = Normal (Operates normally)      AND
At time 0  ,   CDV = Normal (Operates normally)

Prime Implicant #2
At time 0  ,   DO4 = Open  (Open valve)              AND
At time 0  ,   HP1 = Yes   (Available)               AND
At time 0  ,   CV2 = Open  (Stuck open)              AND
At time 0  ,   CV3 = Normal (Operates normally)      AND
At time 0  ,   CDV = Normal (Operates normally)

Prime Implicant #3
At time 0  ,   HP1 = Yes   (Available)               AND
At time 0  ,   DO3 = Open  (Open valve)              AND
At time 0  ,  LUO1 = Open  (Open valve)              AND
At time 0  ,   CV2 = Normal (Operates normally)      AND
At time 0  ,   CV3 = Open  (Stuck open)              AND
At time 0  ,   CDV = Normal (Operates normally)

Prime Implicant #4
At time 0  ,   HP1 = Yes   (Available)               AND
At time 0  ,   CV2 = Open  (Stuck open)              AND
At time 0  ,   CV3 = Open  (Stuck open)              AND
At time 0  ,   CDV = Normal (Operates normally)

Prime Implicant #5
At time 0  ,   CDV = Closed (Stuck closed)
```

If there is only a limited number of prime implicants like in this case then it is possible to investigate each single prime implicant and make recommendations to resolve the issue. For example, in this case the prime implicants 2, 3 and 5 represent single hardware failures. Prime implicant 4 represents a case where both valves are stuck open. This could be viewed as a common cause failure problem if both valves

6 Note that not being able to open the drain valve is a safety issue as well as quality issue. If the drain valve cannot be opened then the product cannot be delivered. This is an issue that needs to be addressed to achieve quality.

are the same. Prime implicant 1 shows no failures. The drain valve is closed because of signals given from the BPCS and safety system. In this case, we need to further investigate the hardware and software used to drive the valves and see under what conditions both signals DO3 and DO4 would represent “Open Valve” and thus keep valve V3 and V4 open (which means they deliver hydraulic pressure to the drain valve).

The drain valve needs to stay closed during normal operation. It only needs to be opened when the mixing process has been completed or in case of a necessary emergency dump. Opening the drain valve at an undesired moment should be prevented. From a safety point of view, it is of interest to investigate under what conditions in the plant the drain valve opens[7]. This situation is reflected in the following situation:

```
Top Event
At time 0  ,   PDV = Open  (DV open)
```

This top event results in the following 7 prime implicants:

```
Prime Implicant #1
At time 0  ,   DO4 = Close (Close valve)           AND
At time 0  ,   CV3 = Normal (Operates normally)    AND
At time 0  ,   CDV = Normal (Operates normally)

Prime Implicant #2
At time 0  ,   CV3 = Closed (Stuck closed)         AND
At time 0  ,   CDV = Normal (Operates normally)

Prime Implicant #3
At time 0  ,  LUO1 = Close (Close valve)           AND
At time 0  ,   CV2 = Normal (Operates normally)    AND
At time 0  ,   CDV = Normal (Operates normally)

Prime Implicant #4
At time 0  ,   DO3 = Close (Close valve)           AND
At time 0  ,   CV2 = Normal (Operates normally)    AND
At time 0  ,   CDV = Normal (Operates normally)

Prime Implicant #5
At time 0  ,   HP1 = No    (Not available)         AND
At time 0  ,   CDV = Normal (Operates normally)

Prime Implicant #6
At time 0  ,   CV2 = Closed (Stuck closed)         AND
At time 0  ,   CDV = Normal (Operates normally)
```

[7] Note that opening the drain valve at an undesired time is a safety issue as well as quality issue. If the drain valve opens it will dump the material to the drowning tank. The material will be lost, which is quality problem.

```
Prime Implicant #7
At time 0  ,   CDV = Open  (Stuck open)
```

In this case there are four prime implicants that represent single points of failures (2, 5, 6, and 7). Prime implicant 1, 3 and 4 represent signals from the BPCS or safety system that closes the valve. As can been seen from these two simple examples, the prime implicants can contain a wealth of information that needs to be investigated and understood in order to be of true value for safety analysis. It is important that the drain valve only operates when desired. From the design point of view a more sophisticated hydraulic system could be recommended, where a single failure will not lead to a stuck open or stuck closed drain valve.

One can imagine that if a complete manufacturing process is modeled that examining all possible prime implicants can be to complex or time consuming. The following section will demonstrate how the importance measures of Chapter 5 can be used to analyze a complex manufacturing process as presented in Chapter 1.

7.3 Analyzing the PETN manufacturing process

With the DFM model in Figure 11 it is possible to carry out a multitude of analysis. The DFM only includes the equipment necessary to carry out the manufacturing process and the intended signals form the BPCS or manual signals from an operator. The BPCS design is not included. As the temperature in the tank plays a critical role, it would be obvious to select as top event the condition that the temperature in the tank equals or exceeds 35 °C (High-High). With this as top event, the DFM tool generates all possible conditions, represented by the prime implicants, which can cause this top event.

For the purpose of this work, different top events have been chosen to demonstrate the capability of DFM as an analysis tool. As the top event can be any condition of interest, and does not necessarily need to be a failed or undesired condition, one of the top events that has been used is presented next:

```
Top Event #1
At Time =  0,   TT = High-High (Temp Tank is High-High)   AND
At Time = -1, PETN = 6% (Concentration PETN is 6%)
```

This top event examines whether a situation is possible where the temperature in the tank reaches 35 °C (High-high) and the concentration of PETN in the tank is 6%. A consistency rule has been applied to simulate as if the cooling water section is working properly.

```
Consistency rule #1

At all times NCW = Pos (Positive flow)
```

The NCW (Net Cooling Water) variable has been made constant by setting it to a positive flow. This indicates that there is always cooling water flowing around the

tank. The difference between defining a consistency rule and not defining it as a condition in the top event is that an analysis will not include any prime implicants that represent conditions of the cooling section. In this way, the analyst can focus the analysis specifically on certain subsections of the system.

This top event created 1292 prime implicants containing 12628 literals. On average, that is about 10 literals per prime implicant. There are too many prime implicants for an analyst to examine by hand. By using the different importance measures the analyst can prioritize the prime implicants, and focus on those prime implicants that matter most.

7.3.1 Measure: Content of the prime implicant

This importance measure allows the analyst to prioritize prime implicants based on their content (see section 5.4.2). The analyst can apply a variety of importance measure. For example, using top event #1 with consistency rule # 1 and prioritizing for single failures states, leads the analyst, among others, to the following prime implicant.

```
Prime Implicant #87
At time -2 ,   WPE = 4     (4 units)                    AND
At time -2 ,   CNF = High-high (Stuck high-high)        AND
At time -2 ,    BW = Normal (Normal)                    AND
At time -2 ,   DO6 = Off   (No speed)                   AND
At time -2 ,   DO7 = Off   (Speed off)                  AND
At time -2 ,   CAM = Normal (Operates normally)         AND
At time -2 ,  PETN = 1%    (1 %)                        AND
At time -2 ,    TT = Low   (Low)                        AND
At time -1 ,   DO6 = Off   (No speed)                   AND
At time -1 ,   DO7 = Off   (Speed off)                  AND
At time -1 ,   CAM = Normal (Operates normally)
```

This prime implicant represents a situation where the nitrator feeder is stuck in the highest speed position. This means that the entire PE available in the weigh funnel is fed, with the highest speed, into the nitrator tank. Even though at time –2 the concentration of PETN (1%) and the temperature (Low) in the tank is still low, within two time steps it is apparently enough to increase the temperature in the tank as also the mixer is not yet activated. This prime implicant can be prevented by either monitoring the condition of the nitrator feeder (see section 7.6.1 on diagnostics) or by implementing a safety interlock (see section 7.4 on safety functions).

To focus also on the cooling water section the following top event has been specified.

```
Top event: #2
At time 0  ,    TT = High-high (High-high)
```

The following consistency rule has been applied to assure that the concentration of PETN is always 6%. This ensures that the focus is on the agitator section, drain valve section, and the cooling section only.

```
Consistency rule #2
At all times PETN = 6% (Concentration)
```

The content measure is applied twice to prioritize for prime implicants with human interaction but without failure states. This combination results in prime implicants where none of the hardware equipment has failed but where manual operation is involved that apparently leads to a too high temperature in the tank. This leads the analyst, among others, to the following prime implicant:

```
Prime Implicant #42
At time -5 ,    M2 = Closed (Manual closed)           AND
At time -5 ,   CV6 = Normal (Operates normally)      AND
At time -5 ,   DO6 = Off   (No speed)                AND
At time -5 ,   DO7 = Off   (Speed off)               AND
At time -5 ,   CAM = Normal (Operates normally)
```

This prime implicant shows that the Valve 6 of the cooling section was manually closed. A process engineer might have closed the valve for maintenance reasons but forgot to open it again. Closing valve 6 means that there is no cooling water flow even though cooling water is available. A design recommendation can be made that there should be no manual valves, or that the flow of the cooling water should be monitored as a safety function (see section 7.4.2 on safety interlock functions).

7.3.2 Measure: Number of literals in prime implicant

This importance measure helps prioritizing the total prime implicants based on the number of literals in the prime implicant. For the top event #1 there are 1292 prime implicants. Table 33 gives an overview of the number of literals in prime implicants.

Table 33. Overview number of literals in prime implicants

# Of literals	# Of prime implicants	% Of total	Cumulative %
0	0	0	0
1	0	0	0
2	1	<1	<1
3	1	<1	<1
4	2	<1	<1
6	97	8	8
7	41	3	11
8	195	15	26
9	242	19	45
10	225	17	62
11	249	19	82
12	125	10	91
13	79	6	97
14	7	<1	98
15	21	2	99
16	7	<1	100

Design rules can be set up that specify for example that the top 10% of the prime implicants with the lowest number of literals needs to be examined. In this case that means that only 142 out of 1292 prime implicants with a maximum of 7 literals need to be examined. Or, the design rule might specify that all prime implicants with less then 10 literals need to be examined. In this case this results in 579 out of 1292 prime implicants. This importance measure can help the analyst prioritize the prime implicants for further analysis. In the existing example that would mean that 713 prime implicants do not need to be addressed. The 579 prime implicants can be examined by hand or other importance measures can be used to further examine these prime implicants.

7.3.3 Measure: Number of variables or states

A very simple but effective importance measures is the ranking based on the number of occurrences of a variable or variable state. The number of times a variable or variable state is present gives an indication of the contribution of this variable or variable state to the top event.

Table 34 gives an overview of the number of times a variable appears in the collection of prime implicants for top event #1. The condition of the agitator motor (CAM) appears the most, with 2702 occurrences. The agitator motor is necessary to mix the nitrating acid with the PE in the tank. The digital output signal DO8 appears the least, with only 28 occurrences. This signal opens the valve controlling the flow of nitrating acid into the tank. This table gives the analyst a way to prioritize the focus on variables (process equipment, process conditions, signals etc.). From this table it

can be concluded that the agitator motor plays a very important role in achieving the high temperature in the tank. The agitator motor is necessary to mix the contents of the nitrator tank. If the contents are not mixed well, it is possible to have local high concentrations of PETN. This does not yet give any understanding about what to do with the agitator motor. It now needs to be investigated further how the agitator motor influences the high-high temperature in the tank as it cannot be removed itself.

Table 34. Number of variables

Name	Description	# Of occurrences
CAM	Condition Agitator Motor	2702
PETN	% of PETN	2584
CNF	Condition Nitrator Feeder	1291
WPE	Weight PE	1289
BW	Band With Feeder	1288
TT	Temperature tank	1148
DO7	Digital Output 7	658
AO1	Analog Output 1	645
DO1	Digital Output 1	645
DO6	Digital Output 6	294
CFV	Condition Feed Valve	56
DO8	Digital Output 8	28

Table 35 gives an overview of the number of occurrences of variable states. The top two positions are held by variable states of the agitator motor. The lowest position is held by the condition of the nitrator feeder (Low-low). This table gives the analyst again a sense of prioritization. The first agitator state, i.e., agitator normal, appears 1274 times. At first this does not makes sense, as one does not expect the desired normal state to have such a high impact, and thus this requires further investigation.

The condition of the agitator motor influences directly the speed of the agitator. Because it functions without any problems, something else in the system must be influencing the speed of the motor. If the agitator motor is functioning normally then the speed is actually determined by the BPCS, as the BPCS operates the motor. There might be many reasons why the actual speed does not achieve good mixing. The control logic (software) might be wrong, an operator might have switch the agitator of, or there might be a failure in the signal path.

A more obvious and understandable situation is the second largest occurrence, i.e., the agitator motor has failed stuck off (868 occurrences). This means that no mixing at all takes place.

As this has such a large influence on the temperature the analyst can now make recommendations to improve the design. From a design point of view this information can be used as being a critical signal for the mixing process. A design recommendation can be made to implement a redundant signal path to the agitator motor, or to have feed back to the BPCS that assures that mixing takes place, or a second mixer can be installed in parallel. Monitoring the correct functioning of the agitator can be used to make a decision to stop the process or to open the drain valve immediately if no mixing is detected (see section 7.6.1 on diagnostics). All these conditions can be investigated further based on the initial finding of the normal state being so dominant. Prioritization helped find the variable CAM (Condition Agitator Motor). The interpretation itself requires a knowledgeable analyst.

Table 35. Number of states

Name	State	Description	# Occurrences	Name	State	Description	# Occurrences
CAM	Normal	Operates normally	1274	WPE	2	2 units	184
CAM	Off	Stuck off	868	WPE	>6	>6 units	184
CNF	Normal	Operates normally	646	WPE	3	3 units	184
DO1	On	Speed on	644	AO1	Medium	Speed medium	182
PETN	5 %	5 %	630	CNF	Medium	Stuck medium	182
PETN	4 %	4 %	630	WPE	1	1 unit	180
DO7	Off	Speed off	602	AO1	High	Speed high	112
BW	Small	Small	574	CNF	High	Stuck high	112
CAM	Low	Stuck low speed	532	CNF	High-high	Stuck high-high	105
TT	Low-low	Low-low	504	AO1	High-high	Speed high-high	105
PETN	3 %	3 %	462	TT	Normal	Normal	84
BW	Normal	Normal	434	DO7	High	High speed	56
PETN	2 %	2 %	392	DO6	Low	Low speed	56
TT	Low	Low	294	PETN	0 %	0 %	56
BW	Wide	Wide	280	DO8	Open	Valve open	28
AO1	Low	Speed low	245	CFV	Normal	Operates normally	28
CNF	Low	Stuck low	245	CAM	High	Stuck high speed	28
TT	High	High	238	CFV	Open	Stuck open	28
DO6	Off	No speed	238	WPE	1	1 unit	4
PETN	6 %	6 %	214	PETN	>6 %	>6 %	4
PETN	1 %	1 %	196	WPE	0	0 unit	1
WPE	6	6 units	184	AO1	Low-low	Speed low-low	1
WPE	5	5 units	184	DO1	Off	Speed off	1
WPE	4	4 units	184	CNF	Low-low	Stuck low-low	1

7.3.4 Measure: Relation between variables or variable states

The purpose of this measure is to identify whether there is a relation between the occurrence of certain variables or variable states within the set of prime implicant that belong to a top event. Once other measures are used to prioritize variables or variables states this measure can be used for further investigation. This measure is useful as it might not always be possible to eliminate the main variable or variable state, as is the case with the agitator motor.

Table 36 gives an overview of the most important relation between the agitator motor normal state and other variables and their states. The most dominating variable state is "DO7 = Off" with 543 occurrences. This variable state represents an off signal from the control system for the high-speed position of the agitator motor. If the DFM model had included the signal path to the controller and the software logic of the control then all variables would have been found that could influence this off signal as they indirectly influence the speed of the agitator motor. It might be a hardware failure in the control system, or signal path to the agitator motor, or an application software design error. In this case, the second largest relation is the off state of DO6, which represents the off state of the low speed agitator motor position. Both variables that influence the state (i.e., the speed) of the agitator most, given that the condition of the agitator motor is normal, are found with this measure. This is an excellent measure that helps the analyst understand previous result by investigating them further. As this seems to be a critical signal, design decisions can be made to improve the reliability of these signals.

Table 36. Relationship normal state CAM and other variable states.

CAM Normal state in relationship with other variable states			
Name	**State**	**Description**	**# Occurrences**
DO7	Off	Speed off	543
DO6	Off	Speed off	240
CNF	Normal	Operates normally	224
TT	Low-low	Low-low	223
DO1	On	Speed on	220
BW	Normal	Normal	148
PETN	3 %	3 %	147
PETN	2 %	2 %	122
PETN	1 %	1 %	109
BW	Wide	Wide	104
AO1	High	High	90
CNF	High	Condition Nitrator feeder	89
WPE	1	1 Unit	66

7.3.5 Measure: Time dependence prime implicants

If periodic maintenance is carried out on specific time intervals then it can be assumed that certain undesired conditions in the process would be detected during a scheduled maintenance activity by carrying out functional tests. Prime implicants with

literals occurring after this period can be eliminated from the analysis, narrowing down the number of prime implicants for the analyst to consider. This supports the analyst as it helps prioritizing the prime implicants.

Assume that everyday a batch of PETN is manufactured in 5 time steps[8]. After each batch the process plant is cleaned and procedures are in place that assure that certain equipment is functionally tested. These tests can be considered periodic tests that are carried out at least every 6 time steps. Therefore, prime implicants that contain failure states of equipment that is included in the test will in actuality not occur if the tests have full functional coverage.

To demonstrate this top event #2 has been applied with consistency rule #2. The time dependence measure has been applied to prioritize for all prime implicants that have literals that occur before time step -5. The filter leads the analyst among others to the following two prime implicants.

```
Prime Implicant #88
At time -7 ,    M2 = Open   (Manual open)              AND
At time -7 ,   CV6 = Normal (Operates normally)       AND
At time -6 ,    CW = Yes    (Available)               AND
At time -6 ,   CV5 = Closed  (Stuck closed)           AND
At time -6 ,   DO9 = Open   (Valve open)              AND
At time -6 ,   CV7 = Normal (Operates normally)       AND
At time -6 ,   DO7 = Off    (Speed off)               AND
At time -6 ,   CAM = Normal (Operates normally)

Prime Implicant #40
At time -7 ,    CW = Yes    (Available)               AND
At time -7 ,    M1 = Open   (Manual open)             AND
At time -7 ,   CV5 = Normal (Operates normally)       AND
At time -7 ,   DO9 = Open   (Valve open)              AND
At time -7 ,   CV7 = Normal (Operates normally)       AND
At time -7 ,   OCW = Flow   (Flow to V6)              AND
At time -7 ,   CAM = Off    (Stuck off)
```

Prime implicant #88 consists of a valve (V5) stuck closed at –6. This valve is part of the cooling system, which is tested between two batch processes. Each time, between batches, the operator tests whether cooling water is flowing. If valve V5 is stuck then this would be discovered during this test. Prime implicant #40 consist of an agitator motor that is stuck at off. Between batch processes a test is carried out that visually verifies whether the agitator motor operates on all speeds. If it doesn't, than it needs to be repaired before the next batch process is started. This prime implicant does not need to be investigated as the failure occurs on time step –7 and would thus have been detected between two batches. Using this measure the operator can focus the analysis on those prime implicants that apply. It gives the operator a sense of prioritization.

This approach can also be used to specify periodic tests. Prime implicants can be prioritized depending on the time step that a literal needs to occur. If all prime implicants are identified that occur above a certain time step then it is possible to

[8] Consider for example that each time step represents 1 hour.

examine these prime implicants and specify functional tests that need to be carried out during the maintenance activity.

7.3.6 Measure: Probability Prime implicant

Even though the challenge with the probabilistic importance measure will lay with the required probability data, some of them will be demonstrated to show their potential usefulness. Table 37 gives an overview of three different ways to rank the prime implicants based on their probability of occurrences. Column 1 presents the ID number of the prime implicant as generated by the database software tool that has been developed to analyze the prime implicants. Column 2, 3 and 4 present respectively the

- Probability of occurrence of the prime implicant;
- Ratio between the probability of the prime implicant and the probability of the top event; and
- Ratio between the probability of the prime implicant and the prime implicant with the maximum probability.

This table can be used to apply design rules that for example specify to examine all prime implicants with a probability of occurrence of more then 1.E-11 must be analyzed. This importance measure helps the analyst to focus the analysis as it prioritizes the prime implicants. Only the first thirty-two prime implicants are listed.

Table 37. Probability Prime Implicant Ranking

Prime Implicant	Probability	$I_{PI_i} = \frac{P(PI_i)}{P(TE)}$	$I_{PI_i} = \frac{P(PI_i)}{P(PI_{\max})}$
3907	1.0583E-08	0.3492	1.0000
3911	5.2915E-09	0.1746	0.5000
3915	2.6457E-09	0.0873	0.2500
4117	2.2048E-09	0.0728	0.2083
4123	1.1024E-09	0.0364	0.1042
4201	9.9215E-10	0.0328	0.0938
4129	5.5119E-10	0.0182	0.0520
4207	4.9607E-10	0.0164	0.0469
4269	4.4647E-10	0.0147	0.0422
3935	4.2332E-10	0.0140	0.0400
4535	3.8406E-10	0.0127	0.0363
4241	3.5717E-10	0.0118	0.0338
4119	2.9397E-10	0.0097	0.0278
4213	2.4803E-10	0.0082	0.0235
3905	2.3518E-10	0.0078	0.0222
4273	2.2323E-10	0.0074	0.0211
4199	2.2048E-10	0.0073	0.0208
3939	2.1166E-10	0.0070	0.0200
3919	2.1166E-10	0.0070	0.0200
4539	1.9203E-10	0.0064	0.0181
4955	1.8753E-10	0.0062	0.0177
4245	1.7859E-10	0.0059	0.0169
4956	1.6886E-10	0.0056	0.0160
4957	1.6700E-10	0.0055	0.0158
4125	1.4698E-10	0.0049	0.0138
4159	1.2248E-10	0.0040	0.0116
3909	1.1759E-10	0.0039	0.0111
4277	1.1162E-10	0.0037	0.0105
4205	1.1024E-10	0.0036	0.0104
3943	1.0583E-10	0.0035	0.0100
4115	9.7990E-11	0.0032	0.0093
4543	9.6014E-11	0.0032	0.0091

7.3.7 Measure: risk reduction worth

Table 38 gives an overview of the results of risk reduction worth (RRW) measure applied to top event # 1 and consistency rule #1. The table contains the RRW as well as Fussell-Vesely results as they are directly related to each other. The condition of the agitator motor appears in the third place. The result literally means that if the normal condition of the agitator motor can be eliminated the probability of the top event would be reduced 33 times. It is of course not possible to eliminate this condition, as it is the desired state of the agitator motor. The result of the agitator motor scoring such a high reduction worth should be interpreted as being an important variable that needs to be investigated further, as it is directly related to the speed of the agitator motor. The reduction worth measure gives the same interpretation result as the non-probabilistic importance measure prioritizing for the number of variables and states (see section 7.3.3). The digital output signal DO1 also scores high in reduction worth. This makes sense of course. If the nitrator feeder is never turned on the material will not be supplied to the nitrator tank.

Table 38. Reduction Worth

Variable	State	Description	RRW	FV
CNF	Normal	Operates normally	88.0569	0.9886
DO1	On	Speed on	44.5644	0.9776
CAM	Normal	Operates normally	33.1114	0.9698
DO7	Off	Speed off	32.8672	0.9696
BW	Normal	Normal	13.0283	0.9232
TT	Low-low	Low-low	11.5044	0.9131
PETN	2 %	2 %	5.52989	0.8192
DO6	Off	No speed	3.38314	0.7044
AO1	High-high	Speed high-high	2.96158	0.6623
PETN	1 %	1 %	2.92949	0.6586
WPE	1	1 unit	2.24416	0.5544
WPE	2	2 units	1.38351	0.2772
PETN	3 %	3 %	1.32664	0.2462
PETN	4 %	4 %	1.19496	0.1631
AO1	High	Speed high	1.18315	0.1548
WPE	3	3 units	1.16090	0.1386
AO1	Low	Speed low	1.09675	0.0882
PETN	5 %	5 %	1.08444	0.0778
AO1	Medium	Speed medium	1.07782	0.0722
TT	Low	Low	1.04363	0.0418
BW	Wide	Wide	1.03714	0.0358
BW	Small	Small	1.02424	0.0237
PETN	6 %	6 %	1.01774	0.0174
PETN	>6 %	>6 %	1.01758	0.0172
WPE	4	4 units	1.01121	0.0110
WPE	0	0 unit	1.00623	0.0062
:	:	:	:	:
:	:	:	:	:
:	:	:	:	:
CAM	Low	Stuck low speed	1.00002	2E-05
CAM	High	Stuck high speed	1.00001	8E-06
CNF	Low-low	Stuck low-low	1.00000	7E-07

7.3.8 Measure: risk achievement worth

Table 39 gives an overview of the results of the risk achievement worth (RAW) measure applied to top event #1 and consistency rule #1. The 6% concentration of PETN is on top of the list. This is an obvious result as the 6% represents the possibility for high local concentrations. The second position on the list is the temperature in the tank being high. This also is an obvious result as it is only one step to go from a high temperature to a high-high temperature. Based on these results it can be decided to prevent for example a 6% PETN concentration by monitoring the concentration and implement a safety interlock (see section 7.4.2).

Table 39. Achievement Worth

Variable	State	Description	AW
PETN	6%	6 %	33.496
TT	High	High	22.560
CNF	Low-low	Stuck low-low	13.131
CNF	Low	Stuck low	11.108
CNF	Medium	Stuck medium	6.6033
TT	Normal	Normal	3.4340
AO1	Low	Speed low	2.6165
DO1	On	Speed on	2.3400
CAM	Low	Stuck low speed	2.2503
CAM	Off	Stuck off	1.9344
AO1	Low-low	Speed low-low	1.7994
DO1	Off	Speed off	1.5983
WPE	0	0 unit	1.5849
AO1	Medium	Speed medium	1.5072
TT	Low	Low	1.4999
DO7	Off	Speed off	1.4594
CNF	High	Stuck high	1.4500
PETN	3%	3 %	1.3556
BW	Small	Small	1.3228
PETN	4%	4 %	1.2894
DO6	Off	No speed	1.1939
TT	Low-low	Low-low	1.0962
AO1	High-high	Speed high-high	1.0075
CNF	Normal	Operates normally	1.0059
BW	Normal	Normal	1.0046

7.4 Identification of safety functions

This section will demonstrate how the DFM model of the manufacturing process can be used to identify safety functions. Two kinds of safety functions are identified. First of all there are the safety instrumented functions. These are safety functions that are often inherent to the design process. For example, the PETN manufacturing plant consists of a drain to the drowning tank that is utilized in case of a too high temperature in the tank. The manufacturing process was already designed to accommodate this feature. Still a safety function is required that carries out this emergency dump.

7.4.1 Safety instrumented function

The following top event is defined to analyze what needs to be operating in order to be able to dump the material into the drowning tank if the temperature gets to high. The emergency dump can only take place if the required equipment operates correctly. In order to investigate what needs to be operating the following top event is specified

```
Top Event #3
At Time 0, TDT = High-High
```

This top event should only occur if the temperature in the nitrator tank is too high and the material is dumped into the drowning tank. The top event actually simulates that the temperature in the drowning tank (TDT) is too high. This temperature can only be to high if material is successfully dumped. To make sure that the drowning tank section itself does not interfere with the analysis the following constancy rule is applied:

```
Consistency rule #3
At all times MI = NO (No mixing) AND
At all times TT = High-High (High-High)
```

This top event should identify which equipment needs to be operating without failure in order to dump the material. The analysis results in 14 prime implicants of which 3 are useful. The remaining 11 prime implicants are not useful to specify the safety function as they contain failure states, which actually favor the execution of the safety function, for example “no hydraulic pressure”. The three remaining prime implicants reflect situation where everything operates normally. Thus there are basically three conditions in the plant that will successfully dump the material. In prime implicant #1 and #6 the dump signal is initiated by the intended BPCS. In Prime implicant #5 the dump is initiated by the intended safety system[9].

```
Prime Implicant #1
At time -2 ,   DO4 = Close (Close valve)              AND
At time -1 ,   CV3 = Normal (Operates normally)       AND
At time -1 ,   CDV = Normal (Operates normally)       AND
At time -1 ,   DO5 = Tank  (To tank)                  AND
At time -1 ,    CD = Normal (Operates normally)

Prime Implicant #5
At time -2 ,  LUO1 = Close (Close valve)              AND
At time -1 ,   CV2 = Normal (Operates normally)       AND
At time -1 ,   CDV = Normal (Operates normally)       AND
At time -1 ,   DO5 = Tank  (To tank)                  AND
At time -1 ,    CD = Normal (Operates normally)

Prime Implicant #6
At time -2 ,   DO3 = Close (Close valve)              AND
At time -1 ,   CV2 = Normal (Operates normally)       AND
At time -1 ,   CDV = Normal (Operates normally)       AND
At time -1 ,   DO5 = Tank  (To tank)                  AND
At time -1 ,    CD = Normal (Operates normally)
```

The interesting finding in these prime implicants is that the diverter needs to be pointing the drowning tank which is represented by CD (condition diverter) and digital output signal DO5. Both these conditions are present in all three prime implicants. The safety function consists not only of measuring the temperature in the tank and

[9] In the original design of the PETN manufacturing process the BPCS controlled both hydraulic shutdown valves (V2 and V3) and the safety system controlled one hydraulic shutdown valve (V2).

opening the drain valve but also the diverter needs to be in the right position. This is an important finding as the batch process actually consists of two nitrators next to each other. In reality it might happen that the second nitrator is dumping successful mixed material to the next step in the manufacturing process (the filter). If this is the case then the material in the first nitrator cannot be dumped unless the diverter is first switched to the drowning tank. Even though it might be obvious that the diverter needs to be to the drowning tank, the interesting part is that the resulting prime implicants contain this condition and that with help of the content importance measures these prime implicant are identified. The definition of the top event and the resulting prime implicants hold the necessary information to define the safety function in terms of input, logic solving and output elements.

7.4.2 Safety interlock functions

There are many conditions in a process that can exist without representing an obvious dangerous situation. These conditions exist because they are normal acceptable operation conditions but only if they appear in certain combinations they turn into in dangerous system conditions. These conditions can be handled by using safety interlock functions. Safety interlock functions can be found by prioritizing the prime implicants, using the content measure, for zero failure states.

For top event #1 this measure results in 252 prime implicants. This means that 252 process conditions exist that can only be prevented by controlling the process in the correct way with the basic process control system. Consider the following prime implicant that was found by prioritizing for zero failure states.

```
Prime Implicant #785
At time -3 ,   WPE = 6          (6 units)              AND
At time -3 ,   DO1 = On         (Speed on)             AND
At time -3 ,   AO1 = High-high  (Speed high-high)      AND
At time -3 ,   CNF = Normal     (Operates normally)    AND
At time -3 ,    BW = Wide       (Wide)                 AND
At time -3 ,   CAM = Normal     (Operates normally)    AND
At time -3 ,  PETN = 1 %        (1 %)                  AND
At time -3 ,    TT = Low-low    (Low-low)              AND
At time -2 ,   CAM = Normal     (Operates normally)    AND
At time -2 ,  PETN = 2 %        (2 %)                  AND
At time -1 ,   DO6 = Low        (Speed low)            AND
At time -1 ,   DO7 = Off        (Speed off)            AND
At time -1 ,   CAM = Normal     (Operates normally)
```

All the literals in this prime implicant represent “normal” conditions that can exist in the batch process. The situation in this prime implicant is that the available PE to be fed into the tank is at its highest value (WPE = 6). The condition of the nitrator feeder is normal (CNF = Normal) and the basic process control system has turned the feeder on at the highest speed (DO1 = ON, AO1 = High-High). The bandwidth of the feeder is “Wide” (BW = wide) which means that with the amount of PE available, the maximum amount of PE is fed into the tank within a very short time period. Within 1 time step the concentrating of PETN in the tank jumps from 2% to 6%. This in combination with having the agitator motor running on low speed (DO6 = Low & DO7

= Off) results in local high concentrations of PETN that cause the temperature to go too high.

The problem in this situation is that the usual width of this particular feeder band is “Normal”. If it is possible to assemble a “wide” band then the application logic in the BPCS should be able to handle understand the difference. This situation represents a safety interlock situation that needs to be incorporated into the application logic of the BPCS. If it is allowed to use a “wide” band for the feeder then the BPCS should be program in a way that does not allow the process to feed the material at a high speed into the tank under the conditions in prime implicant 785. The safety interlock function would prevent the feeder to go into the high feeding position.

This discovery represents the true advantage of DFM and the importance measures. DMF makes a process model and not a failure model. It takes into account all possible situation even the once not planned for in the application logic. With help of the importance measures it was possible to find this particular condition in the process. Now it is possible to either change the design of the feeder or to take the wide bandwidth into account in the application logic of the BPCS.

The following prime implicant represents another useful safety interlock function. The problem in this prime implicant is that the concentration of PETN is very high (6%) and that the mixer is switched of. This condition can be prevented if the mixer is switched on at all time if the concentration of PETN equals 6%. A more resolute solution could be to prevent at all times 6% PETN concentration, never letting the process arrive at this condition.

```
Prime Implicant #311
At time -2 ,   DO1 = Off    (Speed off)                AND
At time -2 ,   CNF = Normal (Operates normally)        AND
At time -2 ,  PETN = 6%     (6 %)                      AND
At time -1 ,   DO6 = Off    (Speed off)                AND
At time -1 ,   DO7 = Off    (Speed off)                AND
At time -1 ,   CAM = Normal (Operates normally)
```

The prime implicants contain all the information to perform the sensing and logic solving part of typical safety interlock functions. The actuation part represents the interlock. For example, in case of prime implicant #785 the system should prevent either the wide bandwidth or a high-high speed of the nitrator feeder.

7.4.3 Verification activity

An independent party can make its own DFM model of the process and define the same top events that have been specified by the designer to identify the safety instrumented and safety interlock functions. The resulting prime implicants can be used as a checklist by the independent party to verify whether the correct safety functions have been implemented. If the plant already exist and safety functions have already been implemented than the resulting prime implicants can be used as fault injection test cases. These fault injection test cases can be used to verify whether the actual functional behavior of the implemented safety functions is correct. In this way it is possible to document what the correct safety functions are, whether they have been implemented and whether they function properly.

7.5 Design safety system

At this point the process equipment has been designed and the safety instrumented and interlock functions have been identified. For the described manufacturing process a safety system had already been designed and it will be demonstrated how DFM and the importance measures can be used to analyze and verify the safety system design. A full description of the safety system is given in section 3.3.2 and 4.6. The safety system is presented in Figure 7 and the corresponding DFM model in Figure 12. The safety system was designed to incorporate the safety function as it was also identified in section 7.4.1. The DFM model of the safety system represents the hardware architecture as well as the application software. At first instance the safety system design is analyzed and verified independent of the process it is trying to protect.

The top event of interest would be to analyze what would prevent the logic solver from giving a signal to open the drain valve while the temperature in the tank was high. The top event and additional consistency rule to reflect this condition would be specified as follows:

```
Top Event #4
At Time 0, LUO1 = Closed (Valve closed)

Consistency rule #4
At all times LUI1 = High     (Temp High)
```

This top event results in 1190 prime implicants with a total of 15440 literals if the deductive analysis traces back 14 time steps. Out of these 1190 prime implicants there are 83 prime implicants with a single point of failure and 654 prime implicants where two failure need to occur. The remaining prime implicants consist of 0, 3 or 4 failures.

The following prime implicant is found in two ways. Either by applying the content measure prioritizing for single points of failure or by applying the measure that prioritizes based on the number of literals. In this case the number of literals is only 1. The prime implicants represents a stuck at high output channel. This is a dangerous fault as everything can be working correctly in a safety system but if the output is frozen nothing will happen in case of a process demand[10]. It is very important to incorporate diagnostics that verify the operation of the output channel (see section 7.6.1).

```
Prime Implicant #1105
At time -1 ,   COC = High  (Stuck high)
```

A very interesting failure that is found when filtering for single points of failure are two prime implicants that involve the BPCS. The problem found in these prime implicants is that the correct functioning of the safety function actually depends on the correct operation of the BPCS. The safety system apparently does not work

[10] It is the author's experience that this is a frequent mistake made in safety system design.

independent of the BPCS. The safety system can sense the position of the diverter but it cannot switch itself the diverter to the drowning tank. For this functionality it relies on the BPCS. A design recommendation should be made to make the safety function operate independent of the safety system (or to involve the BPCS as part of the safety analysis) as recommended in the safety protection layer philosophy (see section 3.2).

```
Prime Implicant #94
At time -10,   CIC = Normal (Operates normally)          AND
At time -9 ,  COP2 = Normal (Operates normally)          AND
At time -9 , CCOMI = Normal (Operates normally)          AND
At time -9 ,  COP1 = Normal (Operates normally)          AND
At time -8 ,  CBUS = Normal (Operates normally)          AND
At time -8 ,  CRAM = Normal (Operates normally)          AND
At time -7 ,  CROM = Normal (Operates normally)          AND
At time -7 ,  CCLK = Normal (Operates normally)          AND
At time -7 ,  CCNT = Normal (Operates normally)          AND
At time -7 , CBPCS = Not   (Not functioning)             AND
At time -5 ,  CCLK = Normal (Operates normally)          AND
At time -2 , CCOMO = Normal (Operates normally)          AND
At time -1 ,   COC = Normal (Operates normally)

Prime Implicant #289
At time -10,   CIC = Normal (Operates normally)          AND
At time -9 ,  COP2 = Normal (Operates normally)          AND
At time -9 , CCOMI = Normal (Operates normally)          AND
At time -9 ,  COP1 = Normal (Operates normally)          AND
At time -8 ,  CBUS = Normal (Operates normally)          AND
At time -8 ,  CRAM = Normal (Operates normally)          AND
At time -7 ,  CROM = Normal (Operates normally)          AND
At time -7 ,  CCLK = Normal (Operates normally)          AND
At time -7 ,  CCNT = Normal (Operates normally)          AND
At time -6 , CBPCS = Not   (Not functioning)             AND
At time -5 ,  CCLK = Normal (Operates normally)          AND
At time -2 , CCOMO = Normal (Operates normally)          AND
At time -1 ,   COC = Normal (Operates normally)
```

The prime implicants have also been filtered for possible human interaction. The following interesting prime implicants are found. They demonstrate the strength of DFM and the importance measures. Prime implicant 29 shows that when the operator puts in a higher value for the temperature limit (COP1 = higher) and all other components are working the output signal will be to keep the drain valve closed. This prime implicant can be found in using different importance measures. The prime implicant will be found if a filter is applied that looks for “no failures”, or “human” states.

```
Prime Implicant #29
At time -13,  COP1 = Higher (Higher value)            AND
At time -12,  CRAM = Normal (Operates normally)       AND
At time -10,   CIC = Normal (Operates normally)       AND
At time -9 ,  COP2 = Normal (Operates normally)       AND
At time -9 , CCOMI = Normal (Operates normally)       AND
At time -9 ,  COP1 = Normal (Operates normally)       AND
At time -9 ,  CROM = Normal (Operates normally)       AND
At time -9 ,  CCNT = Normal (Operates normally)       AND
At time -8 ,  CBUS = Normal (Operates normally)       AND
At time -8 ,  CRAM = Normal (Operates normally)       AND
At time -7 ,  CROM = Normal (Operates normally)       AND
At time -7 ,  CCLK = Normal (Operates normally)       AND
At time -7 ,  CCNT = Normal (Operates normally)       AND
At time -5 ,  CCLK = Normal (Operates normally)       AND
At time -2 , CCOMO = Normal (Operates normally)       AND
At time -1 ,   COC = Normal (Operates normally)
```

From a design point of view it needs to be clear that this value is safety critical. This means that additional procedures need to be in place to assure that only knowledgeable and appointed safety people can change these values. Modern programmable electronic safety system can designate software values as safety critical, which means that they are password protected and that their values can not be overwritten without using a specified procedure.

7.5.1 Testing application software

With DFM it is actually possible to test the design of intended application software. A model of the software can be created using DFM. Once the DFM model of the application software has been tested it can be incorporated into the hardware model (see section 7.5) and later into the complete process model (see section 7.6.1). The following top event has been specified.

```
Top Event #5
At Time 0, SDP = Tank      (Drowning Tank)
```

This top event tests the software routine that interprets whether the diverter position is towards the drowning tank or towards the next step in the batch process. This is an interesting exercise as it verifies what could cause the software variable SDP (software representation of the diverter position) to point at the tank. The top event resulted in 12 prime implicants by backtracking one time step. The first prime implicant is the only prime implicant that is truly wanted. The next three prime implicants clearly indicated a situation that should not lead to the tank position for the variable SDP.

```
Prime Implicant #1
At time -1 , SLUI3 = Tank  (Drowning tank)            AND
At time -1 , SLUI4 = Off   (Off)                      AND
At time -1 ,  SDP1 = Tank  (Drowning tank)            AND
At time -1 ,  SDP2 = Off   (Off)

Prime Implicant #2
At time -1 , SLUI3 = Tank  (Drowning tank)            AND
At time -1 , SLUI4 = Filter (Filter)                  AND
At time -1 ,  SDP1 = Tank  (Drowning tank)            AND
At time -1 ,  SDP2 = Filter (Filter)

Prime Implicant #3
At time -1 , SLUI3 = Off   (Off)                      AND
At time -1 , SLUI4 = Off   (Off)                      AND
At time -1 ,  SDP1 = Off   (Off)                      AND
At time -1 ,  SDP2 = Off   (Off)

Prime Implicant #4
At time -1 , SLUI3 = Off   (Off)                      AND
At time -1 , SLUI4 = Filter (Filter)                  AND
At time -1 ,  SDP1 = Off   (Off)                      AND
At time -1 ,  SDP2 = Filter (Filter)
```

This is a sign that the design of the software routine is wrong and needs to be examined further. The intended software routine was stated as follows:

```
IF ((SLUI3 = SDP1) AND (SLUI4 = SDP2))
THEN (SDP = TANK)
ELSE (SDP = FILTER)
```

The initial design of the software routine is too simplistic as it only compares the input variables SLUI3 and SLUI4 with the stored values in the memory. A fault condition could easily occur if the operator would accidentally switch the values in the memory (see prime implicant 4). This condition would also easily occur if a stuck at failure for the data would happen any were down the path. A more sophisticated software interpretation routine then presented above is required to be on the safe side.

7.6 Integrated analyses of existing software

This section will demonstrate how DFM can be used to find errors in existing software by specifying test cases. Suppose that the complete safety system, i.e., hardware and software exists, and that one would like to analyze whether the software works correctly. The software is modeled as a black box and DFM is used to model the environment of the software (see Figure 21). The safety function consists of several software routines and one of the software routines checks whether the temperature did not exceed its specified limit. On purpose the following error, as specified in Table 40, has been programmed into this particular software routine. The programmer used the “>” sign in stead of the “<” sign.

Table 40. Example software error

	Software Routine
Correct	If T < 35 C Then Close Drain Valve
Design error	If T > 35 C Then Close Drain Valve

Inductive analysis is used to trace forward initial conditions and see how these conditions affect an end system state of interest. As we are only interested in the software behavior all conditional values are set constant to their normal behavior. This means that throughout the analysis they cannot fail and thus it is simulated as if the hardware is operating normally at all times. The starting conditions, that is the test vector has been specified in Table 41.

Table 41. Test vector

Starting Condition	Variable State
For all times	LUI1 = High (Temp high)
For all times	LUI3 = Tank (Drowning tank)
For all times	LUI4 = Off (Off)
For all times	ODP = Tank (Drowning tank)
For all times	OTL = 35C (35 degrees C)
For all times	BPCS = DPTD (DP to DT)
For all times	CBPCS = Normal (Operates normally)
For all times	CBUS = Normal (Operates normally)
For all times	CCLK = Normal (Operates normally)
For all times	CCNT = Normal (Operates normally)
For all times	CCOMI = Normal (Operates normally)
For all times	CCOMO = Normal (Operates normally)
For all times	CIC = Normal (Operates normally)
For all times	CLUI1 = Ok (Temp ok)
For all times	CLUI3 = Tank (Drowning tank)
For all times	CLUI4 = Off (Off)
For all times	CLUO1 = Open (Open drain valve)
For all times	COC = Normal (Operates normally)
For all times	COP1 = Normal (Operates normally)
For all times	COP2 = Normal (Operates normally)
For all times	CRAM = Normal (Operates normally)
For all times	CROM = Normal (Operates normally)
At time 0	LUO1 = Close (Close drain valve)
At time 0	ILUI1 = Ok (Temp ok)
At time 0	ILUI3 = Tank (Drowning tank)
At time 0	ILUI4 = Off (Off)
At time 0	OLOU1 = Close (Close drain valve)
At time 0	RDP1 = Tank (Drowning tank)
At time 0	RDP2 = Off (Off)
At time 0	RTL = Limit (Limit value)

When this test vector is applied and the DFM model is traced forward the safety system eventually results in a state for LOU1 that puts the drain valve in a closed position. This should not have happened as the temperature is LUI1 = HIGH at all times. Since the hardware is operating normally it can only indicate a failure in the software. The actual “>” programming error is not yet found, but the fact that the

software did not perform as intended indicates is the significant discovery. One can now start more detailed analysis to localize where the failure resides.

7.6.1 Diagnostics for the integrated process

At this point in time the process is designed and verified for safety. The software has been designed, analyzed, and integrated into the hardware architecture. The safety functions have been identified and a safety system has been designed that is capable of carrying out the safety functions. The safety system design has been verified for functional safe behavior and is integrated with the process. It is now possible to carry out integrated analysis to verify if the safety system is capable of carrying out the safety function under all conditions. An additional safety layer of protection can be added that monitors whether it is possible to carry out the safety function.

A top event needs to be defined that covers the correct functioning of the safety functions. A representative top event would be top event #6.

```
Top event #6
At Time 0, PDV = Closed

Consistency rule #5
At all times TT = High-High
```

A consistency rule has been used setting the temperature in the tank to a too high temperature (over 35 °C). This makes sure that a demand to the safety system should have been initiated.

```
Prime Implicant #115
At time -1 ,   CDV = Closed (Failed Closed)
```

Safety shutdown only when one hydraulic valve closes but if the drain valve is stuck closed nothing will happen. You can only find that this is part of the diagnostic function because of the integrated model.

7.6.2 Verification activity

Independent party can make an integrated model of the process together with the safety system and the application logic. A top event can be specified that helps identify the required diagnostics. This list of prime implicants can serve as a checklist that the plant design needs to comply with.

7.7 Alarm management using the residual probability concept

The assumption is made that the probability of occurrence of each literal is available and that the literals of each prime implicant are independent events. In practice, this will be very hard to prove, but most applications of reliability techniques are based on simplified assumptions. The purpose is to demonstrate the usefulness of the real-time alarm management system. It is recognized that more sophisticated reliability

algorithms will be necessary in the future. If it can be demonstrated that the concept of the real-time alarm system works, the next step can be to improve the accuracy of the associated reliability data and calculations.

As it is assumed that there is a Boolean AND relationship between the literals, it is possible to calculate the probability of occurrence of each prime implicant by multiplying the probability of occurrences for each literal. The same counts for the residual probability, where the calculation depends on those literals that exist in the plant. The following is an example calculation of the residual probability (see Table 42). The "True" column represents whether the literal actually exists in the manufacturing plant or not. In this example only literal 1, 5, 7 and 8 exist in the plant, which results in a residual probability that can be calculated as follows:

$$P_R = P_{Literal\,2} \cdot P_{Literal\,3} \cdot P_{Literal\,4} \cdot P_{Literal\,6} \Rightarrow$$
$$P_R = 0.0904 \cdot 0.6666 \cdot 0.0003 \cdot 0.9387 = 1.7E-5$$

The situation in the plant can change rapidly; if also literal 2 and 4 become true then the residual probability increases quickly to

$$P_R = P_{Literal\,3} \cdot P_{Literal\,6} = 0.6666 \cdot 0.9387 = 0.63\,.$$

A dramatic change in probability that should truly cause an alarm even though nothing has happened yet in terms of QESH. Suppose that literal 8 represents equipment failure and that this equipment would be repaired then the residual probability would be reduced to 1.37E-2.

Table 42. Example 1 residual probability

Prime implicant 1	Probability of occurrence	Example 1	Example 2	Example 3
		True	True	True
Literal 1	0.0001	☑	☑	☑
Literal 2	0.0904	☐	☑	☑
Literal 3	0.6666	☐	☐	☐
Literal 4	0.0003	☐	☑	☑
Literal 5	0.0045	☑	☑	☑
Literal 6	0.9387	☐	☐	☐
Literal 7	0.2961	☑	☑	☑
Literal 8	0.0219	☑	☑	☐
Residual Probability		1.7E-5	0.63	1.37E-2

With this example, it is also possible to demonstrate that it is not feasible to judge prime implicants purely on the remaining number of literals that need to occur. If literal 1, 4 and 5 do not exist yet in the field then the residual probability equals 1.35E-10. On the other hand if literal 3, 6 and 7 need to occur then the residual

probability equals 0.19. Although, three literals are remaining in both cases the residual probability makes a huge difference putting the focus of the operator on the second situation.

Rules can be set on when an alarm needs to be triggered, i.e., when an unacceptable residual probability has arrived. Based on the alarm the operator can directly verify the applicable prime implicant and see what the condition is in the process. What has happened so far and what needs to happen in order to arrive at an undesired top event. Based on this alarm and the analysis of the prime implicant the operator can make informed decisions on the course of action when controlling the plant.

7.8 Conclusions

In this chapter the importance measures where used to verify the design of the manufacturing process including the programmable electronic safety system. The importance measures were useful to narrow down the number of prime implicants and prioritize the analysis and helped locate random failures, systematic failures and common cause failures. This prioritization method can be used for design analysis as well as verification activities. Once the design is optimized and verified by an independent party it is possible to define an alarm management system for this final design.

Chapter 8 Conclusions & Recommendations

8.1 General conclusions

The goal of this thesis was to develop a method to support process hazard analysis and safety management that is capable of modeling and analyzing all technical aspects that can affect the safe operation of a hazardous manufacturing process. A safety management prioritization method has been developed that enables analysts to understand a manufacturing process in terms of conditions or deviations that affect safety. It supports the design of a manufacturing process including the protection layers and to operate the process in a safe manner. This method also provides the means to integrate safety and quality analysis into one comprehensive analysis.

In Chapter 1 it was demonstrated that safety and quality could be integrated because they have common management elements that depend on a thorough understanding of the process. For both safety and quality the objectives are to identify problems (hazards, conditions), eliminate or control the problems using appropriate measures, and monitor the implementation of these measures to provide feedback and continuous improvement.

In Chapter 2 managing for safety and quality was summarized as understanding the process and the possible process deviations. It was demonstrated that it is recognized in industry that integration can lead to results in terms of profitability, but no method currently existed that can actually support such integration. The method presented in this work is based on a thorough understanding of the process. Only if the process is understood it is possible to identify and analyze the effect of possible deviations. In order to achieve quality and safety it is necessary to either eliminate or control the possible deviations. Once these deviations are known they can be prioritized.

In Chapter 4, DFM was introduced as this methodology is used as basis for the process hazard analysis. With DFM it is possible to capture all technical aspects of a system, including hardware, software, physical parameters, environmental parameters, or human interaction when desired. Deductive as well as inductive analysis can be carried out with DFM. These features allow the analyst to analyze root causes as well as effects of deviations.

Chapter 5 described importance measures that can be used in conjunction with the prime implicants that are derived from the DFM analysis. Probabilistic and non-probabilistic importance measures are introduced. The probabilistic importance criteria are the preferred measures as they can measure the importance of literals and prime implicants in an unbiased manner. As probability data is not always available, or difficult to obtain, non-probabilistic importance measures have been developed. Analysis based on non-probabilistic evaluation should always utilize a combination of non-probabilistic importance measures. When used in combination with each other they can stimulate discussions among knowledgeable people regarding system improvements. The importance measures give the analyst guidance on which prime implicants are most important to analyze and can therefore be used as a prioritization method.

Chapter 6 demonstrated how the importance measures could be used during the design of the process, the identification of safety functions, the design of the safety system, and the identification of diagnostic systems. To support safety management during the operation of the manufacturing process a real-time alarm management system was introduced. The purpose of the alarm system is to give the operator a tool that monitors deviations during the operation of a manufacturing process. The alarm system is based on real-time prioritization of the prime implicants. The prioritization is based on the residual probability concept. The residual probability calculates the probability of occurrence of the remaining literals, i.e., those literals that do not exist yet in the field. The prime implicants can be prioritized based on their residual probability. The alarm management system supports risk informed decision making during the operation of the process.

The process hazard analysis and process safety management method presented in this work has been demonstrated in Chapter 7 using an industrial manufacturing process as an example. The manufacturing process consisted of the actual equipment to carry out the process as well as a programmable electronic safety system to protect the process. The different importance measures were used to analyze the process and safety system in terms of safety. It was demonstrated this method was able to identify problems on any level of the system, i.e., component level, software level, and on the interaction level between different layers of protection. The method was successfully used to support:

- The design of the actual process plant;
- The identification of safety functions and safety interlock functions;
- The design of the safety system;
- The identification of the diagnostic functions; and
- The operation of the process.

8.2 Recommendations further research

This is the first time that a method has been presented that supports lifecycle process hazard analysis and process safety management. Improvements can be made to this method in every step of the lifecycle. Some of these improvements are specific to the work presented in this thesis while other improvements are more general in nature. The following gives an overview of recommendations for further research.

When it comes to the importance measures the focus of research should be on the probabilistic importance measures. As probability data is not always available or incomplete, a major improvement could be made if there was an importance measure that could handle uncertainty in probability data. A good first step has been made with the generalized and DIM measure. Another approach might be to use Monte Carlo simulation and statistical sensitivity analysis to determine the effects of uncertain data on the importance measures, see the work presented in [97]. Research also needs to be carried out how one can collect probability data for the literals of the prime implicants, as they do not represent just standard hardware failure states.

The strength of the non-probabilistic importance criteria lays in the fact that they should be used in combination with each other. Many different importance criteria

can be defined this way. Research needs to be carried out how they can be combined and when these combinations are useful to use.

Research can be carried out on how importance criteria can be implemented that support different modes of operation, for example, startup, normal operation, and shutdown of a process. The same variable or variable state might have different importance based on the mode of operation it is in. For example, consider the position of a valve open during startup and closed during normal operation. A stuck closed valve during startup, or a stuck open valve during normal operation, are both undesired conditions that need to be analyzed and prioritized.

Research can be carried out to help define design rules based on the importance criteria. For standard application industry would truly benefit from the use of design rules. Companies can use standards and norms to implement design rules based on requirements presented in these documents. The design rules can make use of these importance criteria. In other words better guidance on the use of importance criteria needs to become available.

The real-time alarm management system has only been presented in theory. A practical case needs to be introduced that helps further research the implementation of thus system. Among others needs to be researched how often the data presented to the operator needs to be updated as the system is real-time. It is not clear how quickly the priority of variables and variable states will change in a real-life application.

Software is getting more complex and companies try to integrate already existing software into their designs. In both situations it will be difficult to make a DFM model of the software. Research should be carried out that supports the automated use of DFM to analyze software. In terms of time and resources the analysis with DFM would benefit if the real software could be integrated in the model and support the analysis as presented in section 6.4.3.

DFM as a tool can easily be enhanced to use it as an automated Markov modeler. The model that is created represents a state transition diagram. The inductive analysis techniques can be used to determine that effect of component failure modes on system level. This information can be used to determine the states and the transitions between the states for a Markov model.

A DFM model is deterministic and stochastic aspects are not taken into account. It would be useful to research how DFM and stochastic aspects can be integrated.

References

1. Redmill F., Chudleigh M., Catmur J., System Safety: HAZOP and Software HAZOP, Wiley, 1999
2. Leveson N.G., Safeware, System Safety and Computers. Addison Wesley, September 1995
3. Franklin N., The accident at Chernobyl. The Chemical Engineer, p.17-22. November 1986.
4. CNN.com, Authorities seek to question pair in "Love Bug' attack, May 11, 2000, Web posted at: 4:38 p.m. HKT (0838 GMT)
5. Bogard W., The Bhopal Tragedy. Westview Press, Boulder, Colorado, 1989
6. Leplat J., Accidents and incidents production: Methods of analysis. New Technology and Human Error, John Wiley & Sons, New York, 1987
7. Turner B., Man-Made Disasters. Wykeham Publications Ltd., London, 1987
8. Newby H., Risk Analysis and Risk Perception: The Social Limits of Technological Change. Trans IChemE, vol. 75., part B., August 1997
9. Johnson W.G., MORT Safety Assurance System. Marcel Dekker, New York, 1980
10. Apostolakis G.E., Editorial, Reliability Engineering and System Safety, 53, p. 1, 1996.
11. Thucydides, History of the Peloponnesian War, Book II, Penguin Books, 1972
12. Paul O'Neil, CEO ALCOA, Harvard Business School Case Study 9-692-042
13. Russel D.L., Getting a Handle On Risk Management. Chemical Engineering, p. 114 – 120, December 1999.
14. CEPP, General Guidance for Risk Management Programs (40 CFR PART 68), EPA 550-B-98-003, July 1998
15. OSHA-1910.119, Compliance Guidelines and Recommendations for Process Safety Management, OSHA §1910.119 App C, http:/www.osha-slc.gov/OshStd_data/1910_0119_APP_C.html
16. Factory Mutual Global, http://www.fmglobal.com/risk_management/ index.html, Factory Mutual Insurance Company, FM Global WWW site, Rhode Island, 1999
17. Kaplan S., Garrick B.J., On the Quantitative Definition of Risk. Risk Analysis, vol. 1, no. 1, 1981
18. IEC 61508: Functional Safety of Electrical / Electronic / Programmable Electronic Safety-Related Systems, Part 1 - 7, International Electrotechnical Committee, 1999.
19. Faller R.I., Aspects of TUV Type Certification and Safety-Related Applications of Programmable Electronic Systems. Proceedings Symposium “Safety in the Process Industry, Yesterday, Today, Tomorrow”, s’ Hertogenbosch, 1996
20. Xerox Corporation, 1999 Environmental Health and Safety Progress Report, www.xerox.com, 1999

21. 3M Corporation, 3M Pollution Prevention Pays, Moving Toward Environmental Sustainability, www.3m.com/profile/envt/3p.html, 2000
22. Unical Corporation, Health, Environment & Safety Report 1995-96, http://www.unical.com/www.unocal.com/responsibility/95hesrpt/. 1996
23. Safety Service Center. www.safety-sc.nl, 1999
24. ISO/IEC Guide 51 second edition (1997 draft)
25. EEC/97/23, Directive Of The European Parliament And Of The Council On The Approximation Of The Laws Of The Member States Concerning Pressure Equipment, May 1997
26. EN 298, Automatic gas burner control systems for gas burners and gas burning appliances with or without fans. European Norm EN298
27. Integrated Safety Management System Description, Lawrence Livermore National Laboratory, Version 3.0, February 14, 2000
28. Henley E.J., Kumamoto H., Probabilistic Risk Assessment, Reliability Engineering, Design, and Analysis. IEEE PRESS, Piscataway, New Jersey, 1981
29. US MIL-STD-1620, Failure Mode and Effect Analysis. National Technical Information Service, Virginia
30. Lewis, E.E., Introduction to Reliability Engineering. John Wiley & Sons, New York, 1987
31. CCPS, Guidelines for Safe Automation of Chemical Processes. American Institute of Chemical Engineers, New York, 1992
32. Brombacher A.C., Reliability by Design, john Wiley & Sons, Chichester, 1992
33. Billinton R., Allan R.N., Reliability Evaluation of Engineering Systems, Concepts and Techniques. Pitman Books Limited, London, 1983
34. Rouvroye J.L., Stavrianidis P., Spiker R.Th.E., Nieuwenhuizen J.k., Brombacher A.C., Uncertainty in safety, New techniques for the assessment and optimisation of safety in process industry. Proceedings Winter Annual Meeting ASME, San Francisco, November 1995
35. Rouvroye J.L., Goble W.M., Brombacher A.C., Spiker R.Th.E., A comparison study of qualitative and quantitative analysis techniques for the assessment of safety in industry, PSAM 3, Greece, June 1996.
36. Rouvroye J.L., Brombacher A.C., New quantitative safety standards: different techniques, different results? ESREL Conference, 1998
37. TÜV Product Service, Quality Assurance Procedures. Automation, Software and Electronics – IQSE, Public - Version 2.0, Munich, January 1998
38. Garrett C, Apostolakis G.E., Context And Software Safety Assessment. 2nd Workshop on Human Error, Safety, and System Development, Seattle, WA, April, 1998
39. AFISC SSH 1-1, "Software System Safety," Headquarters Air Force Inspection and Safety Center, 5 September 1985
40. FDA, (DRAFT) Reviewer Guidance for Computer-Controlled Devices, Medical Device Industry Computer Software Committee, January 1989

41. FDA, Reviewer Guidance for Computer-Controlled Medical Devices Undergoing 510(k) Review. Office of Device Evaluation, Center for Devices and Radiological Health, Food and Drug Administration, 1991

42. IEEE, Software Safety Plans. IEEE-1228, Institute of Electrical and Electronics Engineers, Inc., 1994.

43. JPL, Software Systems Safety Handbook," Prepared by Jet Propulsion

Laboratory for National Aeronautics and Space Administration, May 10, 1993.

44. MIL-STD-882B, "System Safety Program Requirements," Department of Defense, 30 March 1984

45. Interim Defence Standard 00-55, "The Procurement of Safety Critical Software in Defence Equipment," Parts 1 and 2, Ministry of Defence, UK, 5 April 1991

46. ISO 8402:1994, Quality Management And Quality Assurance – Vocabulary, 1994

47. BS 5750, Quality Systems, Part O: Principal Concepts and Applications, Section O.2 Guide to quality Management and Quality System Elements. London, British Standards Institution, 1987

48. ISO 9000, Quality Management And Quality Assurance Standards - Guidelines For Selection And Use, March 1987

49. ISO 14000, Environmental Management Systems - General Guidelines On Principles, Systems And Supporting Techniques, 1994

50. ISO, Selection and Use of ISO 9000. International Standard Organization. 1998

51. Stimson W.A., Beyond ISO 9000: How to Sustain Quality in a Dynamic World. March 1998

52. ISO/DIS 9000, Quality Management Systems – Fundamentals and Vocabulary. International Standards Organization TC 176/SC 1, October 1999

53. Joan K. Loken, The Haccp Food Safety Manual. John Wiley & Sons, January 1995

54. IEEE 610, Computer Dictionary. Compilation Of IEE Standard Computer Glossaries, 1992

55. Brombacher A.C., Designing Reliable Products in a Cost and Time Driven market: a Conflict or Challenge. Intreerede, Eindhoven University of Technology, Eindhoven, 1999.

56. Unocal Corporation, 1999 Environmental Performance Report, www.unocal.com/responsibility/99report/envperf.htm, 1999

57. Drake E.M., Thurston C.W., A safety evaluation framework for process hazard management in chemical facilities with PES-based controls, Process Safety Progress, vol. 12, no. 2, April, 1993

58. Health and Safety Executive, Programmable electronic systems in safety related applications. Health and Safety Executive, ISBN 0 11 883906, Sheffield, United Kingdom, 1987

59. Kletz T., Cheaper, Safer Plants or Wealth and Safety at Work. The Institution of Chemical Engineers, Rugby, England, 1984

60. Guarro S.B., The Logic Flowgraph: A New Approach to Process Failure Modeling and Diagnosis for Disturbance Analysis Applications, PhD Thesis, University of California, Los Angeles, June, 1983

61. Guarro S.B., A Logic Flowgraph-Based Concept for Decision Support and Management of Nuclear Plant Operation. Elsevier, Reliability Engineering and System Safety 22, 1998

62. Guarro S.B., Milici T., Wu J.-S., Apostolakis G., Accident Management Advisor System (AMAS): A Decision Aid for Interpreting Instrument Information and Managing Accident Conditions in Nuclear Power Plants, OECD/CSNI Specialists meeting on instrumentation to manage sever accidents. Cologne, Germany, March 16-17, 1992

63. Guarro S.B., Okrent D., The Logic Flowgraph: A New Approach to Process Failure Modeling and Diagnosis for Disturbance Analysis Applications, Nuclear Technology, Vol. 67., December, 1984

64. Ting Y.T.D., Space Nuclear Reactor System Diagnosis: A Knowledge Based Approach, PhD thesis, UCLA, 1990

65. Guarro S.B., Wu J.S., Apostolakis G.E., Yau M., Embedded System Reliability and safety analysis in the UCLA ESSAE project. Procedures International Conference Probabilistic Safety Assessment and Management (PSAM), Beverly Hills, CA, 4-7 February, 1991

66. Garrett C.J., Guarro S.B., Apostolakis G.E., The Dynamic Flowgraph Methodology for Assessing the dependability of Embedded Software Systems, IEEE Transactions on Systems, Man and Cybernetics, vol. 25., no. 5, May, 1995

67. Yau M., Guarro S.B., Apostolakis G.E., Demonstration of the Dynamic Flowgraph Methodology using the Titan II Space Launch Vehicle Digital Flight Control System, Reliability Engineering and System Safety 49, 1995

68. Milici A., Yau M., Guarro S., Software Safety Analysis of the Space Shuttle Main Engine Control Software. PSAM 4, New York, September 1998

69. Yau M., Apostolakis G., Guarro S., The Use of Prime Implicants in Dependability Analysis of Software Controlled Systems, Reliability Engineering and System Safety, 62, 23-32, 1998

70. Houtermans M.J.M., Apostolakis G.E., Brombacher A.C., Karydas D.M., Programmable Electronic System Design & Verification Utilizing DFM. Safecomp 2000, Rotterdam

71. ASCA, DFM software tool, ASCA Incorporated, 706 Silver Spur Road, Ste 203, Rolling Hill Estates, CA, 1999

72. Cheok M.C., Parry G.W., Sherry R.R., Use of importance measures in risk-informed regulatory applications, Reliability Engineering and System Safety, 60 213-226, 1998

73. Vesely W.E., A time dependent methodology for fault tree evaluation, Nuclear Engineering Design, vol. 13, p. 337-360, 1970

74. Fussell B.J., How to Hand-Calculate System Reliability Characteristics. IEEE Transactions on Reliability, R-24, No. 3, 1973

75. PSA Applications Guide, EPRI TR-105396, August 1995

76. Barlow R.E., Proschan F., Importance of System Components and Fault Tree Analysis. Operations Research Center, University of California, Report ORC 74-3, 1974

77. Schmidt E.R., Jamali K.M., Parry G.W., Gibbon S.H., Importance measures for use in PRAs and risk management. Proceedings of the International Topical Meeting on Probabilistic Safety Methods and Applications, San Fransisco, CA, EPRI NP-3912-SR, 1985

78. Borgonovo E., Apostolakis G.E., A New Importance Measure for Risk-Informed Decision Making. Accepted for publication in Reliability Engineering and System Safety, March 2001

79. Garrett, C.J., Apostolakis, G., Context in the Risk Assessment of Digital Systems. Risk Analysis, 19, 23-32, 1999

80. Vesely W.E., The use of risk importances in risk-based applications and risk-based regulation. Proceedings of PSA '96, Park City, Utah, September 29 October 3, 1996.

81. TUV Product Service, Frequently Asked Question. IQSE website www.tuvglobal.com/iqse.htm, Danvers, 2001

82. Houtermans, M.J.M, Karydas, D.M., Brombacher, A.C., Overview of Programmable Electronic Systems and their Diagnostic Systems. 9th International Symposium on Loss Prevention and Safety Promotion in the Process Industries, Barcelona, Spain, May, 1998

83. Faller R., Houtermans M., Safety Requirements Specification. TUV Product Service, version 13, Munich / Danvers, July 1999

84. Brombacher A.C., Speakers Corner on “Software Development Awareness”. Philips POS, August 19, 1999

85. Honeywell, Abnormal situation management: giving your control system ability to 'cope'. Honeywell IAC - Journal Article. Internet: http://honeywell.datareturn.com:80/Pub/Journal/199507abnorm.html, 2000.

86. Yang X.P., Expert System Reasoning under Uncertainty with Applications to Incineration Systems. University of California Los Angeles, Ph.D. Thesis, 1990.

87. ANSI/ISA-S84.01-1996, Applications of Safety Instrumented Systems for the Process Industry. Instrument Society of America, Research Triangle Park, N. Carolina, USA, February 1996.

88. ANSI/ISA-S84.0.02-1996, Electrical / Electronic/ Programmable Electronic Systems (PES) - Safety Integrity Level Evaluation, Instrument Society of America, Research Triangle Park, May 1997.

89. Rouvroye J.L., Houtermans M.J.M., Brombacher A.C., Systematic Failures in Safety Systems: Some observations on the ISA-S84 standard. Proceedings ISA-TECH 97, Instrument Society of America, Chicago 1999.

90. Apostolakis, G., Moieni, P., "The Foundation of Models of Dependence in Probabilistic Safety Assessment", Reliability Engineering, 18 (3), 1987, pp. 177-95.

91. Mosleh, A., "Common Cause Failures: An Analysis Methodology and Examples," Reliability Engineering and System Safety, Special Issue, 34, 1991, pp. 249-91.

92. Mosleh, A., Siu, N., "A Multi-parameter, Event-based Common Cause Failure Model", Trans. 9th International Conference of Structural Mechanics in Reactor Technology, vol. M, Lausanne, Switzerland, August 1987.

93. Houtermans M.J.M., Brombacher A.C., Automated Safety and Uptime Analysis of Safety Instrumented Systems. Proceedings ISA-TECH 97, Instrument Society of America, Chicago 1999.

94. Brombacher A.C., MIR: Covering non-technical aspects of IEC 61508. Elsevier, Reliability Engineering and System Safety, Special Issue, Reliability Certification of Programmable Electronic Systems, vol. 66, no. 2., November 1999

95. Ippolito L.M., Wallace D.R., A study on hazard analysis in high integrity software standards and guidelines. NIST report NISTIR 5589, January 1995

96. Rouvroye, J.L., Robust Design Toolbox, Reference Manual, Version 2.3. Eindhoven University of Technology, 1998

97. Rouvroye, J.L., Enhanced Markov Analysis as a method to assess safety in the process industry. Ph.D. Thesis, Eindhoven University of Technology, May 2001

Appendix A Reliability Evaluation

A.1.1 Reliability RAMS

The real-time alarm management system (RAMS) can be part of the basic process control system (BPCS) or the safety instrumented system (SIS), or can be a custom build system that operates as an independent monitoring layer. The RAMS system will be implemented as a programmable electronic system (PES). A PES is a system based on one or more programmable electronic devices, connected to (and including) input devices (e.g., sensors) and/or output devices/final elements (e.g., actuators), for the purpose of control, protection or monitoring. The term PES includes all elements in the system, including power supplies, extending from sensors, or other input devices, via data highways or other communication paths, to the actuators, or other output devices (sensors/other input devices, and actuators/other output devices are therefore included in the term [82]. The purpose of this section is to demonstrate the approach to calculate the reliability of RAMS, which is based on the IEC 61508 standard. This standard addresses functional safety of PESs used in safety-related applications and sets performance standards for these safety-related systems in terms of Safety Integrity Levels (SIL). These SIL levels represent discrete levels of reliability depending on the severity of the process or the equipment under control (EUC). These SIL levels can also be used as a reliability metric to classify the measurements of the field information necessary to make the interpretation.

According to IEC 61508 there are two key attributes characterizing the performance of PESs, namely, its ability to implement its functions, and the required SIL at which these functions need to be carried out. The selection and specification of a PES, therefore, should be guided by the results of a detailed examination of the process safety functional specifications and an evaluation of the process functional integrity requirements. The process safety functional specifications dictate the safety functions that the PES must perform under stated circumstances. In this thesis the functional requirements for the RAMS system are derived by DFM in the format of prime implicants and the diagnostic knowledge base. The functional integrity requirements specify the level of confidence (reliability and availability) these safety functions must be performed in those circumstances. It is beyond the scope of this thesis to explain how SIL levels should be determined for the RAMS systems. For the reader who is interested, more guidance on this topic for safety-related systems can be found in part 5 of IEC 61508 [18].

The functional specifications and the SIL requirements of the process must be matched by appropriate PES design, quality of installation, and operational performance. The following sections address design considerations only, namely hardware architecture and reliability, and software quality and dependability.

A.1.2 Functional Specifications

The functional specifications of the RAMS system define the relevant functional parameters assigned to the PES. Such parameters include the boundaries of the function, interfaces and interactions with other systems, set-points and tolerances, the logical relationship between detection and actuation (in this case the presentation

of the actual results), the response time of the overall function and the time between periodic updates. The functional specification for the RAMS system is determined by the diagnostic database derived using DFM. Characteristics like the set points and tolerances, and time between periodic updates are determined by process safety characteristics and the process safety time respectively.

A.1.3 Integrity Specification

Quantitative definitions of Safety Integrity Levels (SIL) are given in ISA-SP84, and IEC 61508 [18, 87]. Each SIL corresponds to the range of probability of failure on demand. The IEC 61508 definitions are listed in Table 43. The design aspects of the PES include the overall hardware and software architecture (sensors, actuators, programmable electronics, embedded software, application software, etc.) that satisfies safety integrity requirements established by existing standards and the process hazard analysis and risk assessment. To achieve hardware and systematic safety integrity, the overall process for the selection of a safety-related PES of predetermined SIL should include the following elements:

- Architecture modeling;
- Hardware failure modes and failure rates;
- Systematic Failure modes;
- Reliability modeling;
- Reliability evaluation.

Table 43. Safety Integrity Levels [18]

Safety Integrity Level (SIL) [11]	Probability
4	$>10^{-4}$ - $\leq 10^{-5}$
3	$>10^{-3}$ - $\leq 10^{-4}$
2	$>10^{-2}$ - $\leq 10^{-3}$
1	$>10^{-1}$ - $\leq 10^{-2}$

A.1.4 Architecture Modeling

The architecture pertains to the configuration of hardware and software elements of the PES. Typical architectures are for example single, dual, triple-channel architecture, 1oo2 (one out of two), 1oo3, 2oo3 logic (see Figure 26) where X-out-of-Y represents voting and redundancy characteristics. The Y represents the redundancy of the function carried out by the PES. The X represents the voting aspect. For example, 2oo3 means that the function is carried out three times. The 2 means that at least 2 out of 3 channels need to be functioning correctly in order to be able to carry out the function. The result of each function is voted on among the channels.

Architecture modeling addresses the development of a detailed block diagram of the PES identifying each subsystem and the interconnections related to the safety

[11] SIL level 4 does not exist within the framework of ANSI/ISA-S84.01-1996 [87]

function under consideration. Depending on the availability of failure data and appropriate analytical tools the model may extend to a detailed representation of each hardware subsystem, identifying each component or group of components and the interconnections between them.

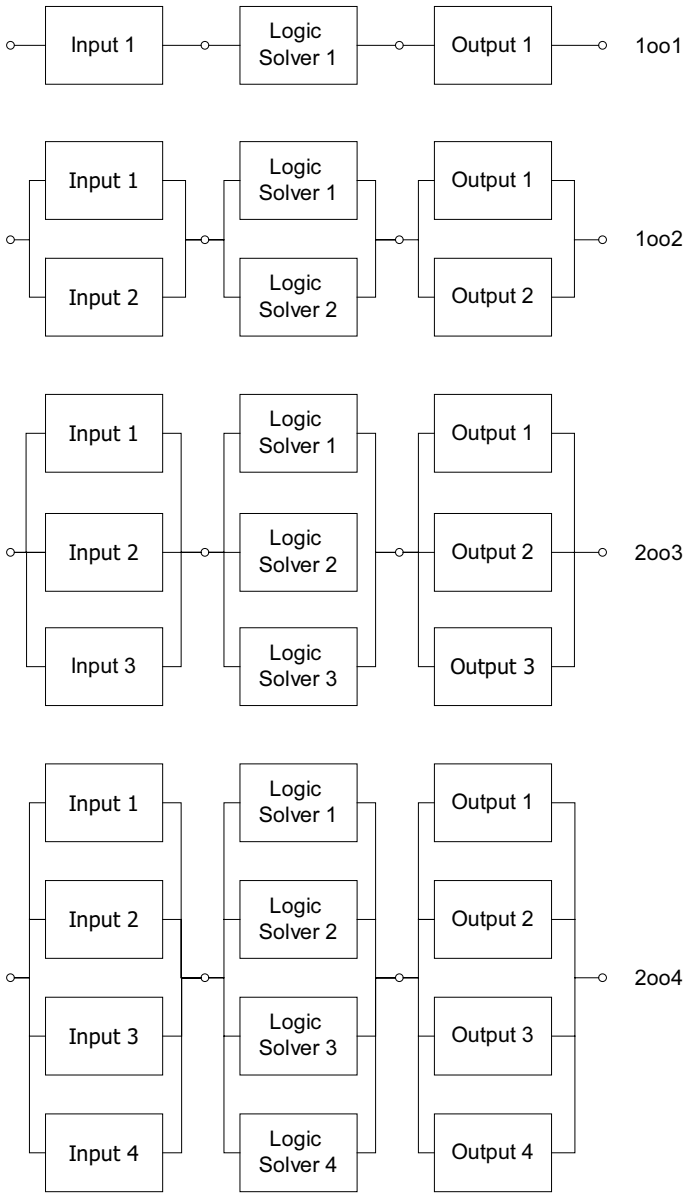

Figure 26. Example Architectures Programmable Electronic Systems

A.1.5 Hardware Failure Modes and Failure Rates

The objective at this stage is to identify the possible failures of the elements of the overall hardware architecture that carry out each separate function. The failures of the PES are addressed in terms of the connection and interaction of its components and subsystems. Failure Modes and Effects Analysis (FMEA) is an appropriate method to identify and evaluate the different failure modes and causes of the system. FMEA tabulates the actions to be taken to eliminate or reduce system failures, and documents the function under consideration. Random failures are identified and failure rates are tabulated for each component included in the modeled architecture.

A list of typical failure modes that need to be addressed for programmable electronic systems is presented in Table 44. The list of possible failure modes should also includes common cause failures, i.e., failures, which result from events causing simultaneous or coincident failures of two or more separate channels in a multiple channel system, leading to system failure.

Table 44. Overview of typical failure modes [18]

Component	Failure modes to consider
CPU	
▪ Register, internal RAM	▪ DC model for data and addresses; ▪ Dynamic cross-over for memory cells; ▪ No, wrong or multiple addressing
▪ Coding and execution including flag register	▪ No definite failure assumption
▪ Address calculation	▪ No definite failure assumption
▪ Program counter, stack pointer	▪ DC model
Bus	
▪ General	▪ Time out
▪ Memory management unit	▪ Wrong address decoding
▪ Direct memory access	▪ All faults which affect data in the memory; ▪ Wrong data or addresses; ▪ Wrong access time
▪ Bus-arbitration	▪ No or continuous or wrong arbitration
▪ Interrupt handling	▪ No or continuous interrupts; ▪ Cross-over of interrupts
Clock (Quartz)	▪ Sub- or super harmonic
Invariable memory	▪ All faults which affect data in the memory
Variable memory	▪ DC model for data and addresses dynamic cross-over for memory cells; ▪ No, wrong or multiple addressing
Discrete hardware	
▪ Digital I/O	▪ DC model; ▪ Drift and oscillation
▪ Analogue I/O	▪ DC model; ▪ Drift and oscillation
▪ Power supply	▪ DC model; ▪ Drift and oscillation
Communication and mass storage	▪ All faults which affect data in the memory; ▪ Wrong data or addresses; ▪ Wrong transmission time; ▪ Wrong transmission sequence
Electromechanical devices	▪ Does not energize or deenergize; ▪ Individual contacts welded, no positive guidance of contacts, ▪ No positive opening
Sensors	▪ DC model; ▪ Drift and oscillation

The system failures are discriminated into detected or undetected through diagnostic coverage. For each subsystem identified in the block diagram it is possible to divide the failure modes on sub system level as:

- Safe Detected
- Safe Undetected

- Dangerous Detected
- Dangerous Undetected

Hardware/software diagnostic functions are used to troubleshoot and identify hardware or software malfunctions of the PES on a continuous on-line (diagnostic test) or periodic off-line (proof test) basis. Diagnostic coverage characterizes the quality of the diagnostic programs. It is expressed quantitatively as the ratio of detectable faults to the total number of faults that may be hidden within the PES and render it inoperable when it is required to perform safety functions. The term diagnostic coverage is used to describe the fractional decrease in the probability of safe and dangerous hardware failures, resulting from the operation of the diagnostic tests.

The overall system failure is a function of the failures of its components. Table 3 provides a short list of random failures rates (i.e., failures occurring at random times and resulting from degradation in the hardware) of the typical PES components. These figures, presented in terms of ranges of a low, average and high value, can only be used for demonstration reasons and do not reflect failure rates supported by field data. They represent consensus rates used for demonstrative reliability quantification of widely used PES architectures included in the ISA Technical Report dTR84.0.02 [88]. For the reliability evaluation of actual systems, site-specific failure rate data are preferred, if available. If this is not possible, then generic data from credible published sources may be used.

Table 45. Hardware Failure Rates [87]

Item	Failures per million hours		
	Range		
Main Processor Board (memory, bus logic, communication)	12.00	25.00	50.00
I/O Processor and/or Common logic I/O module	2.50	5.00	10.00
Single Digital Input Circuit	0.10	0.20	0.40
Single Digital Output Circuit	0.10	0.20	0.40
Single Analog Input Circuit	0.05	0.10	0.20
Single Analog Output Circuit	0.25	0.50	1.00
Electromechanical Timer	1.50	2.50	5.00
Analog Trip Amplifier	0.20	0.40	0.80
Power supply	2.50	5.00	10.00
Sensor	2.00	13.00	42.00

A.1.6 Systematic and Common Cause Failure Modes

Systematic failures introduced in the PES safety requirements specification phase, or in the design and manufacture stages of the hardware, or in the design, implementation and testing phases of the software need to be analyzed

systematically to determine any contribution to the system unreliability. Non-probabilistic or quantitative assessment of selected design features that control and tolerate systematic failures in actual operation, and design procedures that prevent the introduction of systematic failures during the design process. Examples of systematic failures include human error introduced in the PES safety requirements specification phase, or in the design and manufacture stages of the hardware, or in the design, implementation and testing phases of the software. Systematic failures are best addressed through control during the development process. Systematic failures can arise during any step in the product lifecycle. Two different situations are possible that can trigger systematic failures over time [89]. There are

- Systematic failures that are always present in the system and that will be found provided a certain situation occurs. If this situation occurs during testing (after production or during proof test in the field) the failure will be found and can be repaired. If the situation does not occur during testing the failure will never be found and thus never be repaired.
- Intermittent systematic failures. Failures that only occur under certain conditions and disappear when the condition is not present anymore. Failures of this kind are very hard to find, because it is difficult to find and to reproduce the conditions that triggered the failure.

Common cause failures of the PES components should be considered and quantified. Common cause failures are the result of external events that cause multiple components in separate channels of a redundant system to fail, thus rendering the PES unable to perform its intended function. They are a subclass of systematic failures and usually refer to environmental conditions like temperature, humidity, electromagnetic fields, flooding, lighting, or vibration. There is considerable amount of research in the modeling and quantification of common cause failures [90,91,92]. The IEC 61508 standard provides a practical methodology for the evaluation of common cause failures using the b-factor model. The use and application of the b-factor model is not further explained in this document.

Software failures are also considered to be systematic failures. They can be introduced during any phase of the software lifecycle. Software analysis should not only focus the application code but also on the executive code (for example the operating system). The executive code is usually ignored during software analysis, as a user does not have access to the underlying software layers. So far the safety community uses third party evaluation to examine the correct functioning of executive code. It is common in the certification industry to use a combination of quality assurance, verification and validation, software criticality analysis and software testing to address executive software in PESs [19].

Techniques of software evaluation can be quantitative or qualitative. Mathematical models for software modeling have been tried only on very small applications. It is not certain that they can realistically address actual systems. On the other hand "assigning a probability to a software logic is basically meaningless, if design errors are found, they should be fixed rather than left in the code and assign a probability." [2]. It appears that the current state of the art in terms of software evaluation is limited to qualitative, albeit systematic methods from detailed structured checklists to Software Sneak Analysis and Software Fault Tree Analysis (SFTA). On application level the DFM has demonstrated itself as a very effective but qualitative

analysis tool for software [66,68,69]. As it is difficult to quantitatively express software reliability, any particular evaluation techniques are not included in this document. Nevertheless the effect of software reliability is demonstrated parametrically in the results in section 4.2.

A.1.7 Reliability Modeling

The PES needs to be translated into a reliability model that represents the interaction of its components and subsystems, as well as the transition from an operational state to a partially or totally failed state, of either safe or dangerous nature. The following system states exist that can be triggered by any of the sub system failure modes [93]:

- Operational State;
- Safe state;
- Dangerous state;
- Intermediate state.

These system states can be very well demonstrated with an emergency brake of a train (see Figure 27).

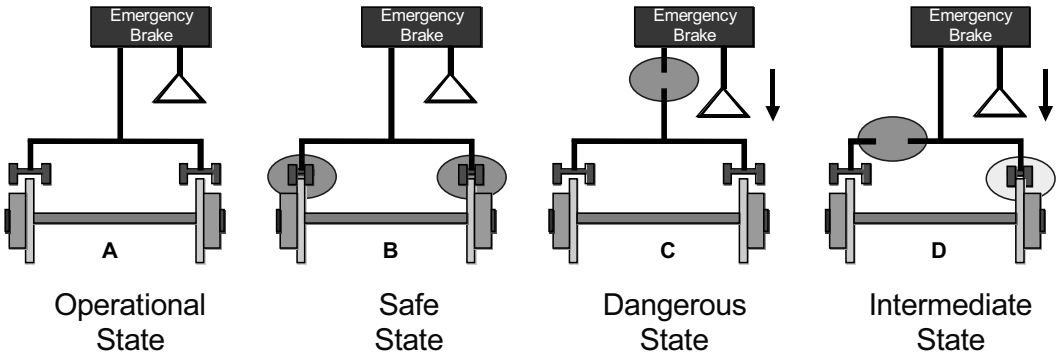

Figure 27. PES System Failure States

A PES used for safety can fail in two generic modes, "Fail Safe" and "Fail Dangerous". A "Fail Safe" mode causes the process to trip while no underlying deviation from safe process boundaries is present. It is a nuisance trip. A "Fail Dangerous" mode describes the condition of the PES not being able to respond to upsets of the process. In such failure mode of the PES, the process will continue its course and may enter a dangerous state. The failed PES is insensitive to this state; therefore this failure mode is dangerous.

Table 46 depicts the fraction of the total failure rates of PES components that is attributed to "Fail Safe" mode. These figures, also, are expressed in a range of low, average, and high values and are characterized by the same limitations of validity, as those in Table 45.

Table 46. Safe Failure Mode Ratios

Item	% Safe Failures		
	Range		
Main Processor Board	40	50	60
I/O Processor/ Common logic I/O module			
Single Digital Input Circuit	40	50	60
Single Digital Output Circuit	25	50	75
Single Analog Input Circuit	25	50	75
Single Analog Output Circuit	25	50	75
Relay (Industrial Type)	25	50	75
Power Supply	50	75	90
Sensor	80	95	99
	20	40	60

Reliability modeling methods include simulation techniques, reliability block diagrams, fault tree analysis, and Markov modeling. The conversion of the PES architecture into a mathematical model facilitates the quantification of the reliability of the PES and its SIL classification. The available methods for mathematical modeling have proliferated in the last 25 years and cover a wide range of sophistication from the simplest to the most complex techniques. In terms of increasing complexity and flexibility the list of methods includes Reliability Block Diagrams, Fault Trees, Dynamic Fault Trees, Markov Models, Hybrid Hierarchical Models, and Simulation. Modern software packages make it possible for the novice analyst to use the mathematical complex techniques as easy as the less complex techniques.

A.1.8 Reliability Evaluation

Reliability evaluation deals with the quantification of the selected PES reliability model. Introduction of the component and subsystem failure rates and probabilities in the model and numerical manipulation of the model generates characteristic reliability curves as a function of time, or inspection and testing interval. The comparison of these output reliability curves with the safety integrity requirements of the process leads to the acceptance or rejection of specific PES architecture and configuration. Figure 28 and Figure 29 demonstrate two reliability curves that result from the reliability evaluation. Figure 28 demonstrates the “classical” and the “initial probability” model. The difference lays in the way systematic failures are treated (see section 3.3) when creating the reliability model. The initial probability model takes into account systematic failures that exist in the PES from the beginning of operation. Hence, at time zero the initial probability curve starts with in initial probability of failure. Figure 29 also demonstrates the probability of failure on demand but this time taking into account the effects of periodic proof tests. Each time a proof test is carried out the probability of failure decreases depending on the coverage of the proof test.

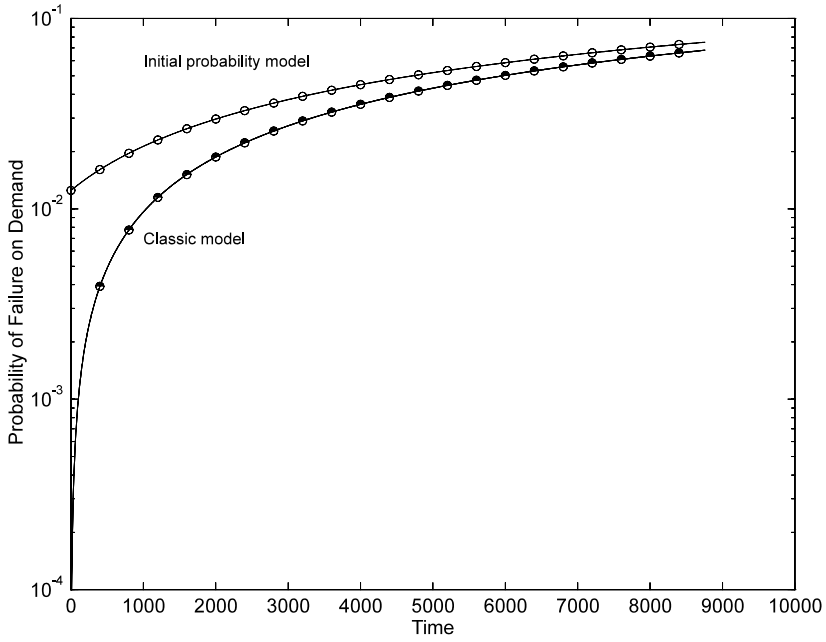

Figure 28. Probability of Failure on Demand

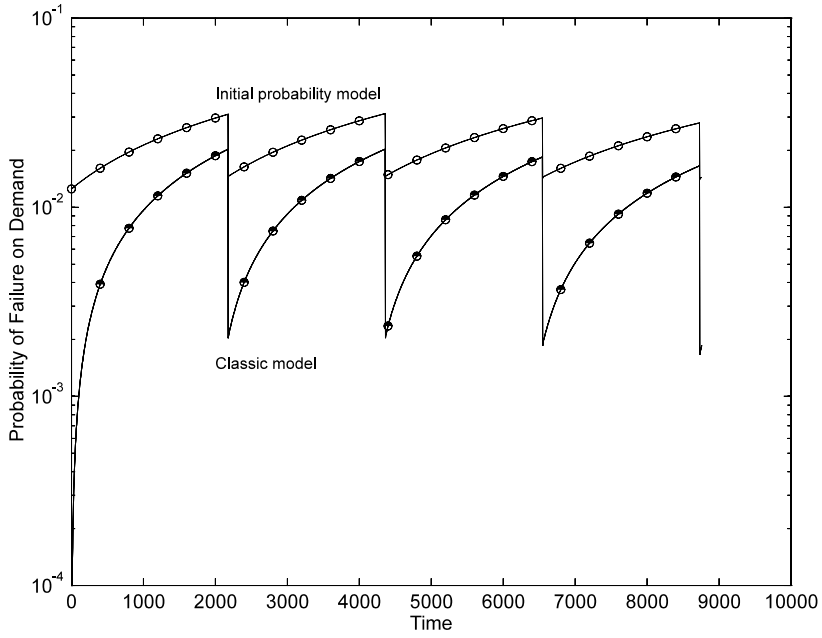

Figure 29. Probability of Failure on Demand with Periodic Proof Test Intervals

A.1.9 Influence of design parameters on the performance of PES

The purpose of this section is to demonstrate to the user the influence of design parameters on the performance of PES in terms of the probability of failure on demand and probability of fail-safe. The framework outlined in section 3 is used to

make the necessary models and calculations. The architectures in Figure 1 are used to demonstrate the different evaluations.

A.2 Evaluations

A Markov model has been created for all architectures. These Markov models have been quantified using the data presented in Table 47, a calculation engine built in Matlab and a Toolbox [96] developed at Eindhoven University of Technology. In total 5 different situations have been evaluated, each for a number of cases. The following sections represent the results of these analyses. In needs to be pointed out that these results only count for these architectures, using these assumptions. The performance of actual safety systems is a result of the interaction of all parameters presented here. This section presents only the effect when changing one parameter while the other parameters remain constant. It is not recommended to generalize the results. Safety needs to be addressed on a case-by-case basis.

Table 47. Reliability data [87]

Component	Failure rate	% Safe failures	% Diagnostic coverage
Sensor	13E-6	50	50
Logic	25E-6	50	90
Actuator	13E-6	50	50
Software	0.9E-6	50	-
Functional failures	0.6E-6	50	-

A.2.1 Evaluation – Influence of hardware, common cause, functional, and/or software failures

The purpose of this evaluation is to show the importance of including all aspects that determine the probability of failure. Three different cases have been carried out. Case 1 calculated only random hardware failures. Case 2 calculated random hardware failures and common cause failures. Case 3 calculated random hardware failures, common cause failures, functional failures, and software failures.

The results are presented Figure 30. Including (or excluding) these parameters can mean a factor of 10 or more difference within the same architecture. The IEC 61508 standard does not require for example to model functional and software failures. In terms of this standard this could mean one SIL level of difference or more. In practice the impact of these failures should be examined and addressed accordingly.

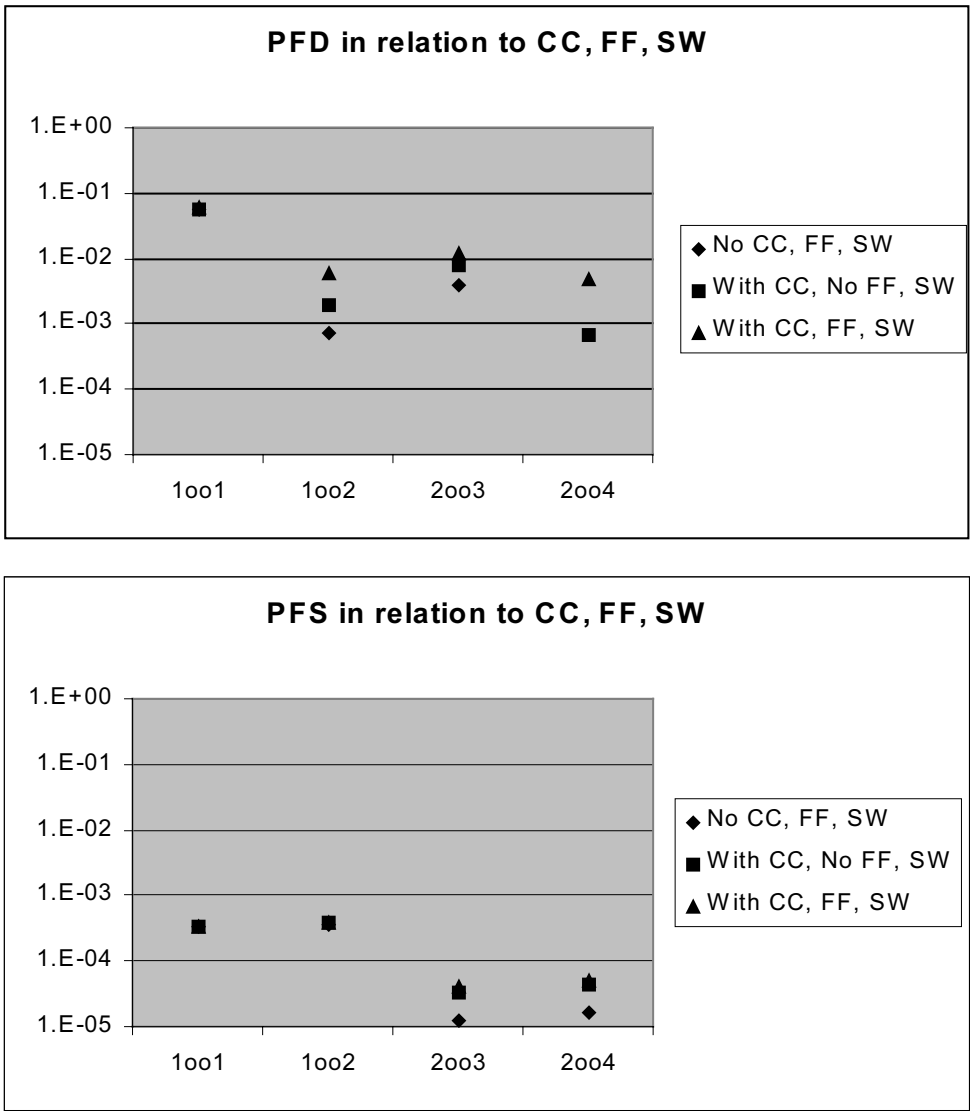

Figure 30. Influence of Common Cause, Functional, and Software failures

A.2.2 Evaluation – Influence of the safe ratio

The purpose of this evaluation is to show the influence of “Fail Safe” design. A lot of the equipment used for safety nowadays is fail-safe. Also the safety loop itself can be designed to follow the fail-safe principle. Seven different cases have been carried out where the % of safe failures for all components has been varied from 0%, 25%, 50%, 75%, 90%, 99%, up to 100%.

The results are presented in Figure 31. The calculated values can easily differ a factor of 10-1000 within the same architecture. Fail-safe designs improve safety (lower PFD) but increase also the number of spurious trips (which means shutdowns and startups). The PFS values for the 1oo2 and 2oo4 architecture are more susceptible to the fail-safe design.

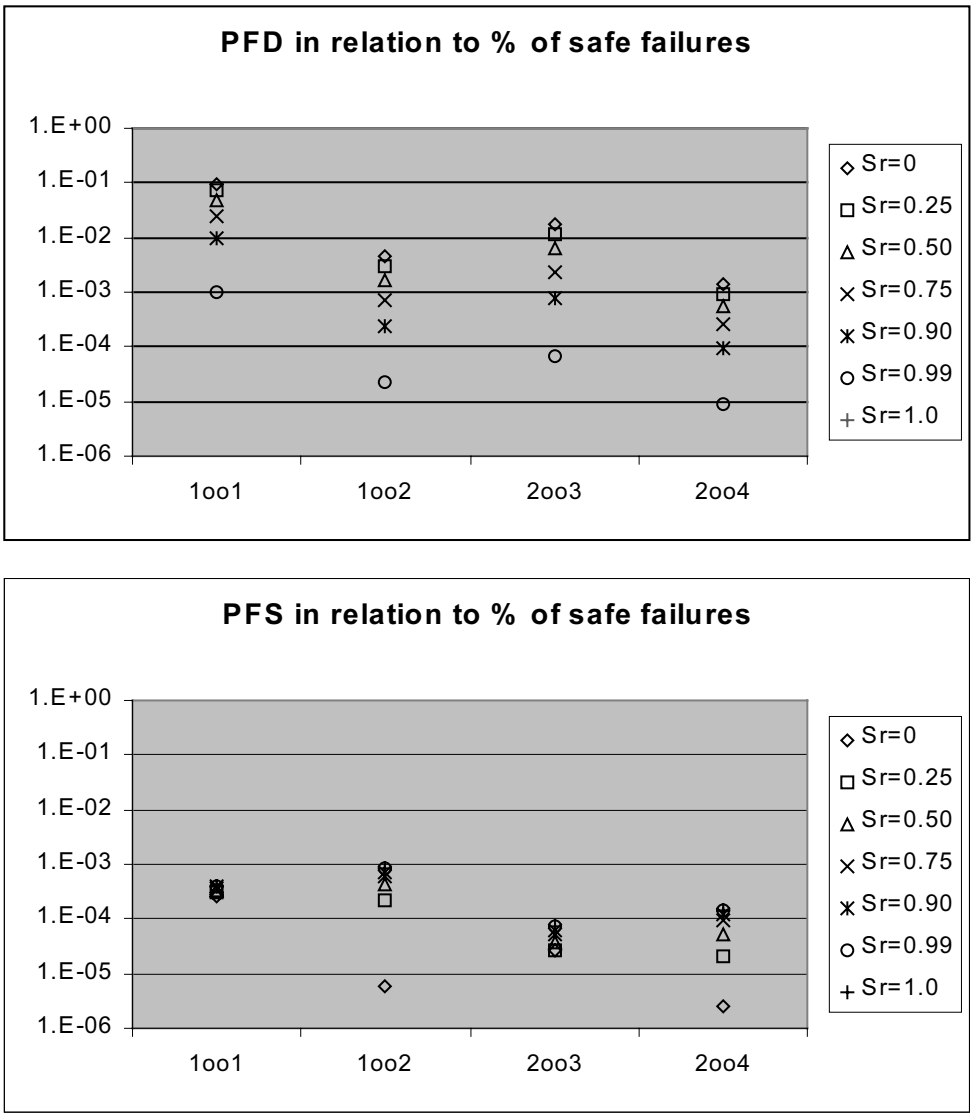

Figure 31. Influence of the Safe Failure Ratio

A.2.3 Evaluation – Influence of the common cause factor

The purpose of this evaluation is to show the influence of the common cause on the performance of the design. Functional or design failures and software failures are actually also common cause failures. But since we can classify these failures in a separate category they are modeled separately. All other causes (mainly environmental) that can lead to a common cause failure are captured by the common cause factor. Depending on the number of redundant components beta factors have been introduced. The factor b2 means that 2 components of the same kind fail simultaneously, b3 means 3 components and b4 means 4 components failing at the same time.

The results are presented in Figure 32. The more redundant the architecture the more susceptible the design is to common cause. Building in diversity in hardware and software can make these designs less susceptible to common cause. Common cause is a dominant factor; therefore the results do not differ as much throughout the

architecture for the PFD value compared to the other design parameters seen so far. The PFS only changes noticeable for the 2oo3 and 2oo4 architecture.

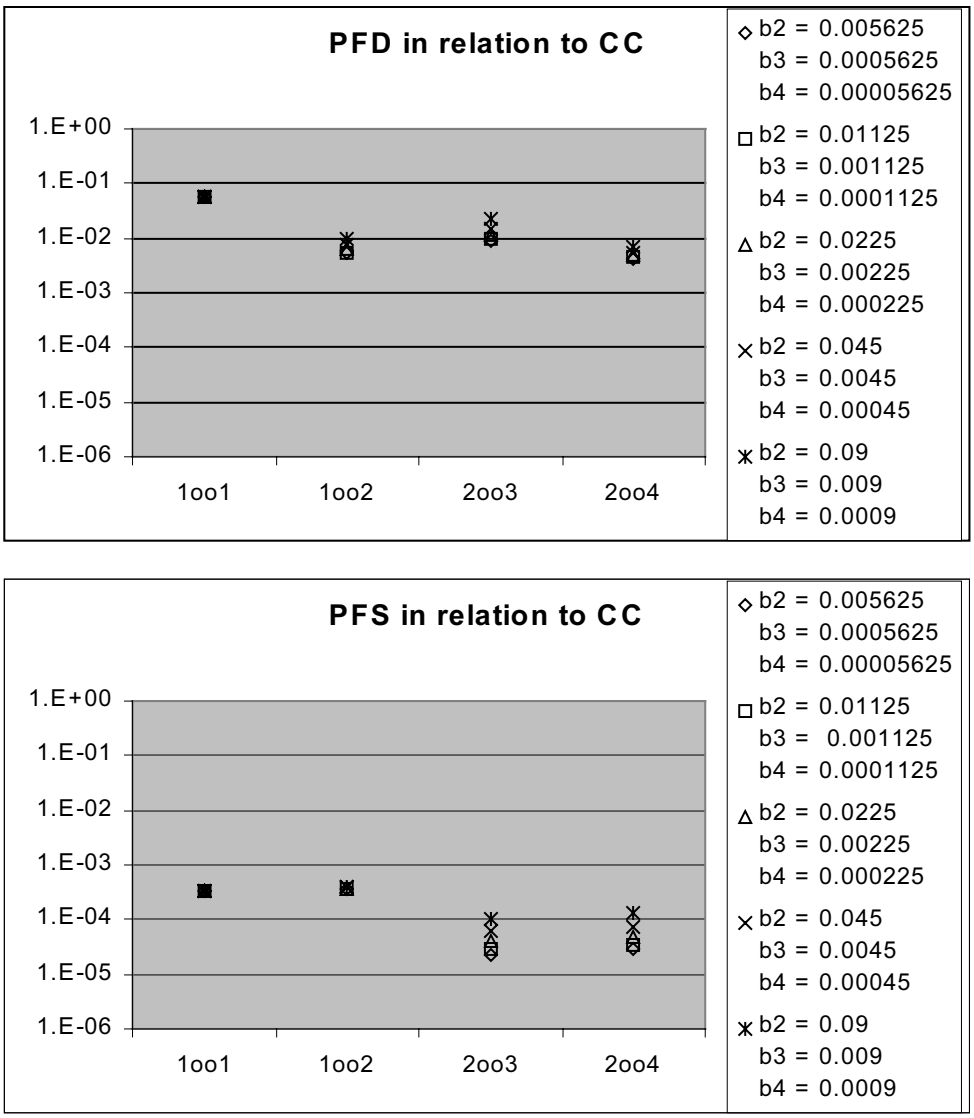

Figure 32. Influence of Common Cause

A.2.4 Evaluation – Influence of the online diagnostics

Programmable Electronic Systems play a major role when it comes to safety-related systems. One of the reasons is, if designed accordingly, because of the excellent online diagnostic capabilities [82]. The purpose of this evaluation was to show the influence of online diagnostics, represented by the diagnostic coverage factor, on the performance. The diagnostic coverage factor is varied from 0%, 25%, 50%, 75%, 90%, 99%, and 100%, for both dangerous and safe failures.

The results are presented in Figure 33. Good online diagnostics improve the performance significantly. Since the PFD is mainly determined by dangerous undetected failures, there is a big difference if the diagnostics can limit the dangerous undetected failures to a minimum. Not only the logic solver part but also the field devices.

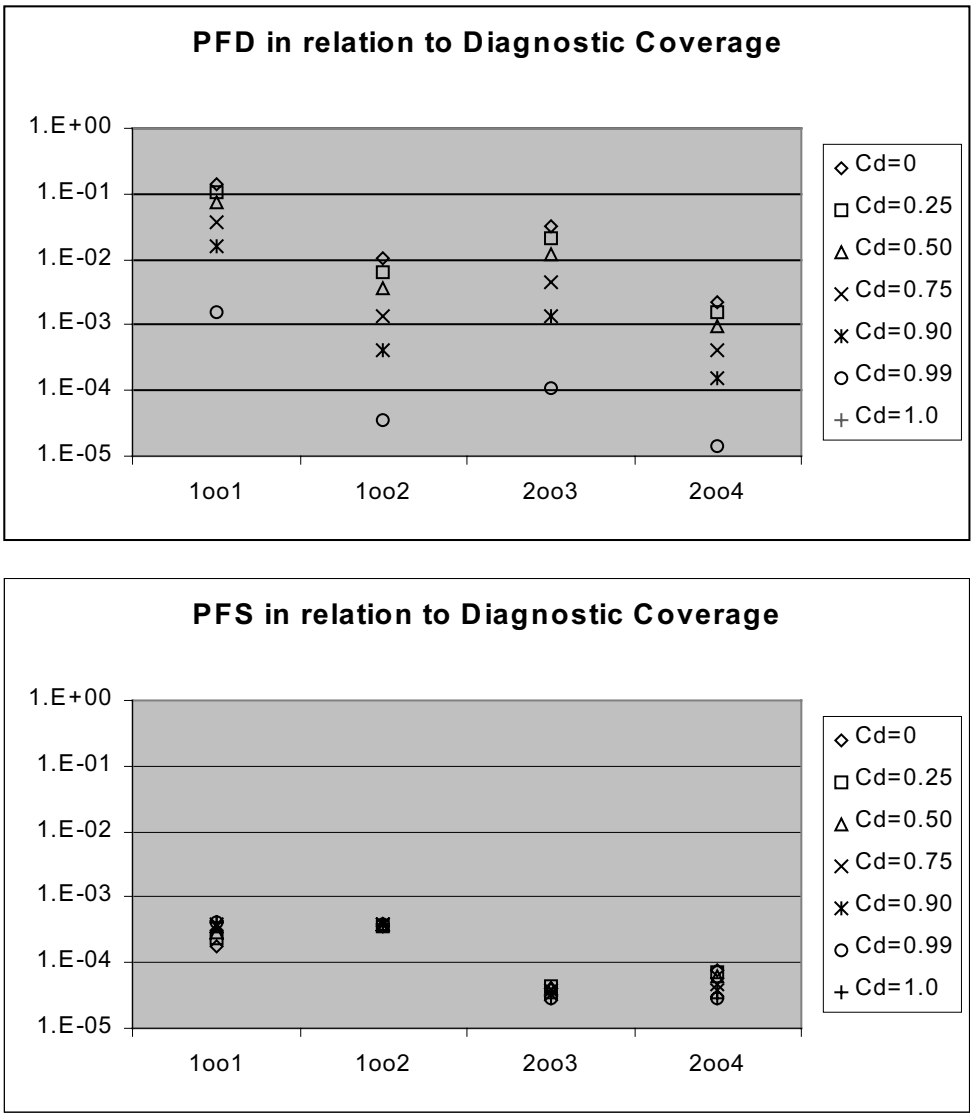

Figure 33. Influence of the online Diagnostic Coverage

A.2.5 Evaluation – The influence of Test Time Interval

A thorough test of the safety loop is often carried out during scheduled maintenance, e.g., every have 6 months or every year. Batch processes also give the opportunity to test the safety loop more frequently. Failures that cannot be found with online diagnostics might be found during these tests. The frequency of these scheduled tests or the Test Interval and the effect on the performance is evaluated by changing the test interval from 1 day, 1 week, 1 month, 3 months, 6 months, up to one year. The purpose of this evaluation was to show the effect of frequent testing on the probability of failure.

The results are presented in Figure 34. Frequent testing improves of course the performance. The proof test time interval has a major impact on the dangerous undetected failures. The PFD improves significantly. The influence on the PFS is less impressive, because detected or undetected safe failure lead to a trip, which in turn will be detected and repaired. The dangerous undetected failures are the true

sleeping failures that are only noticed when it is to late, i.e., after a demand has been placed on the safety system and it was not able to respond.

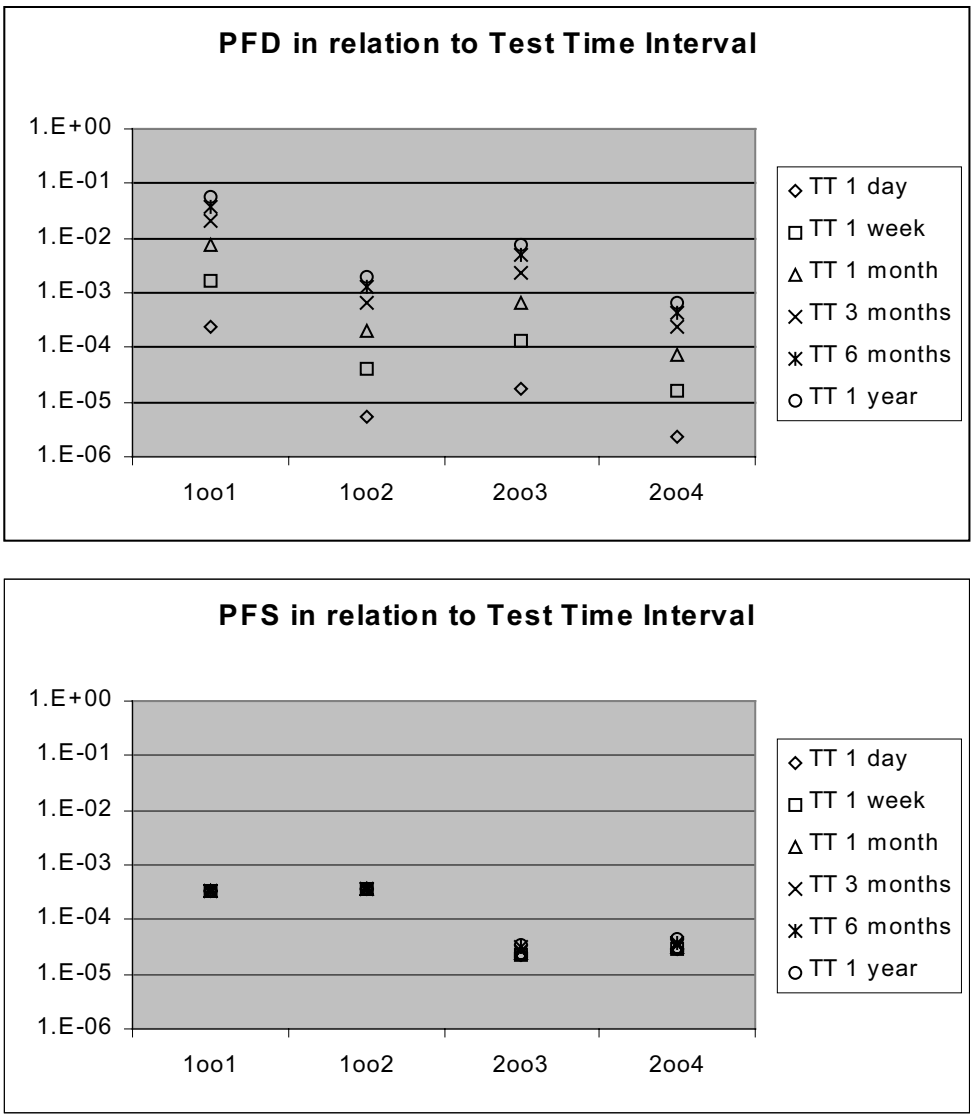

Figure 34. Influence of test time interval

Appendix B Description of the program

The original DFM program [71] utilized to carry out the analysis did not incorporate any ranking capabilities. In order to facilitate ranking, for the purpose of risk management, a program has been written that uses the output created by the DFM program to facilitate this ranking. The program is written in Microsoft Access using the database capabilities and the incorporated visual basic programming language. The reason why the program is written in a relational database environment is because it allows easy filtering and sorting of tables, i.e., filtering and sorting of information. This is a necessity as it will be necessary to rank from high to low importance hundreds or thousands of prime implicants.

The backbone of the database is based on six tables. A description of these tables is given in Table 48. The relationship that can exist between tables is characterized as 1-many (pronounce "one to many") or many-1. For example, the relationship between the table for the variables and the table for the states is characterized as 1-many. This means that a variable can have several states, but a state can only belong to one variable.

Table 48. Description tables

Name	Description	Relationship
Top Event	Stores information associated with top event	1-many: Top Event Item 1-many: Prime Implicant
Top Event Item	Stores the information associated with each state of the top event	Many-1: Top Event
Prime Implicant	Stores the information that identifies a prime implicant	Many-1: Top Event 1-many: Prime Implicant Item
Prime Implicant Item	Stores the information associated with a literal	Many-1: Prime Implicant
Variable	Stores the information associated with a variable	1-many: Variable State
Variable State	Stores the information associated with a variable state	Many-1: Variable

A visual basic program has been written to actually import, store, and analyze information in the previous mentioned tables. The deductive analysis of DFM generates as output a text file (see Figure 13). The first step in the program is to import the text file and store the top event and prime implicants in the corresponding tables, see the flowchart of the program in Figure 35. The output file generated by DFM contains the specified top event and the associated prime implicants with their literals.

A text file like presented in Figure 13 forms the basis for the prioritization program. The first part of the program analyzes the text file. The purpose of the analysis is to determine the top event and the associated prime implicants and store this information in the appropriate tables. The program can read the output file and can distinguish the difference between a top event, a prime implicant, and a literal.

The information is stored in the top event, top event item, prime implicant, and prime implicant item tables.

Once the top event and the prime implicants are imported into the database, it needs to be determined which variables and variable states exist. The software routine that has been written analyzes the literals of the prime implicants and ranks the variables and its associated states. This information is then stored in the variables and variable states tables.

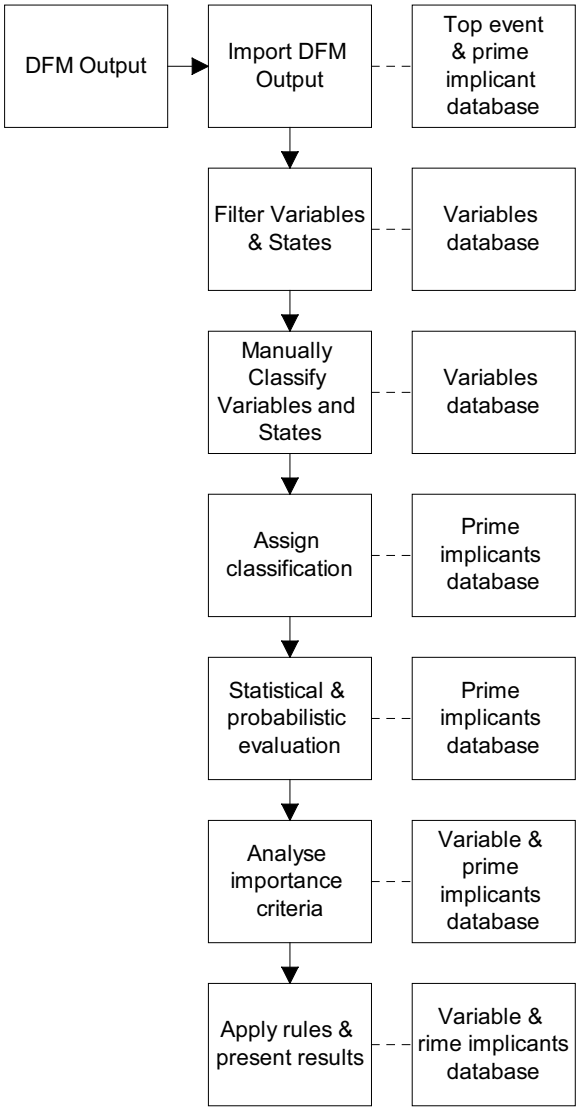

Figure 35. Program Flowchart

The main reason to filter out the variables and their states into separate tables is to be able to carry out the next step in the program, i.e., manual classification of the variables and their states. In this step, a window is opened that allows the user to manually add non-probabilistic and probabilistic characteristics to the variables and variable states. Because the ranking capabilities did not exist in the current version of DFM, it is necessary to add this information after the DFM model has been created. In future versions of the DFM program, it would be more effective to add this information during the system-modeling step.

As the classification is available on variable and state level it now needs to be translated to the literals of the prime implicants. This is the level where the classification needs to be available and analyzed. The step is carried out by a software routine that translates the information from the variables and variable states table to the prime implicants table.

Once the information is available on literal level, it is possible to start statistical and probabilistic evaluation. A software routine has been written that can calculate specific non-probabilistic and probabilistic importance measures. This information is stored in the top event, prime implicant, variable, and state tables.

Depending on the kind of information needed by the analyst it is possible to apply filter and sorting techniques to different tables. For example, it is possible to filter for all prime implicants that have less then four literals or between three and seven. Sorting techniques can be used to present information in ascending or descending order. For example, it is possible to sort the prime implicants on the number of literals, starting from the highest number down to the lowest number. It is also possible to apply several filter or sorting techniques to present the information in the most interesting manner. The kind of filters and sorting techniques to be applied depends on the analysis, the system, and the interest of the analyst and is determined by the rules of categorization that the analyst wants to apply. The advantage of using a database program is that the software routines to carry out filtering are already incorporated and only the customization of each filter needs to be programmed.

Curriculum Vitae

Michel Houtermans was born in Schinveld, The Netherlands, on August 15, 1970. In August 1995 he received his Masters degree in mechanical engineering from Eindhoven University of Technology, Eindhoven, The Netherlands. His Masters project focused on the automated generation of Markov models for programmable electronic safety-related systems.

In September 1995 he started his doctoral work with the section Reliability of Mechanical Equipment at Eindhoven University of Technology. In August 1996 he joined for two years Factory Mutual Research, Norwood, Massachusetts. He supported the department of Risk Engineering Methodologies and the department of Reliability Certification with research and industrial projects.

In July 1998 he joined the Institute for Quality and Safety in Electronics (IQSE) at TÜV Product Service in Munich. In December of the same year he moved to Danvers, Massachusetts, to help startup the IQSE department in the US. Currently he is the manager of the Automation, Software and Electronics – IQSE department responsible for the certification of safety-related programmable electronic systems. His field of expertise includes functional safety, system safety, and reliability engineering.

PROPOSITIONS

With the Ph.D. Dissertation of Michel J.M. Houtermans
A Method For Dynamic Process Hazard Analysis and Integrated Process Safety Management

1. Those who know how to prioritize can build safer systems.
   Chapter 5 of this thesis.

2. The role of diagnostics in safety systems used for the protection of plants should be more prominent as they can be seen as an additional layer in the safety protection layer philosophy.
   Chapter 3 and 6 of this thesis.

3. One of the best ways to achieve a profitable plant is to understand a plant in terms of safety.
   Chapter 2 of this thesis.

4. The evolution of the amount of requirements for programmable electronic systems between two standards is reversed proportional with the amount of requirements that people think they need to meet.
   DIN V VDE 0801 “Principles for computers in safety-related systems” vs. IEC 61508 “Functional Safety of Programmable Electronic Safety-Related Systems”

5. No matter how good or automated safety analysis tools will be in the future, achieving safety is and will primarily be an organizational issue.
   Houtermans M.J.M., et al, IEC 61508 and Management of Functional Safety, ISA EXPO, 2000

6. When a system is modeled and analyzed for safety, most of the time is spent on the analysis phase. When it comes to finding failures though, safety actually follows the 80-20 rule, i.e., of all the failures found in the system 80% are already found during the creation of the model, while the remaining 20% are found during further analysis using this model.

7. The safety philosophy in Europe is based on prevention, while in the U.S. it is based on control. This is why in the U.S. the real safety specialists are actually the lawyers.

8. Like in the stock market world also in the safety world there are two kinds of people. There are people that understand the fundamentals and there are those that follow the herd.

9. Statistics show that one out of three Americans will be injured or killed at some time in a motor vehicle crash. The reason for this is that a driver license exam in the US focuses too much on the facts of driving a motor vehicle instead of the rules of driving a motor vehicle.
   Massachusetts Registry of Motor Vehicles Driver’s Manual.

10. The speed and ease of email makes it the most misused communication tool ever.