UPPSALA
UNIVERSITET

UPTEC F11 063
Examensarbete 30 hp
December 2011

Probabilistic Safety Assessment using Quantitative Analysis Techniques

Application in the Heavy Automotive Industry

Peter Björkman

UPPSALA
UNIVERSITET

Teknisk- naturvetenskaplig fakultet
UTH-enheten

Besöksadress:
Ångströmlaboratoriet
Lägerhyddsvägen 1
Hus 4, Plan 0

Postadress:
Box 536
751 21 Uppsala

Telefon:
018 – 471 30 03

Telefax:
018 – 471 30 00

Hemsida:
http://www.teknat.uu.se/student

Abstract

Probabilistic Safety Assessment using Quantitative Analysis Techniques

Peter Björkman

Safety is considered as one of the most important areas in future research and development within the automotive industry. New functionality, such as driver support and active/passive safety systems are examples where development mainly focuses on safety. At the same time, the trend is towards more complex systems, increased software dependence and an increasing amount of sensors and actuators, resulting in a higher risk associated with software and hardware failures. In the area of functional safety, standards such as ISO 26262 assess safety mainly focusing on qualitative assessment techniques, whereas usage of quantitative techniques is a growing area in academic research. This thesis considers the field functional safety, with the emphasis on how hardware and software failure probabilities can be used to quantitatively assess safety of a system/function. More specifically, this thesis presents a method for quantitative safety assessment using Bayesian networks for probabilistic modeling. Since the safety standard ISO 26262 is becoming common in the automotive industry, the developed method is adjusted to use information gathered when implementing this standard. Continuing the discussion about safety, a method for modeling faults and failures using Markov models is presented. These models connect to the previous developed Bayesian network and complete the quantitative safety assessment. Furthermore, the potential for implementing the discussed models in the Modelica language is investigated, aiming to find out if models such as these could be useful in practice to simplify design work, in order to meet future safety goals.

Handledare: Mattias Nyberg
Ämnesgranskare: Bengt Carlsson
Examinator: Tomas Nyberg
ISSN: 1401-5757, UPTEC F11 063

Contents

Chapter 1

Introduction

Thanks to the industrial revolution many risk factors mainly associated with a few very simple technological systems, have been marginalized. However, the twentieth and twenty-first centuries have brought us a rapid increase in complex technological systems, along with much more frequent exposure to these systems. This development has lead to the importance to consider accidents caused by technological systems [1]. Furthermore, increasing dependability of technological systems, where computers play a crucial role, having the potential to cause catastrophic accidents, has resulted in a growing safety interest in the business world and academia [2].

1.1 Background to Safety Assessment

The automotive industry, in comparison with many other industries, implements a very large amount of systems or functions in most of their products. These systems incorporate a wide range of hardware and software components and the development process uses the principle of the v-model, which is a generally accepted approach in the automotive industry and used in standards such as ISO 26262. This v-model consists of one construction part and one test part. These parts interconnect with each other through modeling, and recursions of those models. The construction part is, in its simplest form, usually divided into three steps; requirements-analysis, system-draft and component-development. Correspondingly, the test part of the v-model consists of component test, system test and conformance test [3].

Systems engineering was defined by the International Council on Systems Engineering as ”an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation” [4].

During this development process in systems engineering, many factors mainly concerning functionality of a system and how a system incorporates customer demands are considered. However, during this process the notion of risk, or safety, has to be added to the equation, meaning that a developed system has to avoid unnecessary risk (chance of loss). This introduces requirements of system safety, often defining acceptable levels of risk, along with *Safety*

Assessment. This assessment determines quantitative or qualitative values of risk associated with combinations of specific situations and threats [5].

1.1.1 Functional Safety

Following the discussion about the development process, safety is considered as one of the most important areas in future research and development within the automotive industry. New functionality, such as driver support and active/passive safety systems, are examples where development mainly focuses on safety. At the same time, the trend is towards more complex systems, increased software dependence and more sensors and actuators, resulting in a larger risk for software and hardware failures. It is of most importance that the automotive industry looks seriously upon this risk and adjusts both products and operational methods to reduce it. Because of this, there exists a great need for processes, which clearly results in safe systems, and also provide proof that accurate safety measures has been satisfied [6].

The area *functional safety* is about safety in systems incorporating electronics and software. Operational methods described in e.g. standards are about focusing on safety during the entire life cycle of a product, classify systems in different safety integrity levels, base safety requirements of a system on its assigned safety classification and adjust work processes, such as testing and examination, towards a given safety classification [6].

1.1.2 BeSafe

Since functional safety is about making systems stay safe during operation, even in the event of faults, there are no standardized ways of assessing these safety issues. In the automotive industry, existing techniques such as those used in standards, i.e. ISO 26262, handle the assessment in a more or less ad hoc way. By identifying benchmark targets concerning models, software and hardware, along with defining measures and methodology for these benchmarks, a project called BeSafe hopes to improve the safety within the automotive industry, hence providing safer vehicles through measuring functional safety [7].

Furthermore, the BeSafe-project runs during a three year period starting in 2011 as a cooperation between leading automotive companies in Sweden. The resulting benchmarks are to be used to compare systems and to analyse how changes in a system affect safety. Also, when specifying requirements to suppliers, or estimating safety properties of systems based upon many safety critical components, the outcome of the project will be useful [7].

1.2 About Scania CV

This thesis work is carried out at Scania CV in Södertälje, Sweden. Scania was founded in 1891 and is today considered as one of the leading manufacturers of heavy trucks and buses in the world, with operations in about 100 countries. In total Scania employs about 35 000 people divided among sales, production and research. Of Scania's 35 000 employees, the research and development employs more than 3 000 and is concentrated to Södertälje [6].

Scania's aim is to "provide the best total operating economy for our customers, and thereby be the leading company in our industry" [6, Scania in brief]. One important factor in achieving this goal is to use a modular product system. This means that by using a limited number of main components, the development and product management costs are kept low, without compromising with customization possibilities. Furthermore, the company claims to be "an industry leader in sustainable effort" [6, Scania in brief]. Hence, sustainable development, concerning the company, customers and society, is an important aspect for Scania [6].

1.3 Objectives and Problem Formulation

Today most safety assessments are made in a qualitative manner, as described in standards such as ISO 26262. However, this standard also demand quantitative analysis of safety. This implies a need for more quantitative processes and methods to create safe systems, together with tools to assess, or prove, that the safety requirements have been fulfilled. A focus on developing safer systems ought to lead to better products, implying potential revenue. As a part of the BeSafe-project, this thesis describes over its aim to develop tools and methods for quantitative assessment of safety in automotive systems. This is done within the scope of functional safety by developing a general method, which can be used in a Scania context, to calculate how likely faults are to give rise to a loss of some sort, i.e. damage to humans, property or the environment. Since the automotive industry, among others, is dealing with implementations of standards for functional safety, the goal of the developed model is also to make use of necessary work already carried out in these implementations, e.g. safety integrity level assessments. Furthermore, the possibilities for this method to be implemented using some modeling language such as Modelica are investigated. Questions discussed in this thesis are summarized as:

- How would a general method for quantitative safety assessment look like?
- Is it possible to adjust the general method to make use of relevant functional safety standards?
- How can Modelica be used as a tool to implement quantitative safety assessment?
- Advantages and disadvantages with quantitative safety assessment methods in general.

1.3.1 Limitations

Since the area of quantitative safety assessment has not been covered to any greater extent previously in the automotive industry, this thesis mainly functions as a pre-study to give an overview of possible ways to regard the problem of functional safety. With this in mind, this thesis is limited towards possible implementations at Scania. Furthermore, only accidents associated with software and/or hardware failures are of importance and examples used merely function as a way to clarify how the method is meant to be implemented, i.e. assessments have not been statistically validated.

1.4 Method

In order to derive a complete method for quantitative safety assessment, several parts are considered. This includes understanding relevant theory, understanding mechanisms of an accident, understanding functional safety standards, understanding hardware and software faults and failures, and understanding the chosen modeling language.

To acquire this understanding, extensive literature studies are performed. These literature studies are based on both relevant academic articles and theoretical books on functional safety related topics. The obtained literature is used both as a way to understand relevant theory and as a basis for empirical reasoning when deriving safety models. The chosen modeling language is Modelica. This is chosen mainly based on its extensive usage at Scania, but it also contain many other advantages (see Chapter 7). When understanding the modeling language Modelica, the literature study is supplemented with hands on experience.

To test and analyze the empirical results mainly based on literature studies, a case study using an actual Scania system, currently implemented in their trucks, is carried out. This case study aims to derive a value of expected loss associated to the system and illustrate modeling possibilities. As a part of this case study, some assessments described in the functional safety standard ISO 26262 are used. To increase the accuracy of these assessments, they are discussed in collaboration with other functional safety knowledgeable people at Scania. Furthermore, this case study also involves limited field tests, in order to further determine the accuracy of the ISO 26262 assessments.

1.5 Outline

This thesis starts with a review over relevant basic theory, Chapter 2. After this theory review, covering areas such as probability theory and Markov models, a suggestion for a general quantitative accident model is discussed in Chapter 3. This model is completely independent of functional safety standards, i.e. ISO 26262 and IEC 61508, hence a discussion on adjustments to fit these standards follow in Chapter 4. Both the non-standard model and the model based on the standards are assuming known probabilities of failures, or hazards. In reality, these probabilities need to be derived on their own, which is discussed in Chapters 5 and 6, following the presentation of the models. Furthermore, when both accident models and failure probabilities have been covered, we move on towards practical implementation. First, Chapter 7 discuss how Modelica can be used with the derived models. Up to this point, the thesis has mainly covered empirical studies, now these findings are considered using an example, namely the fuel level display system, in Chapter 8.

Finally, an overview of the developed method is given in Chapter 9, together with conclusions and topics for further research in Chapter 10.

Chapter 2

Theory Review

In this section the basic theory needed to understand the further reasoning throughout this thesis is presented. Mathematical tools such as probability theory, Bayesian networks and Markov models are described. Note that considerations upon the usefulness of those models are made in later sections. For an overview of vocabulary used in this thesis, see Appendix A.

2.1 Basic Probability Theory

Probability can be seen as in what degree of confidence an event will take place. This event is uncertain and could be exemplified by a failure in a specific hardware part. However, the event have to be associated with an outcome space, which is the possible outcome of the event, e.g. an event might have the possible outcome set $\Gamma = \{1, 2, 3, 4, 5\}$. This outcome set is then associated with a probability distribution P, which maps events to real values as

$$P(\Gamma) = 1 \tag{2.1.1}$$

$$P(\gamma) \geq 0 \quad for \quad all \quad \gamma \in \Gamma \tag{2.1.2}$$

Extensions to Equations (2.1.1) and (2.1.2) imply several interesting conditions where

$$P(\emptyset) = 0 \tag{2.1.3}$$

$$P(\gamma_1 \cup \gamma_2) = P(\gamma_1) + P(\gamma_2) - P(\gamma_1, \gamma_2) \tag{2.1.4}$$

will be of great interest [8].

Another important aspect is the one of conditional probability. This becomes apparent when there are two or more events that are dependent upon each other , e.g. one event, α denotes injury and another event, β denotes accident, hence an injury can be caused by an accident. The conditional probability are formally denoted as

$$P(\alpha|\beta) = \frac{P(\beta, \alpha)}{P(\beta)} \tag{2.1.5}$$

Note that if α and β were independent events, $P(\beta, \alpha) = P(\beta)P(\alpha)$ and hence, $P(\alpha|\beta) = P(\alpha)$. However, keeping the dependency, further investigation of Equation (2.1.5) gives

$$P(\beta, \alpha) = P(\beta)P(\alpha|\beta) \tag{2.1.6}$$

which is called the *Chain Rule* and is written more generally as

$$P(\beta_1, \ldots, \beta_k) = P(\beta_1)P(\beta_2|\beta_1)\ldots P(\beta_k|\beta_1, \ldots, \beta_{k-1}) \tag{2.1.7}$$

where $\beta_1, \ldots, \beta_k$ are events. Additionally, an important implication of the chain rule is *Bayes' Rule*, which allows derivation of conditional probabilities, based on "inverse" conditional probabilities, as

$$P(\alpha|\beta) = \frac{P(\beta|\alpha)P(\alpha)}{P(\beta)} \tag{2.1.8}$$

So far we have considered basic equations in probability theory using any types of events in a specified set. By introducing random variables, as an extension to the event notion, probabilities associated to attributes of the outcome of an event are possible. Here attributes of an accident (the *Random Variable*) might be a single car accident, two car accident, truck accident etcetera. The probability distributions over such an attribute are denoted as $P(Accident = SingleCar)$. Furthermore, the random variable can have different properties, e.g. having discrete sets of possible values or having continuous infinite sets of possible values. There exists a wide range of distributions, where the multinomial distribution and the exponential distribution are common examples of discrete and continuous distributions respectively. As an example of a distribution function, the distribution

$$p_i(x; \lambda) = \begin{cases} 1 - e^{-\lambda_i x} & \text{t} \geq 0 \\ 0 & \text{t} < 0 \end{cases} \tag{2.1.9}$$

is the exponential distribution, often used when modeling failures, where λ is a distribution parameter [8].

Finally, an important aspect in probability theory is the notion of expectation value. This value is defined as

$$\mathbb{E}(X) = \sum_x xP(x) \tag{2.1.10}$$

$$\mathbb{E}(X) = \int xp(x)dx \tag{2.1.11}$$

in the discrete and continuous case, respectively, where X is a random variable. This expectation value should be understood as the weighted average of the outcome of the associated random variable [8].

2.2 Graphical Models based on Probabilistic Principles

To be able to use the basic probability theory, described in the previous section, in implementations, where a large number of random variable are present, having complex interdependencies between each other, the use of graphical notations will be very convenient. Graphical notations that use probabilistic principles come in various types, where Bayesian networks, Influence Diagrams and various types of Markov models are described in the following sections.

2.2.1 Bayesian Networks

Bayesian networks are causal networks, where conditional probabilities represent the causal links. In Bayesian networks, one property provides a tool to model inherent uncertainty, namely the chain rule, see Equation (2.1.6). Formally a Bayesian network consists of [9, p. 33]:

- "A set of variables and a set of directed edges between variables.
- Each variable has a finite set of mutually exclusive states.
- The variables together with the directed edges form an acyclic directed graph (traditionally abbreviated DAG); a directed graph is acyclic if there is no directed path $A_1 \to \ldots \to A_n$ so that $A_1 = A_n$.
- To each variable A with parents $B_1, \ldots, B_n$, a conditional probability table $P(A|B_1, \ldots, B_n)$ is attached."

An example of a Bayesian Network is shown in Figure 2.2.1. In this example, variable A has no parent (conditional dependency), hence its *Conditional Probability Distribution, CPD*[1] only becomes P(A). In the case of variable C, the probability P(C|A,B) needs to be specified. Correspondently holds for B, E, D, F, G [9].

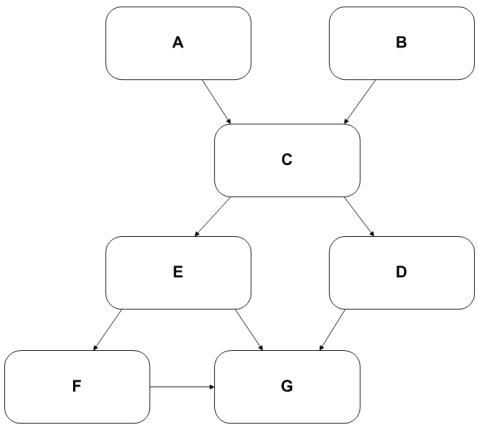


Figure 2.2.1: Example of a Bayesian network.

As mentioned, each variable is attached to a CPD. This CPD is a way to describe the conditional probabilities directly affecting a variable. A CPD over e.g. C, representing $P(C|A, B)$ in Figure 2.2.1 might look like the one in Table 2.2.1 if $A = a_0, a_1$, $B = b_0, b_1$

[1] Also referred to as Conditional Probability Table

and $C = c_0, c_1$. The individual probabilities are then given in the table, where, in this case, $P(C = c_1|A = a_0, B = b_0) = 0.2$. As seen in the table every row also has to sum up to 1, for completeness [8].

Table 2.2.1: Example CPD over C given A and B.

C	c_0	c_1
a_0b_0	0.8	0.2
a_0b_1	0.5	0.5
a_1b_0	0.3	0.7
a_1b_1	1	0

Recalling the chain rule described in Section 2.1, this rule will come to great use when using Bayesian networks. If we again consider the Bayesian network in Figure 2.2.1, which is a Bayesian network over variables $\{A, B, C, D, E, F, G\}$, then the *Joint Probability Distribution* $P(\mathcal{V})$ is derived by

$$P(A, B, C, D, E, F, G) = P(G|E, F)P(F|E)P(E|C)P(D|C)P(C|A, B)P(A)P(B) \quad (2.2.1)$$

In more general terms Equation (2.2.1) derives to

$$P(\mathcal{V}) = \prod_{i=1}^{n} P(A_i|pa(A_i)) \quad (2.2.2)$$

where $\mathcal{V} = \{A_1, \ldots, A_n\}$ is a set of variables and $pa(A_i)$ is the parent of A_i. When calculating these probabilities in practice, the number of probabilities that need to be considered might seem to become very large. However, by the use of *Variable Elimination*, the calculations needed decreases rapidly. The principle behind this is to calculate parts of the joint probability distribution separately and then marginalizing variables out of the equation [9]. This way of calculating joint probability distributions given some evidence, are used in software such as GeNIe.

Furthermore, Bayesian networks have no build in demand for causality, hence the links do not need to represent causal relationships. However, real world systems are usually bound to causality. Therefore, Bayesian networks should be made causal and it does exist various model checking methods, which among other things, check for causality violation [9].

2.2.2 Influence Diagram

Bayesian networks, as described in the previous section, merely provide tools for modeling parts of the world. It mainly supports the modeling of causal links between events. However, these models are often built in order to be used in decision making or utility assessments. An Influence Diagram extends the Bayesian networks and incorporates decision making and utility assessment in a graphical way through adding decision and utility nodes [9].

Formally an Influence diagram holds properties [9, p. 305]:

- ”there is a directed path comprising all decision nodes;

- the utility nodes have no children;
- the decision nodes and the chance nodes have a finite set of states;
- the utility nodes have no states."

Figure 2.2.2 illustrates an Influence diagram, with the introduction of a decision variable, D and a utility variable, U. In a simple example a utility might be the outcome of a game in monetary value, whereas a decision illustrates whether to call or fold. As a basis for the decision, the player has a CPD over the opponent's probability of having a better hand, here exemplified by node C.

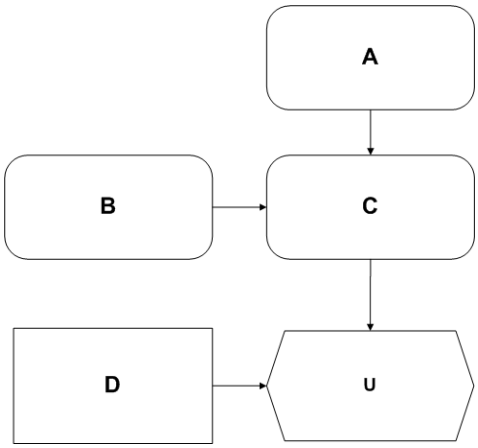


Figure 2.2.2: Example of an Influence diagram.

By using the theory of expectation value, see Section 2.1, the expected utility of a specific decision can be calculated as

$$\mathbb{E}(U, D = d_i) = \sum_C U(C, d_i)P(C|evidence) = U(C = c_0, d_i)P(C = c_0|A = a_0, B = b_0) + \ldots + U(C = c_n, d_i)P(C = c_n|A = a_0, B = b_0) \quad (2.2.3)$$

where the evidence is $A = a_0$ and $B = b_0$, and $C = c_0, \ldots, c_n$. Note that if there is no decision, or only one decision, hence only a CPD linked to the utility node, it is still possible to calculate the expected utility [9].

2.2.3 Markov Models

The basis for the understanding of Markov models is the notion of *Markov Chains*. A Markov chain is a random sequence, with the property that the following state of this sequence is only dependent on the current state, hence the process is memory less and not derived from a series of events. This property is called the Markov property [10] [8]. Formally, a Markov chain is defined as [11, p. 2]

"Consider a stochastic process

$$\{X(n), n = 0, 1, 2, \ldots\}$$

that takes on a finite or countable set M. [...] Suppose there is a fixed probability P_{ij} independent of time such that

$$P(X^{(n+1)} = i | X^{(n)} = j, X^{(n-1)} = i_{n-1}, \ldots, X^{(0)} = i_0) = P_{ij} \quad n \geq 0$$

where $i, j, i_0, i_1, \ldots, i_{n-1} \in M$. Then this is called a Markov chain process. [...] One can interpret the above probability as follows: the conditional distribution of any future state $X^{(n+1)}$ given the past states

$$X^{(0)}, X^{(2)}, \ldots, X^{(n-1)}$$

and present state $X^{(n)}$, is independent of the past states and depends on the present state only."

Based on the Markov chain, Markov models are seen as a graphical representation of these chains. These models can be seen as working in conjunction with *State Machines*, which are diagrams over transitions between different states of a system. An example state machine is shown in Figure 2.2.3, where states are illustrated as circles, and possible transitions as arrows between them. In a state machine every transition is associated with some condition, e.g. transition between states open and closed might have the condition door is closed [3].

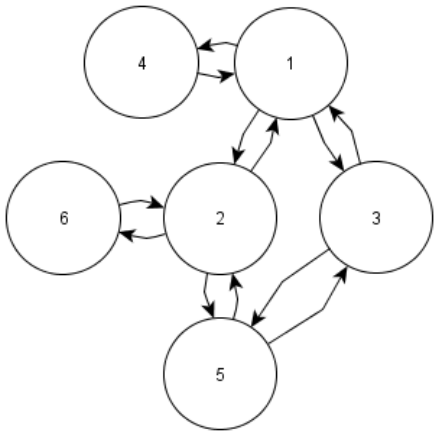


Figure 2.2.3: Example of a State Machine.

Extending the state machine, Markov models can model entire control systems, incorporating faults and failures. This is done by assigning probabilities to transitions, namely the Markov chain probabilities previously discussed. These transition probabilities are given in a transition matrix, such as

$$P = \begin{pmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,m} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m,1} & p_{1,1} & \cdots & p_{m,m} \end{pmatrix} \tag{2.2.4}$$

where p_{ij} is the probability of transitioning between state i and j. Furthermore, Markov models are commonly divided into two categories, discrete time Markov chains (models), DTMC and continuous time Markov chains (models), CTMC.

Discrete Time Markov Chains, CTMC

As the name indicates, this way of modeling Markov chains uses a discrete notion of time. If the state machine in Figure 2.2.3 is represented by discrete time transitions, we start by finding the transition frequencies (number of transitions between two states in one-time step). Based on this, a one-step transition matrix is derived using

$$P_{ij}^{(1)} = \frac{F_{ij}}{\sum_{j=1}^{m}} \tag{2.2.5}$$

The n-step transition matrix is then defined as

$$P^{(n)} = P^n \tag{2.2.6}$$

which is used to calculate the probability of being in a particular state after a given number of steps, by simply multiplying the initial probability configuration of the states with the appropriate n-step transition matrix [11].

Continuous Time Markov Chains, DTMC

Continuous time Markov chains are usable in situations where transitions between states do not occur at specific time steps as in the discrete case [11]. Here the transition probabilities, in a time interval of dt, between states i and j are given as

$$p_{ij} = \lambda_{ij} dt \tag{2.2.7}$$

where $\lambda_{ij} \geq 0$ is the constant conditional failure intensity, or *failure rate*, defined as "the probability that the component fails per unit time" [5, p. 282]. Its reciprocal is the mean time to failure, hence if no transition is possible, the transition rate becomes zero. Based on this, a transition matrix can be defined as [3]

$$\begin{aligned} P &= \begin{pmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,m} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m,1} & p_{m,2} & \cdots & p_{m,m} \end{pmatrix} = \\ &= \begin{pmatrix} 1 - \sum_{\substack{k=1 \\ k \neq 1}}^{m} \lambda_{1,k} dt & \lambda_{1,2} dt & \cdots & \lambda_{1,m} dt \\ \lambda_{2,1} dt & 1 - \sum_{\substack{k=1 \\ k \neq 2}}^{m} \lambda_{2,2} dt & \cdots & \lambda_{2,m} dt \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{m,1} dt & \lambda_{m,2} dt & \cdots & 1 - \sum_{\substack{k=1 \\ k \neq m}}^{m} \lambda_{m,m} dt \end{pmatrix} \end{aligned} \tag{2.2.8}$$

With the transition matrix derived and ready to use, determining probabilities of being in a specific state after a given time, is done by defining $Q_j(t + dt)$, for state j at time t+dt.

Assuming a Markov model with states 1 and 2, it follows that

$$Q_2(t+dt) = \lambda_{1,2}dt(1-Q_2(t)) + (1-\lambda_{2,1}dt)Q_2(t) \tag{2.2.9}$$

$$\Rightarrow$$

$$Q_2(t+dt) - Q_2(t) = -dt(\lambda_{1,2}+\lambda_{2,1})Q_2(t) + \lambda_{1,2}dt \tag{2.2.10}$$

$$\Rightarrow$$

$$\frac{dQ_2(t)}{dt} = -(\lambda_{1,2}+\lambda_{2,1})Q_2(t) + \lambda_{1,2} \tag{2.2.11}$$

$$\Rightarrow$$

$$Q_2(t) = \frac{\lambda_{1,2}}{\lambda_{1,2}+\lambda_{2,1}}(1-e^{-(\lambda_{1,2}+\lambda_{2,1})t}) \tag{2.2.12}$$

using initial condition $Q_2(0) = 0$ [5]. In a model with n states, the probability of being in state i at time t yield

$$Q_i(t) = \frac{\lambda_{in}}{\lambda_{in}+\lambda_{out}}(1-e^{-(\lambda_{in}+\lambda_{out})t}) \tag{2.2.13}$$

2.2.4 Variations of Markov Models

By expanding the conventional Markov models, as previously described, several other properties of systems can be modeled. Extensive research is currently being made on how Markov models can be expanded to suit various needs, where *Semi-Markov Models*, *Markov Random Field*, *Multi-Phase Markov Models* and *Petri Nets* are interesting examples.

Semi-Markov Models

In CTMCs the time between transitions, or the time the system spends in a state, are assumed to be exponentially distributed (Equation (2.1.9)). Since this might not be a correct assumption in all cases a more general Markov model was developed, the Semi-Markov model. This model allows arbitrary distributions dependent on the states connected by a transition [12].

Markov Random Field

Markov Random Field, or Markov Network, is seen as a further generalization of the more common CTMC. It is undirectional, allowing different distributions between variables and dependency on neighboring states in any direction. A system with variables

$$X_1, X_2, \ldots, X_n \tag{2.2.14}$$

is represented as a Markov Network with various undirectional dependencies, e.g. as in figure 2.2.3 without directions. To each of the possible combinations of states, there has to be a general-purpose function, $\phi(X_i, X_j)$ described, which function as a description on how likely

those combinations/transitions are. Combining all these general-purpose functions together with a normalizing constant yields

$$P(X_1, \ldots, X_n) = \frac{1}{Z} \prod_i \phi(X_i, Pa_{X_i}) \tag{2.2.15}$$

where

$$Z = \sum_{X_i, \ldots, X_n} \prod_i \phi(X_i, Pa_{X_i}) \tag{2.2.16}$$

By summing out variables, individual $P(X_i)$ can be calculated [8] [13].

Multi-Phase Markov Models

CTMCs handles transitions based on random events, where these events occur based on average rates, giving continuous probability flows between states. However, regular CTMCs fail to model deterministic restoration actions, such as situations where a transition occur at a given time. Here Multi-Phase Markov Models defines a time, at which re-initialization of the state probabilities takes place [14].

Petri Nets

When modeling discrete-event dynamic systems Petri nets is one of the more common graphical tools. As CTMC, Petri nets are directed graphs as most of the other models discussed, but with the difference that it uses tokens. These tokens are distributed to states, where a token indicates that a state is active or that the state uses a particular set of data depending on the number of tokens currently in the state. This implies the main different to the previous models, namely the possibility of having several states active simultaneously [15].

Chapter 3

Suggestion for a General Accident Model

Based on Kumamoto and Henley [5], Leveson [1] and Neil et al. [2] a basic view of structuring dependability of a system can be illustrated as a three-step model, shown in Figure 3.0.1. The structure links causes and consequences to dependability properties of a system, e.g. reliability or safety, here called system properties. Note; different node shapes merely function as a way to clearly differentiate between nodes. If a cause exists, the event or system property will lead to a consequence of some sort. To exemplify, a common cause could be usage and a consequence high maintenance. However, to implement the model, the dependability properties need to be discussed individually.

Figure 3.0.1: Basic dependability model.

3.1 Accident Model

The aim of this thesis is to describe and model an accident process from system faults to injuries caused by these faults in the case of an accident. Relating back to the previous discussion, an accident is seen as a system property, i.e. a safety related event. An accident can be defined in various ways. Leveson [1] uses the definition; ”an accident is an undesired and unplanned event that results in a specified level of loss” [1, p. 175] while Merriam-Webster says that an accident is ”an unforeseen and unplanned event or circumstance” [16]. Despite these discrepancies, articles which are discussing an accident often do not define its exact meaning. Leveson’s definition argues that a specified level of loss, i.e. damage to life, property or the environment needs to be present if an accident has occurred [1]. In this thesis it is reasonable to use Leveson’s definition since it disregards such accidents without any relevant consequences.

The definition of accident given above might differ from the term ”accident” commonly used in natural language. In natural language an accident is often associated with something that cannot be avoided [1]. This definition would lead to disregarding a lot of accidents that might be prevented and hence, accidents referred to in this thesis can be prevented.

When considering the accident as the system property an understanding about its causes and consequences are important. Based on Kumamoto and Henley [5] and Leveson [1] an accident occurs due to some initiating event, or incident. This incident then leads to some sort of accident depending on what accident prevention strategies are being used, e.g. pulling over to the side of the street might avoid a serious accident when you have a flat tire. If an accident does appear it will have consequences. These consequences can be said to depend on some kind of accident management or consequence mitigation, e.g. getting the injured person to a hospital. An overview of the process can be seen in Figure 3.1.1.

Figure 3.1.1: Basic overview over an accident process.

According to Kumamoto and Henley [5] an incident is a complex event which can be divided into three categories. These are human errors, system failures and environment factors. The human errors are caused by various things such as lack of training, faulty procedures being used and workplace problems. System failures are those failures that relate to hardware and software faults and can either be random or human induced. Finally, the environment factors, or external events, are ”characteristics of the environment in which the system operates” [1, p. 70]. These are independent events and cannot be affected by any human. Examples could be; the place where a vehicle is being driven, weather conditions or number of persons at the system boundary. An expanded model where the incident has been replaced by human errors, system failures and environment factors is shown in Figure 3.1.2.

Further Leveson [1] discusses the accuracy of modeling human error as a cause of an accident. When an accident is thoroughly investigated, the conclusions mostly find that the causing factors are nonhuman related. Instead, it is more likely that a human performs positive actions to prevent an accident. The reason for this misconception is said to be that accidents avoided by humans are seen as regular operation performance. So, if human errors are mostly disguised system failures or caused by environment factors, the human action both functions as preventive and contributive in the case of an accident. Along with the development of better control systems, human interaction functions more as a monitor or backup and acts as a reaction towards the system, concluding that the risk of human interaction is mainly to fail to prevent accidents. By combining the accident prevention and human error into a human preventive action, this discussion can be introduced into the model, as in Figure 3.1.3.

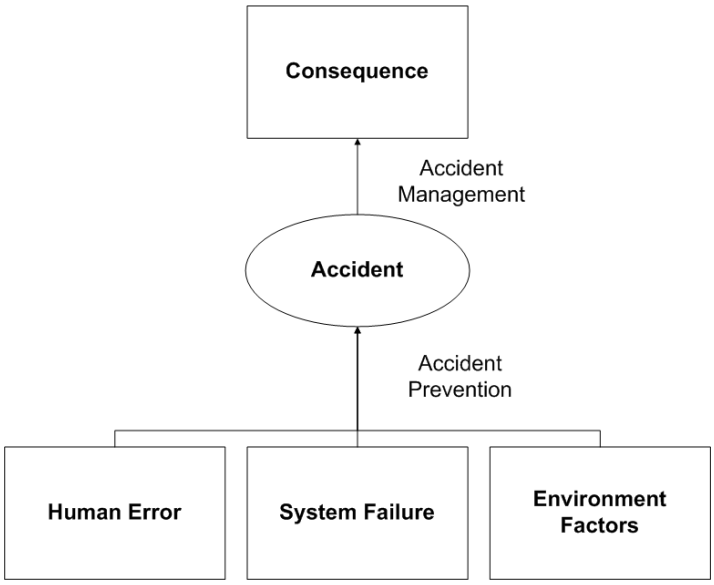


Figure 3.1.2: Expanded model over an accident process.

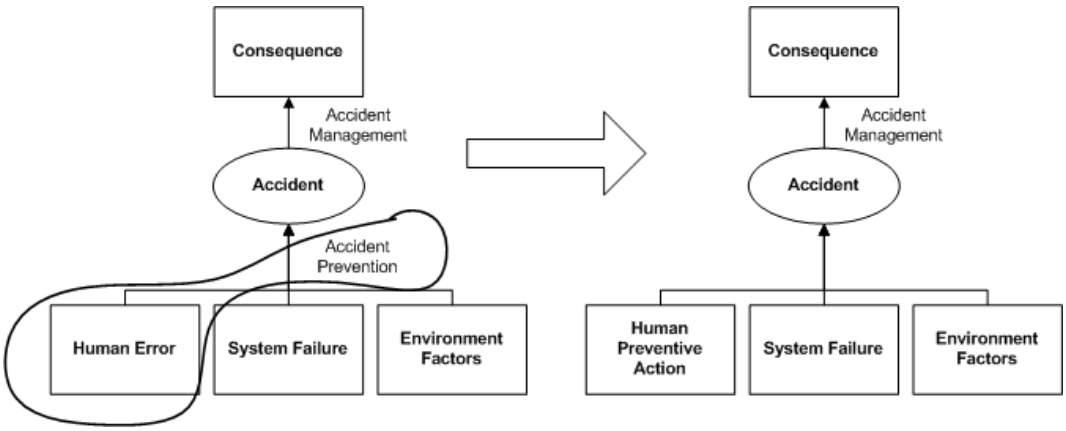


Figure 3.1.3: Human preventive action introduced in the model over an accident process.

3.2 Relating the Accident Model to the Case of an Automotive Accident

With the previous discussion about a general accident event in mind, the discussion now focuses on the causal relationship related specifically to an automotive accident. Causal relationships can be analyzed in two ways; forward analysis and backward analysis. These differ in the sense that backward analysis assumes some event and traces this backwards, while forward analysis assumes a set of failures and analyzes the effects of the failure. To fully understand the chain of events both views should be covered [5]. A forward analysis is used to define the top system events, and was carried out first. Examining the chain of events, associated with an automotive accident the process can intuitively be expressed as:

1. A combination of individual events of system failure, human error and environmental

factors makes the vehicle behave in an unintended way.

2. The Driver attempts to control this behavior using some kind of preventive action.

3. a) Driver, or other persons exposed to danger due to the vehicle, avoids an accident.

   b) Driver, or other persons exposed to danger due to the vehicle, unable to control the situation and an accident occurs.

4. The accident either leads to no injuries, mild injuries or lethal injuries, depending on the accident management.

A backwards analysis would simply start with some injury present and trace this injury back to its origin. Since the above forward analysis covers the chain of events in a very clear and intuitive way, no further discussion is held about the backwards analysis. In the scope of this thesis the dependability of technological systems is the main focus and following the previous discussion about human errors being caused by system failures and environment factors, the changes made when introducing human preventive action seem accurate.

3.3 Bayesian Modeling

In Section 2.2.1 Bayesian networks were described and Onisko et al. [17] claim its main advantage over similar schemes, is the possibility of combining existing data with expert judgment, while Neil et al. [2] argue that this is the best method to use in system dependability assessment problems. More specifically, Bayesian networks add conditional probabilities as a way to describe if an accident occurs and what consequence it may lead to. Furthermore, the system failure and environmental factors need to be addressed to avoid unnecessary work, meaning not all failures or environment factors will actually lead to an accident. How can they be defined to only consider relevant cases? Assuming our system does have vehicle level interfaces, various failures not affecting the probability of accidents have to be removed. By using the term ”hazard”, commonly used in safety assessment problems, the term ”system failures” can be resized [18]. Hazards are defined by Leveson [1] as ”a state or set of conditions of a system that, together with other conditions in the environment of the system, will lead inevitably to an accident” [1, p. 177], while other definitions, such as Kumamoto and Henley’s [5], differ in the sense that a hazard might, or has the potential, to lead to an accident, but it is not a necessity. Here a hazard is considered to be a system failure that has potential to lead to an accident. If the definition involving inevitably was to be used, hazards would be very hard to identify, since most of them have the potential to be avoided by increasing various safety measures, such as by human training. So, with this definition only, the relevant failures are covered, hence hazards can be seen as a subset of failures (see Chapter 5). This also implies that the environment factors are limited to factors that together with a hazard can lead to an accident. Hence, these environment factors, are a subset of all environment factors which here are called Operational Situations, in order to match expressions used in ISO 26262 covered in Chapter 4 [19]. With this in consideration, points 1-3 in the chain of events described in Section 3.1 can preferably be illustrated using a Bayesian network, where hazards, operational situations and human preventive action, discussed in Section 3.2, as the causes of an accident.

The model in Figure 3.3.1 illustrates a Bayesian network with dependencies between hazards, operational situations and human preventive actions as sources for accidents of any kind. In this model, the hazards are assumed to arise both individually, based on external or internal system faults, and caused by operational situations that a vehicle is currently exposed to. Furthermore, the human preventive actions are modeled as dependent on hazards and operational situations, while accidents are dependent on combinations of the three nodes previously mentioned.

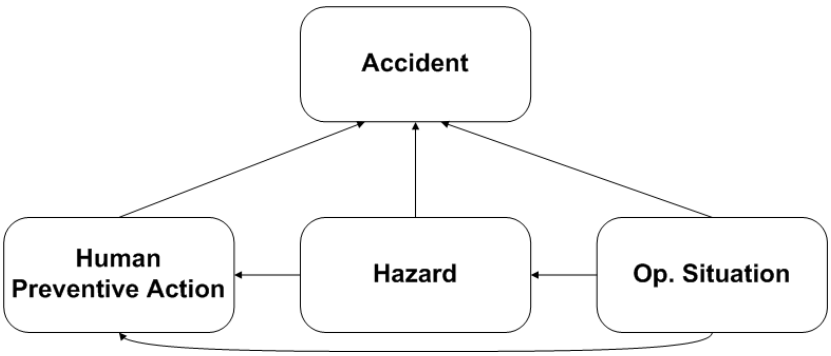


Figure 3.3.1: Bayesian network of an automotive safety related incident.

Example: Accident
A combination of a hazard, an operational situation and a human preventive action might lead to different accidents defined by various probabilities. Assuming only one possible hazard (plus no hazard, h_0); loss of braking h_1, two operational situations; driving in a parking lot, o_0 and driving in a street, o_1 and one preventive action (plus no preventive action, k_0); steer away from harm, k_1. Combinations of these variables might lead to various accidents, say; crash into other vehicle (a_1), or slight bump into vehicle in front of oneself (a_2) (plus no accident, a_0). The CPD over accident given these hazards, operational situations and human preventive actions then follows in Table 3.3.1, where probabilities has been given without any specific consideration.

Table 3.3.1: Example CPD over Accident given Hazard, H, Operational Situation, O and Human Preventive Action, K.

A	a_0	a_1	a_2
$o_0h_0k_0$	1	0	0
$o_0h_0k_1$	1	0	0
$o_0h_1k_0$	0.5	0	0.5
$o_0h_1k_1$	0.8	0	0.2
$o_1h_0k_0$	1	0	0
$o_1h_0k_1$	1	0	0
$o_1h_1k_0$	0.1	0.6	0.3
$o_1h_1k_1$	0.6	0.1	0.3

3.3.1 Loss associated with accidents

In accordance with most safety assessment problems, the main area of interest is the possible outcome of any accident. To be able to determine whether this outcome is associated with

a specific hazard somewhere in the system or an operational situation each of the possible accidents has to be quantified by specifying a loss [9]. This loss includes damage to life, property or the environment as a consequence of an accident [1]. By defining loss, its levels can be defined by natural language or by a classification. In the standard IEC 61508, four levels are defined classifying loss, giving us an example of how this might be done [20]. Extensive research has also been done at various institutions such as insurance companies and in traffic safety investment strategy analyses to relate different consequences, such as life and property losses, to each other resulting in a general loss classification where human lives relate to a monetary value [21] [22].

Adding loss as a utility node allows us to properly model its properties, and the model in Figure 3.3.1 expands it into Figure 3.3.2, which completes the chain of events in Section 3.1 [9]. It should be noted that the figure lacks the term accident management. Since the model is supposed to be used in a general context, the accident management is assumed to follow typical patterns, e.g. an injured person will be taken to the hospital etc. This is a simplification which is made to make an accurate loss assessment. Accordingly we make the following assumption:

Assumption 3.3.1. *Loss is based on the most typical way to manage the accident, such as getting people to the hospital when needed or putting out a fire.*

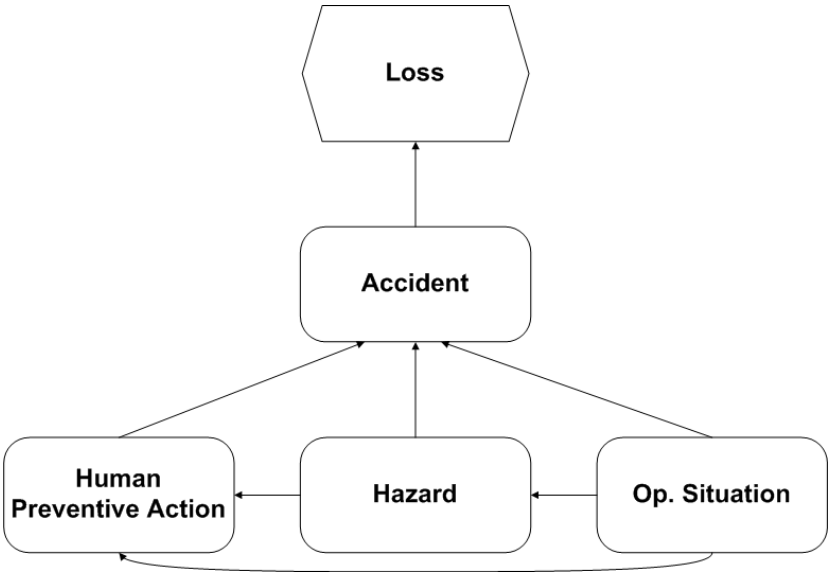


Figure 3.3.2: Model of an automotive safety related incident incorporating fatality assessments.

Example: Loss
Using the accidents discussed in *Example: Accident*, those accidents can be classified by level of loss. Assume the levels; no loss, light injury to human, severe injury to human and death to human exists. Table 3.3.2 indicates the classifications made.

Table 3.3.2: Loss related to accident.

a_0	no loss
a_1	light injury to human
a_2	death to human
a_3	death to 2-10 humans
a_4	catastrophe, more than 10 humans die

3.4 Simplifications of Reality

The model derived in Section 3.1 shows a general view in which an accident happens due to on human preventive actions, hazards and operational situations. These accidents can then be classified introducing the node loss. However, there are some problems associated with this model. These are:

1. The number of probabilities to consider easily gets extremely large. To be able to use Bayesian networks in practice, some template or method needs to be developed to support both small and large scale assessment problems, that is to avoid too large a number or probabilities something has to be done [2].

2. It is hard to estimate all of the necessary probabilities, as this requires extensive statistics gathering.

3. One has to decide on how to discretize the rage of accidents, operational situations, hazards and human preventive actions, that is the resolution of the variables. As an example, the number of operational situations in the real world is almost limitless.

By looking at the problems it is obvious that something has to be done to prevent at least some of these problems. A reasonable way to address the issue would be to simplify the model.

First we consider the dependencies in the model. In the real world model, hazards are believed to arise due to operational situations and internal or external faults. A simplified model where hazards and operational situations are independent events make it possible to assess the probability of every hazard based on hardware and software failure rates only, which could be exemplified by subcontractors. This simplification relates to problem 1. Note that this is a general simplification and the dependency might have to be reinstated when modeling some environment critical functions. An example of this is shown in Chapter 8. Accordingly we have the following assumption:

Assumption 3.4.1. *Operational situations and hazards are independent events.*

Now, let us consider the accidents. In the scope of the thesis the safety assessment of a specific system is of interest, hence only the worst case scenarios are relevant since these require the highest level of safety. All other accidents are considered as no accidents for completeness reasons. This way of simplifying can be seen as a ”noisy-or” generalization [17]. This simplification relates to problem 3 and derives to the following assumption:

Assumption 3.4.2. *In the accident model only worst case accidents for every combination of hazards, operational situations and human preventive actions are considered.*

When it comes to problem 2 there are no obvious simplifications. One way to somewhat simplify is to assign probabilities logarithmically, which will lower the accuracy. The slightly changed model incorporating the above assumptions is shown in Figure 3.4.1. Note the change of accident to worst accident and removed dependability between hazard and operational situation.

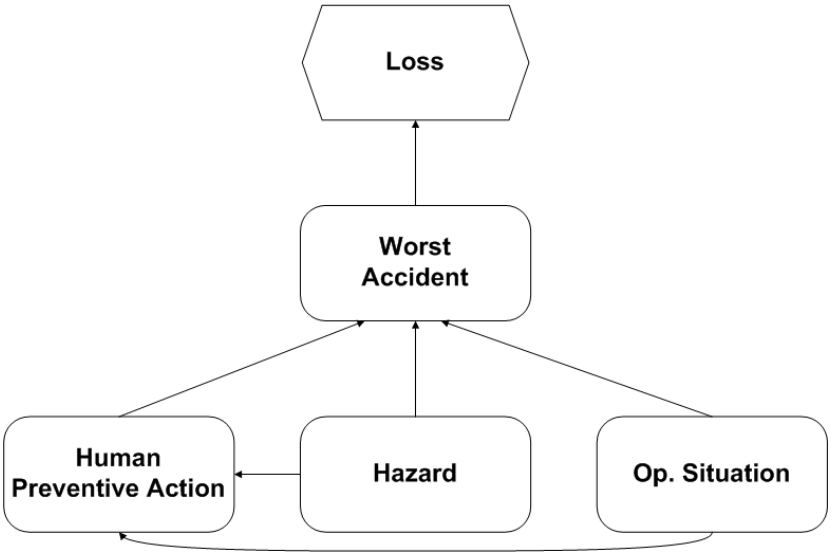


Figure 3.4.1: Simplified model of an automotive safety related incident incorporating fatality assessments.

Chapter 4

Models based on ISO 26262

Based on the standard IEC 61508, ISO 26262 was developed as a standard with a specific focus towards the automotive industry. ISO 26262 aims to apply to all safety-related issues concerning electrical, electronic and software components. This standard was created as an answer to the emergence of safety as an increasingly important factor in automotive development. Content wise, ISO 26262 provides an automotive safety life cycle, a risk-based determination of integrity levels and a requirement specification along with requirements for validation, confirmation and supplier relations [19].

In Chapter 3, a general model of an automotive accident-process was discussed. The model aims to assess loss of a system. In practice, the automotive industry is, or will be, working towards the ISO standard 26262 covering functional safety [23]. Since this standard has the same purpose, i.e. safe technological systems, as the previous model, an understanding of ISO 26262 is useful. Therefore, by relating the general model, and if possible adjusting it, to ISO 26262, having to work with two separate views might be avoided.

In ISO 26262 one sub-phase of the standard is called hazard analysis and risk assessment. This sub-phase objective is to identify and categorize possible hazards and to formulate safety goals to prevent hazardous events. To initiate a hazard analysis, a system definition has to be in place, on which the analysis is based. ISO 26262 uses the classifications; probability of exposure, controllability and severity to assess each accident. Based on the definitions of these classifications the previous Bayesian network can be adjusted, illustrating the dependencies between these classifications [19].

Recalling the basic model in Figure 3.4.1, the differences between ISO 26262 definitions mainly concern loss, worst accident and human preventive action. In the case of loss, the standard only focuses on loss of human lives where it differentiates in levels spanning from no injury, up to someone dying in an accident, based on AIS stages [19]. This means, to fit the standard, the loss in the general model will have to change from a large number of possible losses into a level of injury. Therefore, when discussing the ISO 26262 the loss node will be called injury. The worst accident (see Figure 3.4.1) and human preventive action (see Figure 3.4.1) nodes do bring a more extensive problem. By conducting a hazard analysis (see Section 8.1.1) every combination of hazard and operational situation, called hazardous event, will be classified by severity, controllability and probability of exposure and given a safety level, called ASIL,

illustrated in Figure 4.0.1. The probability of exposure is directly mapped to the operational situation and fits the model. However, controllability and severity will have an impact on the model.

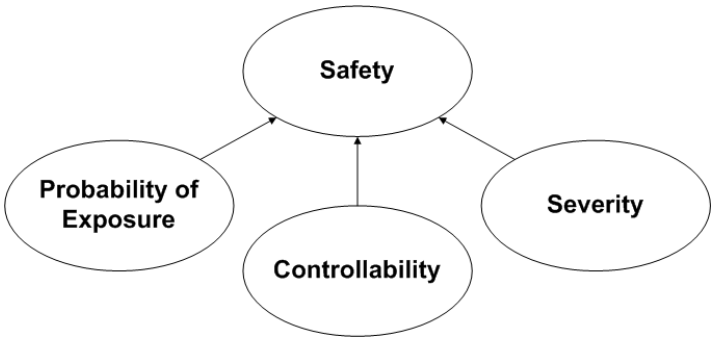


Figure 4.0.1: ISO 26262 classification of safety.

The severity classification assesses the worst possible injury following a hazardous event, while the controllability classification assesses how hard it is to handle a specific hazardous event [19]. One could argue that a combination of these two classifications will form the worst accident (from the general model) classification, incorporating human preventive action. For example, classifying a hazardous event with the highest severity level, meaning probable death, with a low controllability, meaning easy to control/prevent, would mean a somewhat less severe probable worst accident than if a high controllability level was also given. If accepting this view and merging the nodes worst accident and human preventive actions, the previous model will convert into the one in Figure 4.0.2, where the probability assessments are discussed in the next section.

In the new model changes have been made concerning human preventive action, worst accident and loss. It introduces *Injury* and *26262 Worst Accident* (combines previous worst accident and human preventive action). Examples of CPDs and further probabilistic reasoning follows in the next section.

Example: Controllability and Severity
Assume we have a set of combinations of operational situations and hazards, h_0o_0, h_0o_1, h_1o_0 and h_1o_1. Then combinations, or hazardous events, can lead to different accidents, a_0, a_1, a_2, a_3, a_4, a_5 and a_6. These accidents are, in a hazard analysis, severity and controllability classified. The accident with the highest combinations of classifications, possible for the specific hazardous event, will then be considered as the worst accident and will be used in the model. Table 4.0.1 illustrates the process clearly.

4.1 Probabilistic Translations

To be able to use the classification of hazardous events described in detail in Section A.3 the natural language level definitions of controllability, severity and probability of exposure are translated using probabilistic notations to facilitate the calculations.

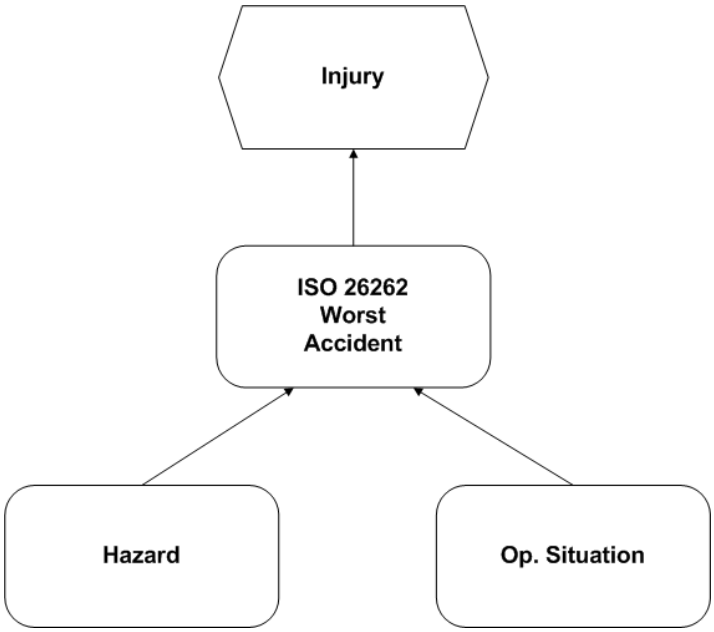


Figure 4.0.2: Converted model to fit ISO 26262.

Table 4.0.1: Using severity and controllability classification when assessing the worst accident, where l/h meaning low severity and high controllability (l = low, m = medium, h = high). Note that a_0 now corresponds to all other accidents except the worst accident.

(a) Assessing and mapping of the hazardous events to possible accidents and classifications.

A_w	a_0	a_1	a_2	a_3	a_4	a_5	a_6
h_0o_0	l/h	0	0	0	0	0	m/m
h_0o_1	l/h	h/h	0	h/m	0	l/m	m/m
h_1o_0	l/h	h/h	m/l	0	h/l	0	m/m
h_1o_1	l/h	0	m/l	h/m	h/l	l/m	m/m

(b) Highest combinations, i.e. singling out the worst accidents.

A_w	a_0	a_1	a_2	a_3	a_4	a_5	a_6
h_0o_0	l/h	0	0	0	0	0	m/m
h_0o_1	l/h	0	0	h/m	0	0	0
h_1o_0	l/h	0	m/l	0	0	0	0
h_1o_1	l/h	0	0	h/m	0	0	0

4.1.1 Worst Accident, A_w

In accordance with the previous discussion, the worst accident preferably combines the severity and controllability classifications gathered in a hazard analysis. These classifications are, together with the operational situation, given an ASIL level. To be able to use severity and controllability classifications, a safety level has to be determined only dependent on those factors. Looking at Table A.3.4 and disregarding the probability of exposure by choosing the highest ASIL possible for each combination of severity and controllability, because we do not

want to miss any safety issues, the table transfers to the one in Table 4.1.1.

Table 4.1.1: ASIL classification of Severity, S and Controllability, C. Any combination containing S0 or C0 are classified as $\mathcal{QM}_{sc}$.

Severity Class	**Controllability Class**		
	C1	**C2**	**C3**
S1	$\mathcal{QM}_{sc}$	$\mathcal{A}_{sc}$	$\mathcal{B}_{sc}$
S2	$\mathcal{A}_{sc}$	$\mathcal{B}_{sc}$	$\mathcal{C}_{sc}$
S3	$\mathcal{B}_{sc}$	$\mathcal{C}_{sc}$	$\mathcal{D}_{sc}$

Here the ASIL corresponds to probabilities of the worst possible accident occurring (compare to *Example: Controllability and Severity*). Now $\mathcal{QM}_{sc}$-$\mathcal{D}_{sc}$ can be converted into probabilities. This conversion has not been covered in the discussed standards. The standards merely discuss the probabilistic conversion of required failure rates (see Chapter 5). However, based on the definitions of the different severity and controllability classes, conversions to probabilistic terms might be possible. Table 4.1.2a and Table 4.1.2b show one possible conversion based on ISO 26262 [19].

Table 4.1.2: Translating severity and controllability levels to probabilities separated by orders of magnitude.

(a) Severity to probability.

Level	Probability of fatal injury
S0	$\leq 10^{-3}$
S1	10^{-2}
S2	10^{-1}
S3	10^{-0}

(b) Controllability to probability.

Level	Probability not to control situation
C0	$\leq 10^{-3}$
C1	10^{-2}
C2	10^{-1}
C3	10^{-0}

In Table 4.1.3 one possible translation ASIL QM_{sc}-D_{sc} is made, which is based on Table 4.1.1 and probabilities used in Table 4.1.2, e.g. $D_{sc} = C3 \cdot S3 = 10^{-0} \cdot 10^{-0} = 10^{-0}$.

Table 4.1.3: ASIL translated into probabilities.

$ASIL_{sc}$	$\mathcal{QM}_{sc}$	$\mathcal{A}_{sc}$	$\mathcal{B}_{sc}$	$\mathcal{C}_{sc}$	$\mathcal{D}_{sc}$
Magnitude	$\leq 10^{-4}$	10^{-3}	10^{-2}	10^{-1}	10^{-0}

Example: ASIL as probabilities
Assume there are two worst accidents, a_1 and a_2 associated with a number of hazardous events. From figure 4.0.2, the worst accident is dependent as

$$P(A_w, H, O) = P(A_w|O, H)P(O)P(H) \tag{4.1.1}$$

Assuming P(O) and P(H) are known, an example CPD of A_w given O and H is shown in Table 4.1.4. Note that every hazardous event only associates to one worst accident and a_0 corresponding to all other accidents.

Note that the sum in Table 4.1.4 does not clearly summarize to one. This is because the value 10^{-0} shall be seen as the range 10^{-1}-10^{-0}, and 10^{-4} as 10^{-5}-10^{-4}. This means, e.g. $10^{-0} + 10^{-4} \approx 1$. Not exactly specifying the values simplifies the assessments greatly and will be used throughout this thesis.

Table 4.1.4: Example CPD over A_w given O and H.

A_w	a_0	a_1	a_2
$h_0 o_0$	10^{-0}	$\mathcal{QM}_{sc} = 10^{-4}$	0
$h_0 o_1$	10^{-0}	0	$\mathcal{A}_{sc} = 10^{-3}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$h_n o_n$	10^{-0}	$\mathcal{B}_{sc} = 10^{-2}$	0

4.1.2 Operational Situation, O

Each operational situation is associated with a probability of exposure [19]. The probability of exposure of an operational situation, o_i is classified in a hazard analysis. These classes are based on numeric probabilities of a situation occurring per hour, and are clarified in Table 4.1.5 separated by orders of magnitude.

Table 4.1.5: Probability of exposure classes translated into probabilities.

$P(O)$ **Class**	**E0**	**E1**	**E2**	**E3**	**E4**
Magnitude	$\leq 10^{-4}$	10^{-3}	10^{-2}	10^{-1}	10^{-0}

An example CPD, over operational situations, where n operational situations relevant for a system has been identified in a hazard analysis, is shown in Table 4.1.6.

Table 4.1.6: Example CPD over O where o_0 is all operational situations not covered by o_1-o_n.

O	o_0	o_1	...	o_n
	10^{-0}	10^{-3}	...	10^{-1}

4.1.3 Injury, I

Let us consider the injury node. This is a utility node and represents the outcome of an accident [9]. As discussed before, in accordance with ISO 26262, the loss spans from no injury to death. The utility of an injury can be assessed in different ways. In accordance with Section 3.3.1 one way would be to use utilities as exemplified in Table 4.1.7a. This method uses levels based on natural language and is valuable when looking at the possible outcome of a specific accident. If instead we want to be able to assess the entire system safety, incorporating all identified accidents, operational situations and hazards, the need for a numeric utility classification becomes necessary. One way to do this is to introduce the probability of fatality, P(fatality$|A_w$). This denotes the probability of fatal consequences of an accident, e.g. P(fatality$|A_w$)=1 in the case of extremely critical injuries and P(fatality$|A_w$)=0 in the case of no injuries. The values between 0 and 1 are assessments of how serious the injury is compared to death. For simplicity the values are separated by orders of magnitude. Table 4.1.7b illustrates an example.

Since we are interested in the probability that a system failure will lead to fatal injuries, the utility is the probability of death associated with this specific worst accident. With

Table 4.1.7: Different ways of defining the utility of injury.

(a) Example of fatality when using natural language based levels.

Injury
no injuries
light injuries
moderate injuries
critical injuries
extremely critical injuries
severe injuries, not life-threatening
severe injuries, life-threatening

(b) Example of P(fatality$|A_w$).

Accident	Injury
a_0	0
a_1	10^{-3}
a_2	10^{-1}
a_3	10^{-0}
a_4	10^{-3}

$\mathbb{E}(I)$ representing the system's total expected loss, i.e. probability of fatality, and the utility function $I(A_w) = P(fatality|A_w)$, we have

$$\mathbb{E}(I) = \sum_{A_w} \sum_{H} \sum_{O} I(A_w) P(A_w, H, O) \tag{4.1.2}$$

Expanding Equation (4.1.2) gives

$$\begin{aligned} \mathbb{E}(I) &= \sum_{A_w} \sum_{H} \sum_{O} I(A_w) P(A_w, H, O) = \\ &= \sum_{A_w} \sum_{H} \sum_{O} P(fatality|A_w) P(A_w|H, O) P(H) P(O) \end{aligned} \tag{4.1.3}$$

Example: System Fatality
To clarify the line of thought, a simple example of how to calculate the probability of fatality for a given system follows. Assume that we have two hazards (h_1, h_2), and two relevant operational situations (o_0, o_1), with probabilities given in Table 4.1.8a and Table 4.1.8b. Then, combinations of these are assessed and given an $ASIL_{sc}$ class. The highest $ASIL_{sc}$ classification for each hazardous event is considered to be the worst accident. These worst accidents (a_1, a_2), combined with its $ASIL_{sc}$, are translated into probabilities as in Table 4.1.8c. Now, the accidents need to be individually fatality assessed. An example of this is shown in Table 4.1.8d. By using all the information now available, the total system probability of fatality

can be calculated. By using Equation (4.1.3), we obtain

$$\begin{aligned}\mathbb{E}(I) &= \sum_{i=0}^{2}\sum_{j=0}^{2}\sum_{k=0}^{1} P(fatality|a_i)P(a_i,h_j,o_k) = \\ &= \sum_{i=0}^{2}\sum_{j=0}^{2}\sum_{k=0}^{1} P(fatality|a_i)P(a_i|h_j,o_k)P(h_j)P(o_k) = \\ &= 0 + 10^{-3}\cdot 10^{-4}\cdot 10^{-0}\cdot 10^{-0} + \ldots + 10^{-1}\cdot 10^{-0}\cdot 10^{-5}\cdot 10^{-2} = \\ &= 10^{-7} \end{aligned} \tag{4.1.4}$$

The result from Equation (4.1.4) says that there is a 10^{-7} probability that someone will die per hour follow usage of the system. Note, in this example the worst accidents were not specifically defined and the values given to the probability of fatality not given much consideration. An accurate example must be preceded by an investigation about probable accident outcome, along with injury to death value mapping.

Table 4.1.8: Example: System Fatality.

(a) Example of probability of exposure of an operational situation per hour.

$P(o_0)$	10^{-0}
$P(o_1)$	10^{-2}

(b) Example of probability of hazard per hour.

$P(h_0)$	10^{-0}
$P(h_1)$	10^{-7}
$P(h_2)$	10^{-5}

(c) Example of P(A_w|O,H).

A_w	a_0	a_1	a_2
h_0o_0	10^{-0}	$\mathcal{QM}_{sc} = 10^{-4}$	0
h_0o_1	10^{-0}	0	$\mathcal{A}_{sc} = 10^{-3}$
h_1o_0	10^{-0}	$\mathcal{C}_{sc} = 10^{-1}$	0
h_1o_1	10^{-0}	$\mathcal{C}_{sc} = 10^{-1}$	0
h_2o_0	10^{-0}	0	$\mathcal{B}_{sc} = 10^{-2}$
h_2o_1	10^{-0}	0	$\mathcal{D}_{sc} = 10^{-0}$

(d) Example of probability of fatality.

Accident	Injury
a_0	0
a_1	10^{-3}
a_2	10^{-1}

4.2 Further discussion on ISO 26262

While adjusting the general model to fit ISO 26262, an assessment not considered in the standard, namely the probability of fatality, or injury outcome, assessment of every worst accident, had to be made. Preferably, this extra assessment should be avoided, we should only consider necessary assessments according to the ISO 26262 standard and then just implement using a model. Is it possible to further adjust the model so it is completely decided by assessments done in ISO 26262?

One possible way to do this would be to somehow use the severity classification as the loss. The problem here would then be to introduce the controllability classifications to reduce the severity. Recalling the simplified general model in Figure 3.4.1, one possible way of solving

the problem would be to merge the worst accident node and the loss node. By doing so, no loss assessment for each worst accident needs to be made. Instead, by introducing the severity as used in ISO 26262, i.e. worst potential loss of a hazardous event, as a utility node, this assessment is made in the ISO 26262 hazard analysis. The severity node is now dependent upon how likely a hazardous event is controlled. This controlling mechanism has previously been called human preventive actions. By using the ISO 26262 controllability classification as the incorporated probabilities, the node name is changed to 26262 preventive action. Hence, this way of modeling an accident, instead of merging the nodes human preventive action and accident (as in Figure 4.0.2), mergers the node's worst accident and loss. Severity can then be referred to as loss of worst accident and controllability as human preventive action. The transformation of the model, compared with former transformation, is shown in Figure 4.2.1.

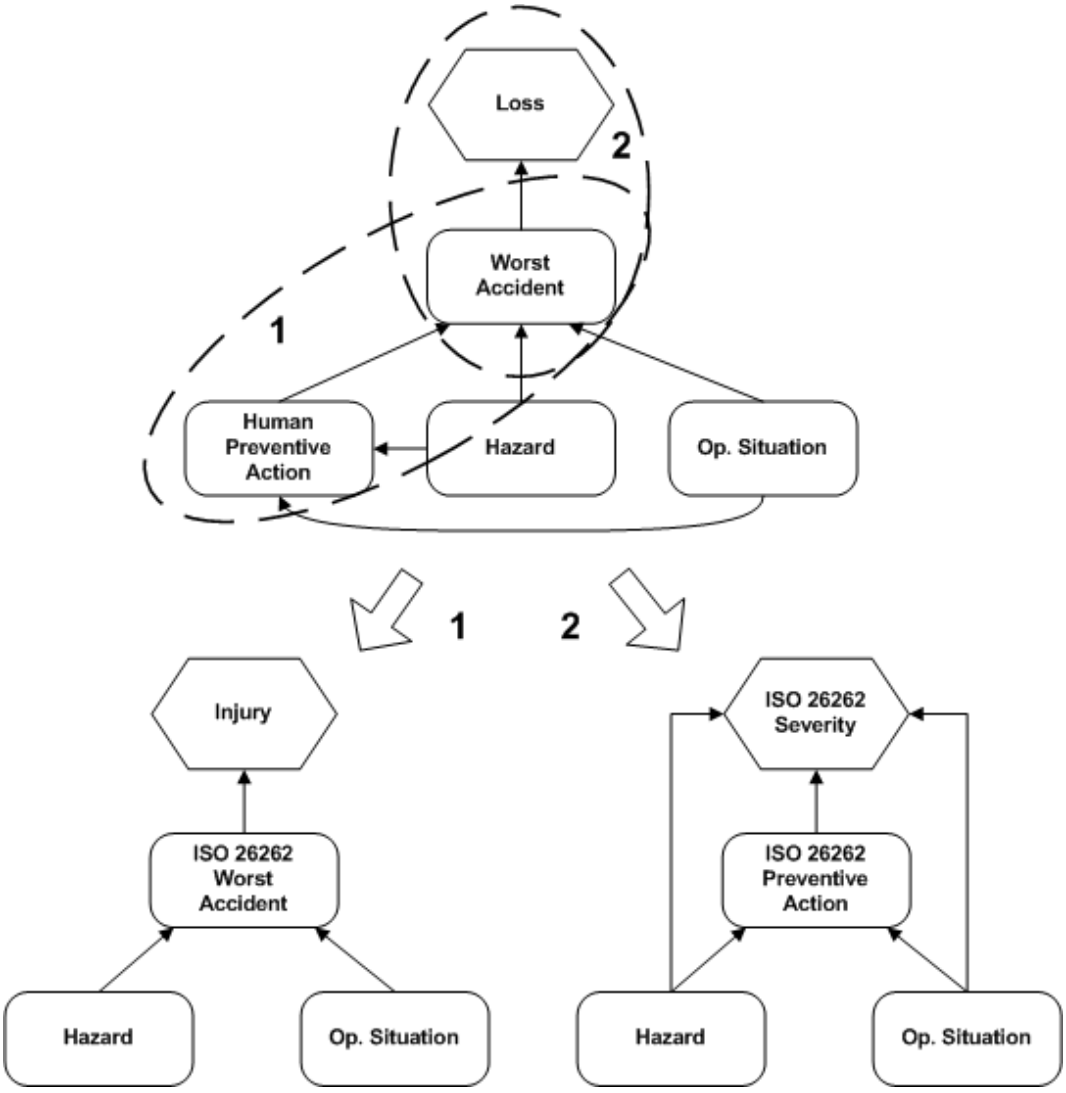


Figure 4.2.1: Number 1 illustrates the ISO based model described in previous section. Number 2 illustrates the new ISO based model where only assessments described in ISO 26262 are used. The figure also clearly compares the models as to how they relate to the general model derived in Chapter 3.

Example: System Fatality 2
Assume the severity and controllability levels are translated as in Table 4.1.2. Then hazardous events (o_0h_0,o_1h_0,o_0h_1 and o_1h_1), with individual probabilities given in Table 4.2.1a and 4.2.1b, are severity and controllability classified as in Table 4.2.1c and Table 4.2.1d. Now, by using the same method as in Section 4.1.3, and the utility function $S(C = c_1, H, O) = P(\text{fatal injury}|C = c_1, O, H)$, we obtain the expected severity as

$$\begin{aligned}\mathbb{E}(S) &= \sum_H \sum_O S(C = c_1, H, O)P(C = c_1, H, O) = \\ &= \sum_H \sum_O P(\text{fatal injury}|C = c_1, O, H)P(C = c_1|H, O)P(H)P(O) = \\ &= 0 + 10^{-0} \cdot 10^{-0} \cdot 10^{-7} \cdot 10^{-0} + \ldots + 10^{-1} \cdot 10^{-1} \cdot 10^{-7} \cdot 10^{-2} = \\ &= 10^{-7} \end{aligned} \tag{4.2.1}$$

Table 4.2.1: Result of a hazard analysis. Note that ASIL level are not used.

(a) Example of probability of exposure of an operational situation per hour.

$P(o_0)$	10^{-0}
$P(o_1)$	10^{-2}

(b) Example of probability of hazard per hour.

$P(h_0)$	10^{-0}
$P(h_1)$	10^{-7}

(c) Severity.

S	Loss = P(Fatal injury)
$c_1o_0h_0$	S0$\leq 10^{-3}$
$c_1o_1h_0$	S1=10^{-2}
$c_1o_0h_1$	S3=10^{-0}
$c_1o_1h_1$	S2=10^{-1}

(d) Preventive action.

C	P(Control situation)=c_0	P(not Control situation)=c_1
o_0h_0	10^{-0}	C0$\leq 10^{-3}$
o_1h_0	10^{-0}	C1=10^{-2}
o_0h_1	10^{-0}	C3=10^{-0}
o_1h_1	10^{-0}	C2=10^{-1}

4.3 Safety Requirements

Due to the high risk nature of the transport business, the automotive industry requires a high safety level. In the context of this thesis the safety issues are related to injuries and system failures. Therefore requirements for the probability of lethal accidents have to be minimized. The problem then becomes, what limit on probability of fatality is safe enough? One possible approach is to consider required failure rates, which will be discussed further in Chapter 5. These failure rates define the highest allowed failure rate on each element of a system [19]. If, in the simplest case, a system only consists of one component with a required failure rate, and this component is the only cause of a hazard. Assuming this hazard always leads to a severe accident classified as ASIL D. Then it would be possible to say that this possibility is the highest allowed probability of fatality. In ISO 26262 this would give the highest allowed expectation value as [19]

$$max\mathbb{E}(I) = 10^{-8} \tag{4.3.1}$$

And in IEC 61508 the value would be (IEC, 2010)

$$max\mathbb{E}(I) = 10^{-9} \tag{4.3.2}$$

The reason for this deferens is that ISO 26262 does not differ between one death or many deaths, which IEC 61508 do [19] [20]. In the light of this, when incorporating heavy trucks and buses in an automotive context the lower limit is probably a more reasonable target. This because e.g. buses are more likely to cause death to more than 10 people, as compared to a personal car. However, in the calculations of expectation of severity or injury using the ISO model, no differentiation between one and many deaths was made. So, to be able to use that model the higher limit has to be used unless severity classifications change.

If using a real world example, we will get a clearer view if the values really are low enough, compared to how many lives we are prepared to sacrifice due to system failures. Let us discuss an example using Scania trucks in Sweden.

Example: Risk Using a Scania Truck in Sweden
A recent estimation made by Scania [6] says, that there are about 40 000 active Scania trucks operating in Sweden. Statistics has also shown that about 270 persons died in traffic in Sweden in 2010. Amongst those 270 persons, 52 were driving a heavy truck. While Scania currently have about 40% of the heavy truck market, roughly 20 persons died in a heavy Scania truck in 2010 [24] [6]. Note that the amount of people getting less serious injuries is not covered in the statistics, hence those numbers are not considered. Neither people being externally affected by Scania trucks are covered in the statistics. However, assuming every Scania truck operates about 8 hours a day, 5 days a week. Then, one person is killed in a Scania truck, 20 persons/(52 weeks*5days*8 hours) = 0.0096 times an hour. For every truck, this means a fatality rate of 0.0096/40 000 = 2,4 10^{-7}. Furthermore, unconfirmed rough estimations indicates that 2% of all accidents are caused by system failure, hence the fatality rate connected to functional safety derives to 0.02*2,4 10^{-7} = 4.6 10^{-9} deaths per hour. Since requirements are based upon one system, and roughly a truck consists of at least 10 vehicle level systems, Scania's death per hour value decreases to about 4.6 10^{-10} per system.

As seen in the previous example, required risk factors, as those in Equations (4.3.1) and (4.3.1) might be sufficient goals when designing safe technological systems or, as in the case of Scania, the values could be even lower. One incentive to develop safer systems could then be to further lower the required expected injury probability. Maybe 10^{-10} is a good number. However, this question is a complex issue with many aspects to consider. One way to look at it, which is also discussed by Leveson [1], is that the goal always has to be to eliminate hazards and thereby eliminate accidents and their consequences. This is unfortunately an impossible task and unreasonable safety requirements might lead to unbearable costs for the industry, or forced functionality compromises, losing the point in developing and producing new products. One way to tackle this problem would be to conduct some kind of risk-benefit analysis, and thereby find a reasonable safety level, which all different stakeholders can agree upon. Although, safety assessments, used in the models developed in this thesis, are hard to make completely accurate, especially those concerning extremely serious, but very unlikely events such as automotive accidents with hundreds of injured persons, and need rigorous consideration [1].

Chapter 5

Faults, Failures and Hazards

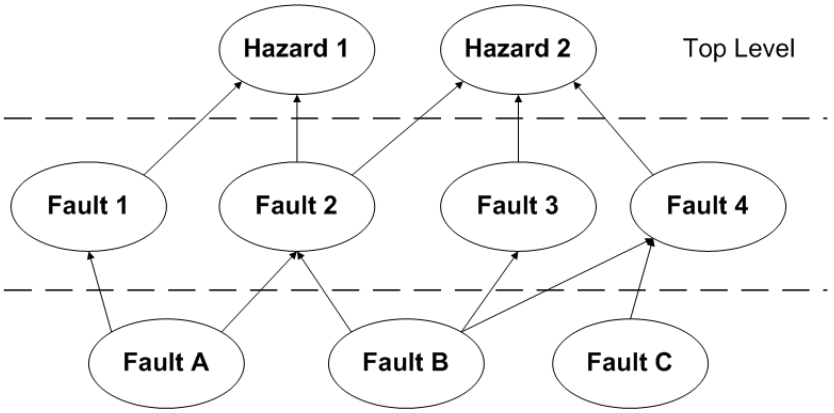
In previous sections, hazards were thought of as a subset of top level failures, i.e. those failures affecting the environment of a system and would, in the automotive case, have a direct effect to the probability of an accident. This is a recognized view and is used in the academia and the business world [1] [5] [19]. Figure 5.0.1a illustrates clearly the hazards as a subset of all possible failures. In order to discuss hazards further though, these have to arise from somewhere. Here faults become important. Faults can be either internal or external and software or hardware related. Alternatively, faults directly leading to hazards can be viewed as subsystem failures, which also has its own faults [1]. This creates a level-model illustrated in Figure 5.0.1b. In the figure only three levels are modeled and *Fault A* can be seen as a so called common cause failure since it is the cause of both *Fault 1* and *Fault 2* [5].

At this point, we have derived a general model describing an automotive accident process and adjusted it to fit the ISO 26262. This process was modeled with the use of a Bayesian network. To be able to use this model in actual calculations, hazards need to be determined and assigned a probability, determining how often they are believed to occur.

5.1 Identification of Hazards

Identifying hazards are one of the first things that need to be done when designing a system. In IEC 61508 and ISO 26262 identification of hazards is described as a part of the hazard analysis. This hazard analysis recommends methods like brainstorming and checklists to find relevant hazards [20] [19]. Additionally, Leveson [1] describe identification of hazards through what she calls a Hazard Identification Process, which is divided into several parts. These parts include determining ”what hazards might exist during operation of the system” [1, p 295], developing system design guidelines, hazard control initiation, responsibility management and determining of required work efforts needed to minimize safety problems. Here the first part concerning the actual identification of possible hazards are of interest at this point and corresponds to the identification described in the mentioned standards. Furthermore, to support the identification of hazards e.g. Kumamoto and Henley [5] suggests that a basic top level system description, illustrated by a state machine, should be used to get a clear overview of a system.

(a) Hazards as a subset of Failures.



(b) Example on how Hazards are determined by lower level internal and external faults.

Figure 5.0.1: Relationship between Faults, Failures and Hazards

Starting with a state machine, a clear and understandable view on a system is given, making it easier to avoid missing relevant hazards, along with expansion possibilities, which will be discussed later. Here we choose to use system states[1] as an inspiration for identification of hazards, which then will be assessed using methods described in the mentioned standards. To describe the system we will use a basic state machine [25]. State machines model states of a system together with its related transitions. An example of a system described by a state machine is illustrated in Figure 5.1.1, which starts from initial state 0, transitioning into state 1 and 2 when the associated condition have been fulfilled [25] [1].

When the state machine is in place we have a valid top level system description[2], which will be used as a base for identification of hazards. Since hazards are a subset of failures, which is defined as general "termination of ability of a system to perform function as required" [19, Part 1] in ISO 26262, the term failure mode will be used for a specific failure. Hence, every hazard is a failure mode and all failure modes are failures. Now, all possible failure modes, assuming an accurate state machine, can be found. Since transitions are associated with some change in a system, failures can only occur where a transition is possible. Hence, the example in Figure 5.1.1 have failure modes associated to condition 10, condition 01, etcetera. How many

[1]Referred to as Operating Mode at the top level in ISO 26262, not to be confused with Mode of Operation in IEC 61508.

[2]Can also be seen as top level system requirement or properties, describing what the system should do [26].

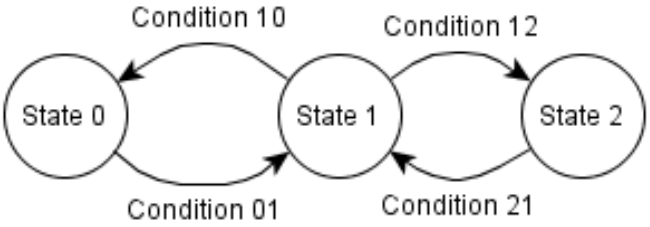


Figure 5.1.1: State Chart over an example system with state 0 as initial state.

failure modes associated with each condition can differ depending on the condition and are seen as transitions happening when they are not supposed to (e.g. too early), transitions not happen when they are supposed to (e.g. too late) or transition to a state displaying a continuous value but giving a faulty result (e.g. too high value).

The remaining problem is now to define which failure modes are actually relevant. One could picture a system as having a nominal mode where everything works as supposed along with several failure modes. These failure modes can either be relevant for the modeled system or not. The failure modes not relevant for the system can be seen as general failures and are typically failures that are not connected to a specific state or transition. Such failures can be e.g. a traffic light switches from a red light to a blue light, which would be impossible. A relevant failure mode would instead have been a traffic light not switching to red when supposed. More examples of relevant failure modes are given in Chapter 8.

When the relevant failure modes have been identified, each failure mode then has to be assessed as a hazard or not. In ISO 26262 this assessment is recommended, as stated before, to be done by e.g. brainstorming, checklists, quality history and field studies, with the goal to determine if a failure mode have the potential to lead to an accident [19]. For simplicity, examples used in this thesis have been assessed through brainstorming. An example is shown below to clarify the method.

Example: Failure Mode
Assume a simple system of a motion sensitive lamp. When someone walks into the room, the light should be switched on and turned off when the person has left the room. A state machine over this system is shown in Figure 5.1.2 where off means that the lamp is off and similarly for the on state. By using the state machine as an inspiration, the associated failure modes can be identified as given in Table 5.1.1, which also shows the result from the assessments of each failure mode, determining if it is a hazard or not. In this example two of the four failure modes are believed to be potential sources of accidents and hence, are assessed as hazards. Notation used in the state machine are *timed automata* which might ease model checking and is used in e.g. the UPPAAL tool used for modeling, simulation and verification of real-time systems [27].

If the states would have involved displaying of a continuous or discrete value, the state can fail even if no transition between states are made, i.e. going from displaying a correct value to a faulty value. In that case extra failure modes have to be considered as shown in Section 8.1.1.

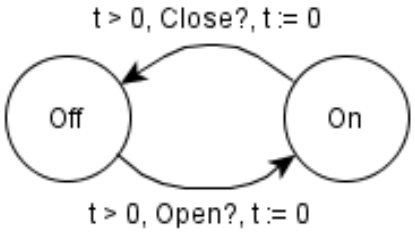


Figure 5.1.2: Motion lamp example, initially the lamp is off. The lamp should be turned on when someone enters the room and turned off when the person has left.

Table 5.1.1: Failure modes in the motion lamp example.

Failure Mode	**Description**	**Transition**	**Hazard**
1	Switched on when it should not	Off -> On	No
2	Not Switched on when it should	Off -> On	Yes
3	Switched off when it should not	On -> Off	Yes
4	Not Switched off when it should	On -> Off	No

5.2 Failure on Demand or in Continuous Operation

When discussing failure modes and hazards, questions about their nature arise. Will a specific failure mode be possible to repair and how are the failure probabilities measured? The first part of the question has one general answer in the scope of this thesis; no, the failure modes will not be modeled as repairable. In real life the failure modes might be possible to repair, but in the automotive case this will demand some kind of service, which will only be made after the failure mode has been discovered, hence a possible accident has already happened. After discovering the failure mode and avoiding accident, the driver is aware of the problem, which will distinctly lower the probability of an accident. However, e.g. some sensors are often modeled as repairable, since sensor failures often occurs only temporary [28].

When it comes to the question about the nature of the measured failure rates, those can be considered in two different ways. Either they are measured as *on demand* or in *continuous operation*, i.e. P(Failure Mode|Demand) or P(Failure Mode|1 Hour Driving) [19] [20]. Probability of a failure mode on demand are translated as the unavailability of a system to perform a function when a demand occurs and probability of failure per hour translates as unreliability over a given period of time (i.e. here 1 hour) [20] [3]. In IEC 61508 both on demand and continuous failure rates are covered. They also differ between two types of on demand failure rates; low demand and high demand. Low demand means that demand occurs less than once a year and high demand means that demand occurs more often than once a year.

In ISO 26262 no discussion about on demand failure modes are present. The probability requirements seen in Table 5.3.1c are the only ones covered. The reason for choosing this in ISO 26262 could be to create a consistency throughout the standard since the classifications made on probability of exposure are also considered as per hour of operation. Note also that failure rates and probability of exposure in ISO 26262 are connected to the operation of a

single average vehicle in the specific market being considered [19]. If this was not the case for e.g. the probability of exposure, this would get a very high value for every operational situation since it is very likely that some vehicle somewhere usually drive in that particular operational situation, e.g. driving on a slippery road in northern Canada. Comparison between different alternatives of defining probability of failure are shown in Table 5.2.1, where P(Condition Occur|Average Drive, 1 Hour) is the one used in ISO 26262 and has been, and will continue to be used in this thesis due to the broad usage in the standard. The average drive parameter needs to be added due to the probability of any condition to occur during one hour drive for someone somewhere is very high. Furthermore, if this definition were not to be used, a failure rate investigation needs to be done for every single market. By using an average drive, the same failure rate can be used in most markets.

Table 5.2.1: Overview over ways to assess e.g. failure rates. [1]Not given in IEC 61508.

Probabilistic failure rates	**Range of Requirement in IEC 61508**
P(Condition Occur|Low Demand)	$[10^{-5}, 10^{-1}]$
P(Condition Occur|High Demand)	$[10^{-9}, 10^{-5}]$[1]
P(Condition Occur|1 Hour Driving)	$[0.8, 0.95]$
P(Condition Occur|Average Drive, 1 Hour Driving)	$[10^{-9}, 10^{-5}]$

5.3 Top-Down vs. Bottom-Up

As for an actual implementation of a model, it can be constructed using two different views; top-down or bottom-up. A top-down view breaks down the system from top to bottom and gradually expands to new levels of complexity, whereas a bottom-up view starts from the bottom level working its way upwards by linking different subsystems together. In the scope of this thesis a top-down view will be associated to required hazard probabilities, while a bottom-up view will be associated to measured failure probabilities of every individual hardware or software component. A top-down view is used in ISO 26262 [19].

5.3.1 Measured Probabilities of Lower Level Failures - Bottom-Up

Based on Güdemann and Ortmeier's [18] discussion about risk analysis, the failure rates or probability of a failure mode of a subsystem can be used to calculate the probability of every hazard or failure mode of its top level system. If we consider every failure mode of a subsystem as a fault in the top level system, these faults might lead to a hazard. The main problem here would be to allocate fault or faults to the correct hazard.

5.3.2 Required Probabilities of Lower Level Failures - Top-Down

In a top-down view, probabilities will no longer be based on the estimated failure probabilities of the individual subsystems. Instead, every hazard will have to be given a required failure probability. One way to do this is given in ISO 26262. While having conducted a hazard analysis, every hazardous event has been given an ASIL. By using the highest ASIL associated

with a single hazard, this level will function as a hazard failure probability requirement. In this case the ASIL:s are translated into probabilities as in Table 5.3.1c [19, Part 5, Table 6]. This translation into probabilities is only made for levels B-D in ISO 26262. However, other earlier standards do map probabilities to every safety integrity level (SIL). Examples are shown in Table 5.3.1a and Table 5.3.1b, where the levels are given a number rather than a letter [2, Table 1] [20, Part 1, Table 3]. Note that both level B and C in ISO 26262 seem to correspond to level 2 in IEC 61508. By reading the standards, ASIL C in ISO 26262 do correspond to SIL 2 development requirements and SIL 3 verification requirements in IEC 61508, hence ASIL C is a combination of SIL 2 and 3. Furthermore, level 4 in IEC 61508 does not have any corresponding level in neither ISO 26262 nor the older standard IEC 65A [19] [20].

Table 5.3.1: Different ways of translating safety integrity levels to probabilities.

(a) Probabilistic requirement mapped to SIL in IEC 65A.

Level	**Random Failure target value per hour**
4	$< 10^{-8}$
3	$< 10^{-7}$
2	$< 10^{-6}$
1	$< 10^{-5}$
0	$< 10^{-4}$

(b) Probabilistic requirement mapped to SIL in IEC 61508.

Level	**Random Failure target value per hour**
4	$< 10^{-9}$
3	$< 10^{-8}$
2	$< 10^{-7}$
1	$< 10^{-6}$

(c) Probabilistic requirement mapped to ASIL in ISO 26262.

Level	**Random Failure target value per hour**
D	$< 10^{-8}$
C	$< 10^{-7}$
B	$< 10^{-7}$

In accordance with the bottom-up case, hazards need to be allocated to some subsystem. When allocating a hazard to a subsystem, the subsystem inherits the failure rate requirement given to the hazard. This way every subsystem gets a failure rate requirement, which then is allocated to lower level subsystems. It should also be noted that ISO 26262 allows different requirements to be assigned to different hazardous events, hence if one part of a subsystem clearly can be allocated to that hazardous event, only this part will have the higher requirement, e.g. if only situations where driving on a freeway are classified as ASIL D, a part in a subsystem making it impossible to drive on a freeway would have the highest requirement while other parts would have a lower requirement, even if the associated hazard itself is connected to many different subsystems [19]. The top-down view will be used in the further reasoning in this thesis.

5.4 Top-Down Assessment of Hazard Probability

The Bayesian network introduced in Section 3.3 assumes probabilities assigned to each hazard. Possible ways to assign those probabilities will be discussed in this section. Typically safety assessments or ways to assign these probabilities are divided between qualitative and quantitative analysis. Qualitative analysis are exemplified by expert analysis, failure modes and effects analysis (FMEA) along with hazard and operability analysis and are useful in early system design work. However, to be able to thoroughly model a system, a quantitative method is to prefer since all those qualitative methods have the disadvantage that, to a great extent, rely on expertise and skill of individual engineers [18]. More specifically, qualitative analysis often functions as a basis for a quantitative analysis, who classify safety using mathematical models [29]. Quantitative analysis use model-based techniques for probability assignment, which allow systems to be analyzed in a more formal way making it possible to find possible failures and faults earlier [18]. In addition, a quantitative analysis will have the advantage that the total probability of failure of a complex system can be divided into partial failure probabilities of its subsystems. These partial probabilities are usually easier to assign and by fault propagation the total system probability of failure is able to be determined [3].

Amongst quantitative analysis methods Markov models and Fault Tree Analysis are the most common [29]. Fault Tree Analysis breaks down the hazards into the lowest level failure modes connected through logical gates. Common problems are difficulties in relating the different failure modes in a correct logical combination while not being too pessimistic about the safety and that a fault tree easily gets very big and unmanageable. Markov models on the other hand, possess increased modeling power/possibilities compared to, amongst others, Fault Tree Analysis, although with the disadvantage of relatively high analysis complexity [5] [29].

Markov models, or Markov Chains, are divided into Continuous Time Markov Chain (CTMC) and Discrete Time Markov Chain (DTMC). When assessing safety, discrete time models have historically most widely been used [25] [14]. However, current research, e.g. the CESAR project, which aims to develop a reference technology platform to be used in safety related embedded systems, works on quantitative safety assessment techniques including both Markov chains (CTMC and DTMC) and fault trees, which are to be elaborated due to current problems in relating failure probabilities and model-based approaches to each other in a satisfying way [30]. Furthermore, in addition to Markov models and Fault Trees, recent advances more and more seem to use models like continuous-time semi-Markov, Petri nets and stochastic activity networks [31]. An overview over different techniques currently used in research for quantitative safety analysis is shown in Table 5.4.1.

Table 5.4.1: Different probabilistic methods used for quantitative safety analysis.

Probabilistic Methods for Quantitative Safety Analysis
Deductive Cause Consequence Analysis
Reliability Graphs
Series-Parallel Acyclic Directed Graphs
Product-Form Queuing Networks
Event Tree Analysis
Fault Tree Analysis
Fault Trees with Repeated Events
Block Diagrams
Petri Networks
Generalized Stochastic Petri Nets
Deterministic Stochastic Petri Nets
Discrete-Time Markov Models
Continuous-Time Markov Models
Continuous-Time semi-Markov Models
Multi-Phase Markov Chain
Markov Random Field
Markov Regenerative Processes

Chapter 6

Modeling Systems Incorporating Failures

One of the main things when it comes to actually using techniques for safety assessments, is to make it available to the user. This is done by various kinds of models and implementing them using some software tool. At Scania, quantitative analysis in the context of functional safety is not currently used and the following modeling discussion will consider models most suited for Scania.

6.1 Handling Parallel Systems

It is possible that one system handles several different functions, e.g. a fuel level system in a car, which displays a continuous value and triggers a warning when the value reaches a threshold. These parallel systems can either be combined to form a single state machine or modeled by many smaller systems. While this problem is sparsely covered in the literature, singling out the parallel systems would greatly simplify the modeling process, which obviously is preferable. However, the parallel systems will need to be joined eventually, when calculating probabilities, in systems where multiple point failures are important. Here systems are modeled both using parallel state machines, to create a basic system understanding, and a joined state machine when modeling failure modes. An example of this is shown in Chapter 8. Figure 6.1.1 clarifies the meaning of multiple point failures. Note that in large systems *combinational explosion* should be considered, but will not be discussed in this thesis.

6.2 Expanding State Machines with Failure Modes

To be able to estimate and model probability of hazards, those hazards have to expand the previously discussed state machines, describing the functionality of a system. One way of doing this is to use hierarchical state machines and introduce every hazard as a state. Then the state machine describing a system has one state called nominal mode and several failure mode states, i.e. each hazard would be a relevant failure mode state. An example on how

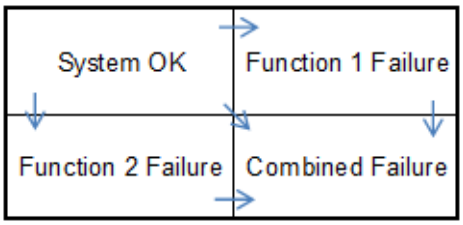


Figure 6.1.1: In systems with parallel functions failures can relate either to only one function or combined. The figure shows an example with two parallel functions.

this is modeled using an actual function at Scania is given in Chapter 8. Figure 6.2.1 shows a simple example where a nominal mode and its failure modes are modeled. Recall that this is a hierarchical state machine, hence every state consists of its own functional states. Here the nominal mode might consist of the state machine in Figure 5.1.2.

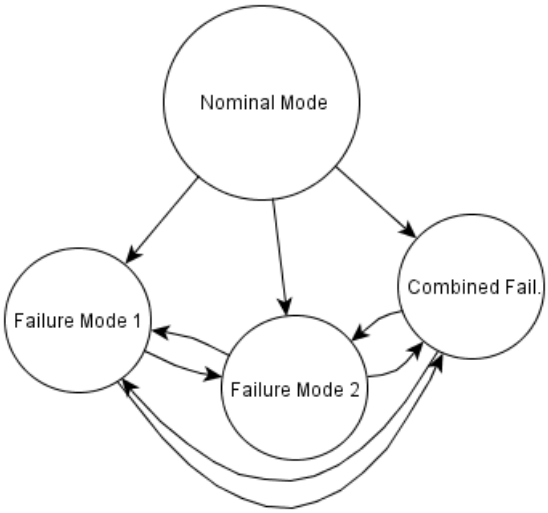


Figure 6.2.1: A model illustrating nominal mode and failure modes of a system. Nominal modes indicate that the system works as supposed, whereas each failure mode indicates some faulty behavior of the system.

6.3 Using Markov Models as a Basis for Hazard Probability Assessment

In Table 5.4.1 several alternatives to quantitative safety analysis techniques are shown. Among these, Markov models accommodates an increased interest in resent research. Rouvroye and van den Bliek [29] concluded in a comparison among safety assessment techniques that Markov models possess the most powerful modeling possibilities, although the Markov models were also shown to be the most complex models in a comparison with mainly Fault Tree Analysis. Later e.g. Felger and Frey [14], Güdemann and Ortmeier [18] and Hildebrandt [32] have further

argued for the usage of Markov models in safety related assessments. Furthermore, Börcsök [3] pointed out the possibility of simple graphical representation and its close bond to traditional state machines, primarily adding transition probabilities. However, disregarding Fault Tree Analysis, recent research has considered Petri nets as a possible modeling approach. Petri nets use tokens and allow for many states to be active simultaneously but will not be used in this thesis.

Remaining is then different kinds of Markov models. There are mainly Continuous Time Markov models (CTMC) and Discrete Time Markov models (DTMC), which implements time in differnt ways. Models can also differ in the distribution of the transition probability. Here semi-Markov models, Multi-Phase Markov models and Markov Random Field extend the regular CTMC or DTMC. See Section 2.2.3 for a review of the different models.

In this thesis CTMC:s will be covered to show possibilities regarding hazard probably assessment, since accident processes are continuous risks, and failures do occur continuously. DTMC:s have been most widely used in the past but since failure rates are very small, combined with time spans of many years, constant time steps might be too large and creates uncertainty if an accurate time step has been used or not. This problem is solved by using CTMC:s, which by dynamic step size control gives a fast and reliable result [14]. Hence, using continuous time makes it possible to argue about whether a state has been reached within a given time, which is exactly what we are aiming for [18].

As a basis for CTMC:s we need state machines over nominal modes and their corresponding failure modes. These state machines need to be defined for each level, spanning from basic top level functions to hardware and software level functions, exemplified in Figure 6.2.1. The trick is now to first assign appropriate probabilities based on failure rates or ASIL:s and then merge all state machines correctly making it possible to derive a transition matrix of the entire system. This matrix will then include the probability of being in a hazardous state, hence the probability of a hazard. This is implemented using an example in Chapter 8.

6.3.1 Assigning Transition Probabilities and Deriving Transition Matrices

In a top-down view, which is being handled in this thesis, failure rates are assumed to have been assigned to a top level model based on ASIL:s assessed in a hazard analysis described in ISO 26262. Now, the probability of being in a hazard at a certain time is still unknown and needs to be calculated. Here Markov models come into play. By using constant failure rates, which is recommended in reliability contexts because of its memory less properties, these probabilities are calculated [5] [28].

In Figure 6.3.1 a state machine, similar to the one in Figure 6.2.1, is extended with failure rates, λ_{ij} at every transition.

With the failure rates defined along with appropriate time spans, a transition matrix of the

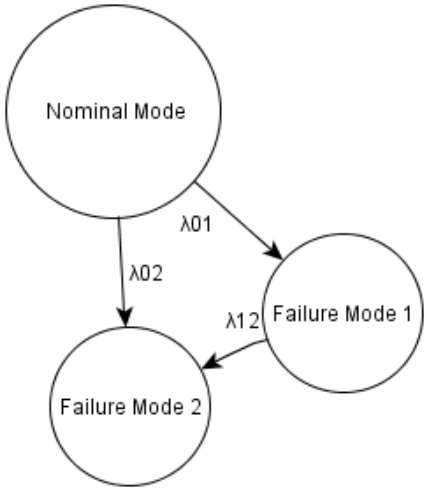


Figure 6.3.1: A Markov model illustrating nominal mode and failure modes of a system and its associated failure rates. Note that this is not the same example as in Figure 6.2.1.

CTMC derives to [3] [5]

$$P = \begin{pmatrix} p_{0,0} & p_{0,1} & p_{0,2} \\ p_{1,0} & p_{1,1} & p_{1,2} \\ p_{2,0} & p_{2,1} & p_{2,2} \end{pmatrix} = \begin{pmatrix} 1 - \sum_{\substack{k=0 \\ k \neq 0}}^{2} \lambda_{0,k} dt & \lambda_{0,1} dt & \lambda_{0,2} dt \\ \lambda_{1,0} dt & 1 - \sum_{\substack{k=0 \\ k \neq 1}}^{2} \lambda_{1,k} dt & \lambda_{1,2} dt \\ \lambda_{2,0} dt & \lambda_{2,1} dt & 1 - \sum_{\substack{k=0 \\ k \neq 2}}^{2} \lambda_{2,k} dt \end{pmatrix}$$
$$= \begin{pmatrix} 1 - \sum_{k=0}^{2} \lambda_{0,k} dt & \lambda_{0,1} dt & \lambda_{0,2} dt \\ 0 & 1 - \sum_{\substack{k=0 \\ k \neq 1}}^{2} \lambda_{1,k} dt & \lambda_{1,2} dt \\ 0 & 0 & 1 \end{pmatrix} \tag{6.3.1}$$

where p_{ij} is the probability of a transition from i to j within time dt.

Given this transition matrix, which is upper triangular due to nonrepairable failures, the probability of being in a failure mode is calculated. By defining

$$Q_1(t + dt) = P(\text{Failure mode 1 at } t + dt) = \lambda_{0,1} dt(1 - Q_1(t)) + (1 - \lambda_{1,2} dt) Q_1(t) \tag{6.3.2}$$

which gives the differential equation

$$\frac{dQ_1(t)}{dt} = -(\lambda_{0,1} + \lambda_{1,2}) Q_1(t) + \lambda_{0,1} \tag{6.3.3}$$

This equation, with initial condition $Q_{nom}(0) = 1$ and $Q_1(0) = 0$, assuming exponential distribution of failure rates, have the solution

$$Q_1(t) = \frac{\lambda_{0,1}}{\lambda_{0,1} + \lambda_{1,2}} (1 - e^{-(\lambda_{0,1} + \lambda_{1,2})t}) \tag{6.3.4}$$

In a more general way, $Q_i(t)$ can be written as

$$Q_i(t) = \frac{\lambda_{in}}{\lambda_{in} + \lambda_{out}}(1 - e^{-(\lambda_{in}+\lambda_{out})t}) \tag{6.3.5}$$

This holds for every state and makes it possible to derive probabilities of being in a particular state. If a state have several inputs, as failure mode 2 in Figure 6.3.1, where one input is caused by direct failure from the nominal mode, and the other as indirect from failure mode 1, extra calculations are needed. When calculating Q(t) for these kinds of states, the notion of secondary failures is used. The failure rate associated with such a state is then defined as $\lambda_{in} = \lambda_{dir} + \lambda_{indir}$, where $\lambda_{in} = \lambda_{0,1}$ and $\lambda_{indir} = \lambda_{0,1}\lambda_{1,2}$ for the example illustrated in Figure 6.3.1 [5].

Furthermore, t can be seen as the operating time during which a failure might occur. The average driving time, the life time of a vehicle or one hour of driving are possible values of interest here. However, since we are considering unrepairable failures, a time span of the entire life time of a vehicle leads to higher probability of failure than it should, due to the lack of including maintenance in the model. Since we are dealing with unrepairable failures with constant failure rates, and are interested in the probability of death per hour, t = 1 hour will be used in calculations.

Example: Failure Mode Probability Calculation
Assume the configuration in Figure 6.3.1 with $\lambda_{0,1} = 10^{-7}h^{-1}$, $\lambda_{0,2} = 10^{-6}h^{-1}$ and $\lambda_{1,2} = 10^{-6}h^{-1}$. Then

$$P = \begin{pmatrix} 10^{-0}dt & 10^{-7}dt & 10^{-6}dt \\ 0 & 10^{-0}dt & 10^{-6}dt \\ 0 & 0 & 1 \end{pmatrix} \tag{6.3.6}$$

yielding

$$Q_1(t) = \frac{10^{-7}}{10^{-7} + 10^{-6}}(1 - e^{-(10^{-7}+10^{-6})t}) \tag{6.3.7}$$

$$Q_2(t) = 1 - e^{-(10^{-6}+10^{-7}10^{-6})t} \tag{6.3.8}$$

With t=1 hour, the probability of being in failure mode 1 and 2 after 1 hour derives to

$$Q_1(1) = 9.999945 \cdot 10^{-7} \tag{6.3.9}$$

$$Q_2(1) = 9.99995 \cdot 10^{-6} \tag{6.3.10}$$

These values are about one order of magnitude larger than their respective failure rates.

6.3.2 Assigning Probabilities Between Levels of Abstraction

Knitting levels together, by assigning transition failure rates on a higher level (as in Figure 6.3.1), to transition failure rates on a lower level, is needed when deciding lower level probability of failure modes. In a bottom-up view, the lowest level failure rates are gathered

through statistics. These failure rates are then tied together with a higher level, in order to assign failure rates. In a top-down view, required failure rates function as a base for requirements on a lower level. One possible way of modeling this process is to use a Bayesian network. This Bayesian network has one "layer" for each level of a system. If using a top-down view, a Bayesian network of a system with only two levels, where variable L_{ij} are the modes of level i (nominal mode, failure mode 1, failure mode 2 etc.) and component j, is exemplified in Figure 6.3.2, with corresponding CPD in Table 6.3.1.

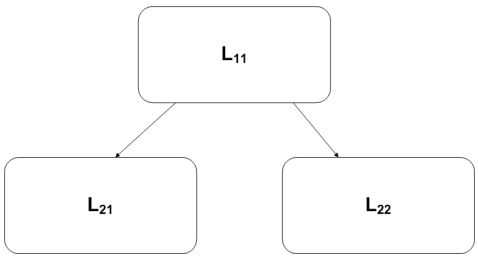


Figure 6.3.2: Bayesian network illustrating the connection of required failure rates between levels.

Table 6.3.1: CPD over L_{22}, illustrating which failure rates corresponds to each other.

L_{22}	Nom_2	$Fail1_2$	$Fail2_2$	$Fail3_2$
Nom	1	0	0	0
Fail1	0	p_{11}	p_{12}	p_{13}
Fail2	0	p_{21}	p_{22}	p_{23}

This way of modeling is one way of illustrating the connections between levels. The problem with this model is to correctly map different levels to each other. Assuming a correct mapping, calculations of failure mode probability can be made. However, this thesis will not cover any further discussions on lower level connections or probability calculations.

Chapter 7

Implementation in Modelica

So far, models have shown how functional safety can be assessed quantitatively, using Bayesian networks combined with CTMCs. If this approach is to be used on a daily basis in organizations such as Scania, the method needs to be easily implemented in a modeling language. By modeling in such a language, and thereby generating calculations, the working process and understanding of a system could become more efficient and comprehensive. Ideally, a state machine of a system is modeled followed by classifications made in a hazard analysis. These are then given as input, yielding the loss of the system as output. However, as stated, this thesis specifically focuses on the Modelica language and what possibilities it has to offer.

7.1 Why the Modelica Language?

The Modelica Language is an object-oriented programming language aiming to be a universal language for high level computational modeling and simulation. It can be applied in a wide range of areas covering continuous, discrete and hybrid systems, along with many extension possibilities. Modelica is commonly used in areas such as hydraulic, control, mechanical, electronic and electrical [33]. Common platforms using the Modelica language are Dymola and OMEdit (provided by Dassault Systeme and Open Source Modelica Consortium respectively). Alternatives are mainly UML, MapleSim and Simulink/MatLab, where Modelica have the advantage of being able to model non-causality, that systems are unequivocally determined (i.e. closed systems, not possible to change internal parameters from the outside), that it is equation controlled, of being possible to implement continuous-time-based models, of having large extension possibilities and of being developed as an open standard. Simulink or MapleSim might be usable to implement the developed method as well as Modelica. However, Modelica has been used in many applications previously at Scania, providing competence and experience mainly using Dymola as the platform. As for the language itself, Modelica is, as stated before, an object-oriented language, which is controlled by equations. The most common classes are models, types, blocks and connectors, where models are the most general class and can be anything except connectors. Connectors define connections between classes and are not allowed to include equations.

The main part discussed in this thesis is the possibility to use the CTMCs in Modelica. Hence,

in the following sections, since modeling physical systems is the main focus of Modelica, discussions about state machines and CTMCs and their relation to physical models, are presented. Possibilities for modeling Bayesian networks and performing calculations are also discussed briefly.

7.2 General System Description in Modelica

One extension available in Modelica is the Modelica_StateGraph2 library. This library provides components for deterministic hierarchical state diagrams. This makes it possible for easy modeling of discrete events and reactive and hybrid systems [34]. Specifically, this library adds classes such as step (i.e. state), transitions and parallel (used to make hierarchical states), together with appropriate connectors. An example of a state graph is shown in Figure 7.2.1. In the example, the steps are illustrated by the smaller ellipses, where step1 serves as the initial state. Transitions are the classes named T1-T7, and are here given the condition true or a time based condition. The bigger ellipse, including both steps and transition, is the parallel class. This is a hierarchical state, where the suspend and resume ports defines whether or not the step is active, while the entry and exit ports are used if inputs and outputs are present.

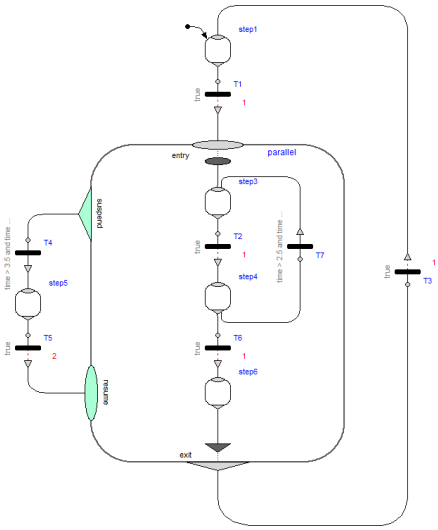


Figure 7.2.1: Example of a system modeled with the Modelica_StateGraph2 library [34].

Fritzon [33] describes how similar classes are implemented manually, by defining a state class, a transition class and a connector class. This way of defining classes used to implement state graphs is also used by Felger and Frey [14] in their CTMC implementation example. However, since the Modelica_StateGraph2 provides similar classes, this library is used here in order to implement system descriptions and CTMCs.

7.3 Continuous Time Markov Chain in Modelica

By using the Modelica_StateGraph2 library, we want to model the CTMCs derived for a system. These CTMCs are based on hierarchical state machines, hence the parallel class is used. These hierarchical states are used to model a CTMC with nominal mode and failure modes. Inside each of these states, the corresponding system state machines are modeled. In the nominal state, the internal states are the ones present when the system works as supposed, whereas the internal states of a failure mode corresponds to the possible functionality when the failure is present.

We want these CTMCs to depend on actual physical properties of a system, hence the state transitions are connected to physical failures. These physical failures occur, based on failure rates, and trigger a transition. This means that modeling both the physical properties of a system and the more abstract functionality of a system in a joint model is necessary. This way of creating a system model will also provide a better understanding of a system.

However, Modelica does not currently provide complete support for CTMCs. This is a problem, so let us discuss the current possibilities and what functionality that needs to be developed. In Figure 7.3.1 a model created in the Dymola tool is shown. In this model the *System Layout* (1), *Connectors* (2), *State Transitions and Properties* (3) and *Graphics and Code* (4) are highlighted. These will be discuss individually, expanded with *Calculations* (5), not shown in the figure.

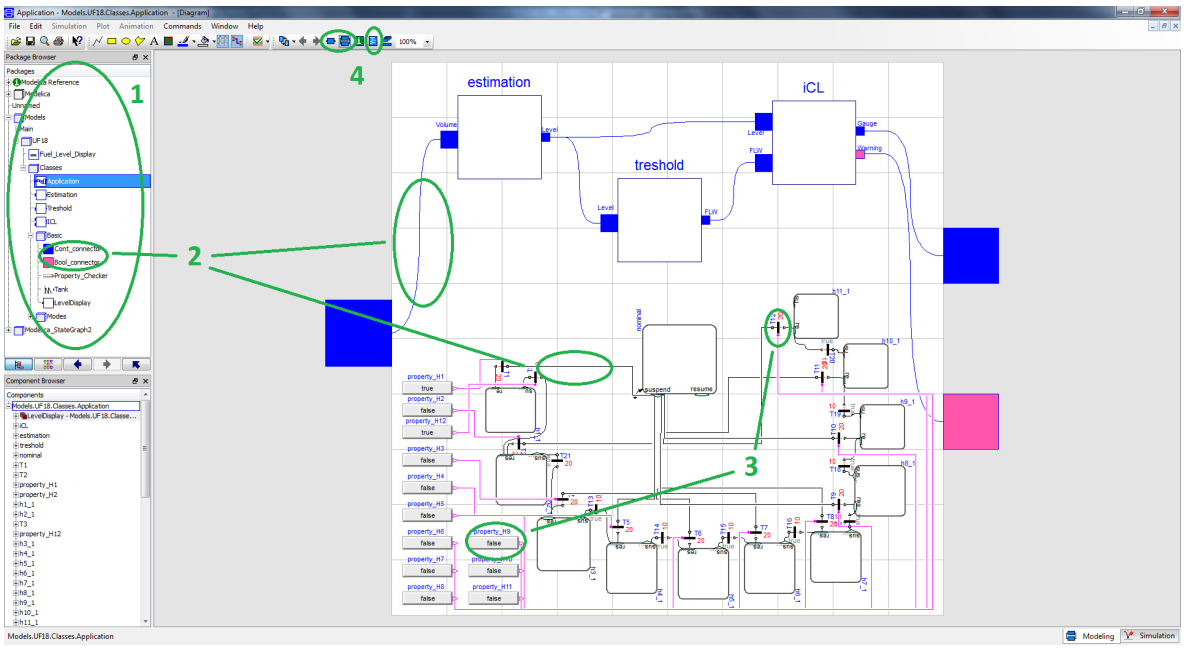


Figure 7.3.1: View of a model created in the Dymola tool.

1. System Layout

It is of great importance to get a clear and easily understood overview of a system. To get a suitable overview, package and class structures play an important role. By defining a package called models containing all systems, or a category of systems, in a company, several different systems are collected in one place. These systems are preferably defined in separated packages. This way, every system has a single package. Inside the package of a system, all its different components are defined. These components are the physical components of a system, at different levels. Every system also includes packages containing property and connector classes, such as hierarchical state machines. A closer look at the structure is shown in Figure 7.3.2.

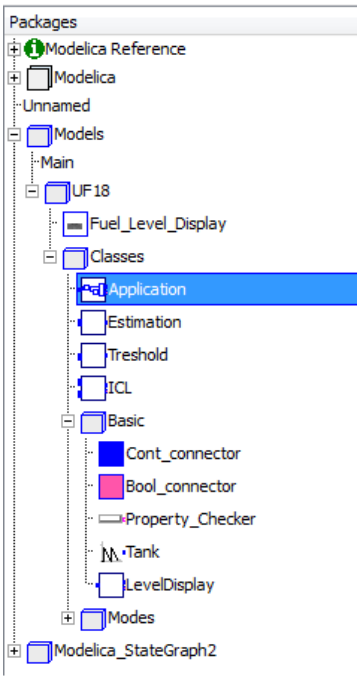


Figure 7.3.2: A closer look at the way to organize packages and models.

2. Connectors

To create models, different kinds of ports are used. In a physical model of a system, non-causal ports are the most commonly used ports in Modelica. These ports are easily defined using connector classes, and are either boolean or real valued. Now, since we aim to create a combined physical and abstract model, where the abstract (state machine) uses values defined in the physical part, the non-causal ports need to be used in combination with state machines. As it shows, the non-causal ports easily connect to ports in the state machine and no additional functionality are needed. However, graphical representation of these connections should be concidered further.

3. State Transitions and Properties

In the Modelica_StateGraph2 library, transitions between states are defined using a specific transition class. This class uses conditions to enable a transition. By setting a condition as true, it fires with or without an associated delay time. Setting a condition is made by giving a default value or connecting the transition to a boolean input.

In order to actually enable transitions between states, such as between nominal and failure modes or between functional on and off states, these transitions need to connect to some property. In the case of transitions in a function oriented state machine, this property can be anything spanning from a given time when a state is shifted, to a specific level of a physical entity at which a transition fires, such as the properties in Figure 7.3.3. Jardin et al. [26] discuss the implementation of these kinds of properties in a recent study. According to them, the best way to implement properties is to define a general property class, which then is used whenever needed. Using this way of implementing the properties, it is easy to get an overview of used properties in the graphical view of a system. Although, additional graphics to clarify which property connects to what physical value might be useful. Furthermore, when using hierarchical state machines, the function oriented states are modeled inside a parallel class object of the Modelica_StateGraph2 library. This cause problems concerning connections between modeling levels. To be able to use physical values on the properties inside parallel states, the parallel class needs to be revised with extra connectors.

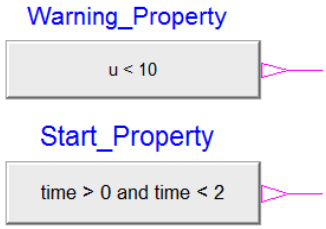


Figure 7.3.3: Properties defining a threshold level and a time interval.

Considering transitions between nominal and failure modes of a system, these transitions are meant to fire based on failure rates. This introduces the need to model uncertainties in Modelica. Unfortunately, uncertainties are currently not supported to accurately illustrate processes given in Markov models. This means that uncertainties cannot currently be located to individual transitions in order to trigger them. However, a project called OPENPROD - Open Model-Driven Whole-Product Development and Simulation Environment, has implemented uncertainties to some extent. Here the uncertainties have been connected to physical quantities to get a more accurate simulation of the behavior of a model [35]. Certain elements used in a model might be useful in the Markov case, by e.g. defining a distribution class and extending the transition class with this distribution allowing probabilistic based transition. The remaining problem will then be how to incorporate the lifetime of a function in a simulation.

Schallert [36] presents an alternative way of modeling, where every specific fault is modeled in the physical system description. Problems still arise concerning implementation of uncertainties, but if a comprehensive physical system description does exist, a fault class with boolean

connectors could preferable be connected to a state transition.

4. Graphics and Code

Modelica provides a graphical interface for easy implementation a system of some sort. This graphical model is directly associated to source code. These different views provide various advantages and disadvantages. The advantages include simple and manageable implementations in the graphical case and powerful configuration possibilities in when using the code view. Disadvantages occur mainly when switching between modes. If a connector is changed in the graphical mode, e.g. connector 1 between component 1 and 2, is replaced by connector 2 between component 1 and 3, the new connection is added automatically in the bottom of the code. This result in difficulties to get a clear overview of the code, hence a better automated organization of the code would be helpful when designing models using both graphical and code based views. Furthermore, it is hard to create nice looking models using the graphical view. Especially the lines connecting different objects are hard to organize in a god way. Example code is shown in Figure 7.3.4.

```
model ICL
  Models.UF18.Classes.Basic.Cont_connector
                  flw "Continuous Connector"
    a;
  Models.UF18.Classes.Basic.Cont_connector
                  level "Continuous Connector"
    a;
  Models.UF18.Classes.Basic.Cont_connector
                  gauge "Continuous Connector"
    a;
  Models.UF18.Classes.Basic.Bool_connector
                  warning "Boolean Connector"
    a;
  Modes.SecondLevel.ICL.Nominal nominal(
        initialStep=true,
        use_inPort=false,
        use_outPort=false,
        FLW_Checker(y=flw < 0.5),
        Level_Checker(y=level > 0),
        nSuspend=2) a;
  Modes.SecondLevel.ICL.Fail1 fail1_1(
        use_inPort=false,
        use_outPort=false,
        nResume=1) a;
  Modelica_StateGraph2.Transition T1(condition=false)
        a;
  Modes.SecondLevel.ICL.Fail2 fail2_1(
        use_inPort=false,
        use_outPort=false,
        nResume=1) a;
  Modelica_StateGraph2.Transition T2(condition=false)
        a;
equation
 warning = (if (flw > 0.5) then true else false);
 gauge = level;

      connect(T1.outPort, fail1_1.resume[1]) a;
      connect(T1.inPort, nominal.suspend[1]) a;
      connect(nominal.suspend[2], T2.inPort) a;
      connect(T2.outPort, fail2_1.resume[1]) a;
 a;
end ICL;
```

Figure 7.3.4: Code example illustrating parameter definitions, equations and connectors.

5. Calculations

In the present version of Modelica no calculations, needed for transition matrix generation or probabilities of being in a state, are supported. Possible ways to solve these problems are extending Modelica with the functionality or to export data, generated by Modelica, into a more powerful tool to perform calculations.

7.4 Bayesian Networks in Modelica

To make the implementation of the method in Modelica complete, the Bayesian networks discussed in previous sections need to be implemented and connected to the CTMCs. However, Modelica does currently not provide a library for Bayesian networks. Additional support for the graphical implementation of Bayesian networks should not be too hard to develop. Although, generating CPDs and resulting expectation values is a bigger problem. Here Modelica needs to be connected to some software, such as MatLab, to make these calculations.

Chapter 8

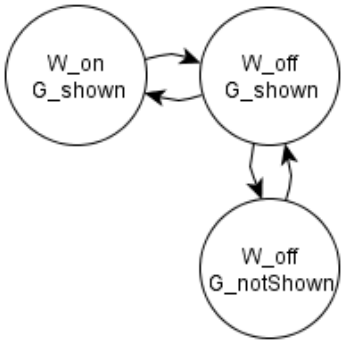
Method Example: Fuel Level Display

One basic system, that is present in most vehicles, is a fuel level display system shown in Figure 8.0.1. This system typically handles the basic functionality concerning calculating and visually displaying the fuel level to the driver, and to warn the driver when the fuel level reaches a specific threshold. At Scania, this Fuel Level Display system, or User Function 18 (UF18), is described in various documents and will be used as a first example to implement the model described in previous sections. The reason why this particular system was chosen is divided into three parts. Firstly, the fuel level display is present in most vehicles, making it easy for developers to relate to. Secondly, this is a relatively simple system, highlighting the assessment process, since the work might otherwise focus too much on the basic understanding of a system. Third, there are less confidentiality restrictions associated with the fuel level display compared to other available systems.

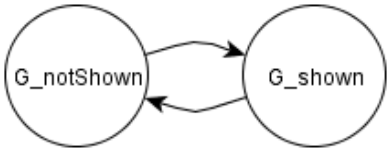
Figure 8.0.1: Fuel level display in a Scania truck.

8.1 Top Level System Description, or Item Definition

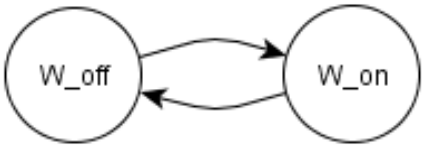
As discussed before, it is recommended to start with defining a basic state machine model of the system to be analyzed. Based on descriptions given by Scania a possible state machine over UF18 is shown in Figure 8.1.1 [6].



(a) State Machine over Fuel Level Display in nominal working mode.



(b) State Machine over Fuel Level Gauge.



(c) State Machine over Fuel Level Warning.

Figure 8.1.1: State Machines for UF18.

It is also useful to define the system boundaries and function explicitly. Figure 8.1.2 show the boundaries of the system, which are; the current physical fuel volume in the tank, the gauge shown to the driver and the warning lamp shown if the fuel level is low. The function of the system is to calculate the current fuel level based on the described inputs and to continuously display the current fuel level using a gauge available to the driver. The system functions also to display a warning to the driver when the fuel level reaches below a predefined threshold.

Figure 8.1.2: Fuel Level Display Boundaries.

8.1.1 Hazard Analysis

The hazard analysis is described in ISO 26262 and is conducted following its description. First, the hazards of the system are systematically identified. ISO 26262 recommends doing this by methods such as brainstorming. Here we base the identification upon state machines as described in the previous section. When the hazards are identified, relevant operational situations are found. Only the situations possibly able to give high ASIL classifications are covered, hence classifications are made on all hazardous events, after identifying the relevant operational situations and hazards. Classifications for severity, probability of exposure and controllability are made, giving an ASIL [19].

Note that this hazard analysis was made in collaboration with Patrik Sternudd and Caroline Erlandsson, both master's thesis workers at Scania in 2011[1]. Their respective approach to functional safety differ from the one discussed in this thesis, why a collaborative hazard analysis was thought to be a great way to get a more comprehensive view.

8.1.2 Hazard Identification

From the state machines in Figure 8.1.1, system failures are found from transitions. By assessing these failures, some of them are identified as hazards. All the possible single point failures are shown in Table 8.1.1, where the multiple point failures are every combination of the failures.

Table 8.1.1: Possible failures mapped to the Fuel Level Display system.

	Associated Transition	**Description**	**H**
Warning	$W_{off} \rightarrow W_{on}$	Fuel Level Warning displayed when it should not.	H1
	$W_{off} \rightarrow W_{on}$	Fuel Level Warning not displayed when it should.	H2
	$W_{on} \rightarrow W_{off}$	Fuel Level Warning turned off when it should not.	
	$W_{on} \rightarrow W_{off}$	Fuel Level Warning not turned off when it should.	
Gauge	$G_{notShown} \rightarrow G_{shown}$, $G_{Shown} \rightarrow G_{shown}$	Fuel Level Gauge display too high value.	H3
	$G_{notShown} \rightarrow G_{shown}$, $G_{Shown} \rightarrow G_{shown}$	Fuel Level Gauge display too low value.	H4
	$G_{notShown} \rightarrow G_{shown}$	Fuel Level Gauge not shown when it should.	
	$G_{shown} \rightarrow G_{notShown}$	Fuel Level Gauge shown when it should not.	
	$G_{shown} \rightarrow G_{notShown}$	Fuel Level turned off when it should not.	H5
	$G_{shown} \rightarrow G_{notShown}$	Fuel Level not turned off when it should.	

Table 8.1.1 also show if the failure is assigned as a hazard or not. This was done by looking at every single point failure and will have to be done for all possible combinations as well. All hazards associated to the fuel level display system are given in table 8.1.2. In Table 8.1.3 the relevant operational situations are given. These operational situations were found through brain storming, by first identifying the situations which were thought to give the highest ASIL. The result from this first session gave operational situations O1-O4, O7-O9 and O11. After conducting safety classifications (see Section 8.1.3) based on those operational situations, another session of finding operational situations was performed.

[1]Patrik Sternudd; "Unambiguous Requirements in Functional Safety: Dream or Reality?", Uppsala University
Caroline Erlandsson; "Better Requirenments in Functional Safety", Uppsala University

Table 8.1.2: Hazards related to the Fuel Level Display system.

Hazard	Description
H1	Fuel Level Warning displayed when it should not and Fuel Level Gauge indicates a correct fuel level.
H2	Fuel Level Warning not displayed when it should and Fuel Level Gauge indicates a correct fuel level.
H3	Fuel Level Gauge indicates higher fuel level than actual fuel level and Fuel Level Warning displayed when it should.
H4	Fuel Level Gauge indicates lower fuel level than actual fuel level and Fuel Level Warning displayed when it should.
H5	Fuel Level Gauge indicates no fuel level when it should and Fuel Level Warning displayed when it should.
H6	Fuel Level Warning displayed when it should not and Fuel Level Gauge indicates higher fuel level than actual fuel level.
H7	Fuel Level Warning displayed when it should not and Fuel Level Gauge indicates lower fuel level than actual fuel level.
H8	Fuel Level Warning not displayed when it should and Fuel Level Gauge indicates higher fuel level than actual fuel level.
H9	Fuel Level Warning not displayed when it should and Fuel Level Gauge indicates lower fuel level than actual fuel level.
H10	Fuel Level Warning not displayed when it should and Fuel Level Gauge indicates no fuel level when it should.
H11	Fuel Level Warning displayed when it should not and Fuel Level Gauge indicates no fuel level when it should.

8.1.3 Risk Assessment

By combining hazards with their relevant operational situations, risk assessments are made. In Table 8.1.4 the severity, probability of exposure, controllability and ASIL classifications are given. The process of deriving these classifications is made using brainstorming, in the following way:

- By looking at the hazards previously given, the worst possible consequence for each of them are found. In this case, these consequences were; controlled stop (H1, H4, H5, H7, H11), semi-uncontrolled stop (H2, H3, H6, H9, H10) and uncontrolled stop (H8).
- Map relevant operational situations to each of the hazards, e.g. operational situations including low fuel level in tank have no relevance for hazards where the warning is turned on when it should not.
- Assess every hazardous event (combination of hazard and operational situation) by severity, probability of exposure and controllability.
- Set ASIL for every hazardous event based on Table A.3.4.
- Add operational situations with high probability of exposure (e.g. O5 and O6 in Table 8.1.3) where appropriate, hence where the overall highest ASIL assigned to a hazard is believed to become higher. In this example operational situations incorporating good conditions were added, giving a higher probability of exposure, which in some cases resulting in a higher overall ASIL.
- Review all assignments and make sure it is complete and accurate.

Table 8.1.3: Operational situations relevant to the Fuel Level Display system. O13 includes all operational situations not covered in O1-O12 and are present for completeness reasons.

O	**Description**
O1	Impaired vision and slippery road while driving on a freeway with high traffic and not low actual fuel in tank.
O2	Impaired vision and slippery road while driving on a highway with high traffic and not low actual fuel in tank.
O3	Impaired vision and slippery road while driving in a city with high traffic and not low actual fuel in tank.
O4	Driving in confined spaces with impaired vision and not low actual fuel in tank.
O5	Good conditions while driving on a freeway with high traffic and not low actual fuel in tank.
O6	Good conditions while driving on a highway with high traffic and not low actual fuel in tank.
O7	Impaired vision and slippery road while driving on a freeway with high traffic and low actual fuel in tank.
08	Impaired vision and slippery road while driving on a highway with high traffic and low actual fuel in tank.
09	Impaired vision and slippery road while driving in a city with high traffic and low actual fuel in tank.
O10	Good conditions while driving on a freeway with high traffic and low actual fuel in tank.
O11	Driving in confined spaces with impaired vision and low actual fuel in tank.
O12	Good conditions while driving on a highway with high traffic and low actual fuel in tank.
O13	Operational situations not covered in O1-O12.

In Table 8.1.5 the highest ASIL assigned to each of the hazards are given, together with their required failure rates (from Table 5.3.1). Out of all hazards, H8 was assigned the highest ASIL (C), which means that associated subsystems have higher safety requirements than the others. Following H8, hazard H6 has ASIL B and subsequently hazards H1-H5, H7 and H9-H11 have the lower ASIL A.

Table 8.1.4: Result of the hazard analysis conducted on the fuel level display. All combinations containing O13 have been classified as QM since the worst cases are relevant in O1-O12. Definitions of S, E, C and ASIL are found in Appendix A.

	Op. Sit.	S	E	C	ASIL
H1	O1	3	2	1	QM
	O5	3	3	0	QM
	O2	3	3	1	A
	O6	3	4	0	QM
	O3	2	4	0	QM
	O4	3	2	1	QM
H2	O7	3	1	2	QM
	O10	3	2	1	QM
	O8	3	2	2	A
	O12	3	3	1	A
	O9	2	3	1	QM
	O11	3	1	2	QM
H3	O7	3	1	2	QM
	O10	3	2	1	QM
	O8	3	2	2	A
	O12	3	3	1	A
	O9	2	3	1	QM
	O11	3	1	2	QM
H4	O7	3	1	1	QM
	O10	3	2	0	QM
	O8	3	2	1	QM
	O12	3	3	1	A
	O9	2	3	0	QM
	O11	3	1	1	QM
H5	O7	3	1	1	QM
	O10	3	2	0	QM
	O8	3	2	1	QM
	O12	3	3	1	A
	O9	2	3	0	QM
	O11	3	1	1	QM
H6	O1	3	2	1	QM
	O5	3	3	1	A
	O2	3	3	1	A
	O6	3	4	1	B
	O3	2	4	1	A
	O4	3	2	1	QM

	Op. Sit.	S	E	C	ASIL
H7	O1	3	2	1	QM
	O5	3	3	0	QM
	O2	3	3	1	A
	O6	3	4	0	QM
	O3	2	4	0	QM
	O4	3	2	1	QM
H8	O7	3	1	3	A
	O10	3	2	2	A
	O8	3	2	3	B
	O12	3	3	3	C
	O9	2	3	2	A
	O11	3	1	3	A
H9	O7	3	1	2	QM
	O10	3	2	1	QM
	O8	3	2	2	A
	O12	3	3	1	A
	O9	2	3	1	QM
	O11	3	1	2	QM
H10	O1	3	1	2	QM
	O10	3	2	1	QM
	O2	3	2	2	A
	O12	3	3	1	A
	O3	2	3	1	QM
	O4	3	1	2	QM
H11	O1	3	2	1	QM
	O5	3	3	1	A
	O2	3	3	0	QM
	O6	3	4	0	QM
	O3	2	4	0	QM
	O4	3	2	0	QM

Table 8.1.5: Highest ASIL associated to each of the hazards. The requirements in ISO 26262 are used, except in teh case of ASIL A, where the corresponding requirement for SIL 1 in IEC 61508 is used, since no requirement on ASIL A is given in ISO 26262 (see Table 5.3.1).

Hazard	**Highest ASIL**	**Required Failure Rate** h^{-1}	Parameter
H1	A	$< 10^{-6}$	$\lambda_{0,1}$
H2	A	$< 10^{-6}$	$\lambda_{0,2}$
H3	A	$< 10^{-6}$	$\lambda_{0,3}$
H4	A	$< 10^{-6}$	$\lambda_{0,4}$
H5	A	$< 10^{-6}$	$\lambda_{0,5}$
H6	B	$< 10^{-7}$	$\lambda_{0,6}$
H7	A	$< 10^{-6}$	$\lambda_{0,7}$
H8	C	$< 10^{-7}$	$\lambda_{0,8}$
H9	A	$< 10^{-6}$	$\lambda_{0,9}$
H10	A	$< 10^{-6}$	$\lambda_{0,10}$
H11	A	$< 10^{-6}$	$\lambda_{0,11}$

8.2 Expanding the Model with Failure Modes

We now have a basic system description, in the form of a state machine, as well as relevant hazards with their associated risk assessments. To further reason about quantitative safety of the fuel level display, the state machine is expanded with hazards. This is illustrated in Figure 8.2.1, where all the possible transitions are incorporated, which is a hierarchical state machine. This means that the state machine in Figure 8.1.1a is incorporated in the nominal mode. Similarly, all hazards represent complete state machines on their own. Example top level state machines of hazards H2 and H5 are shown in Figure 8.2.2.

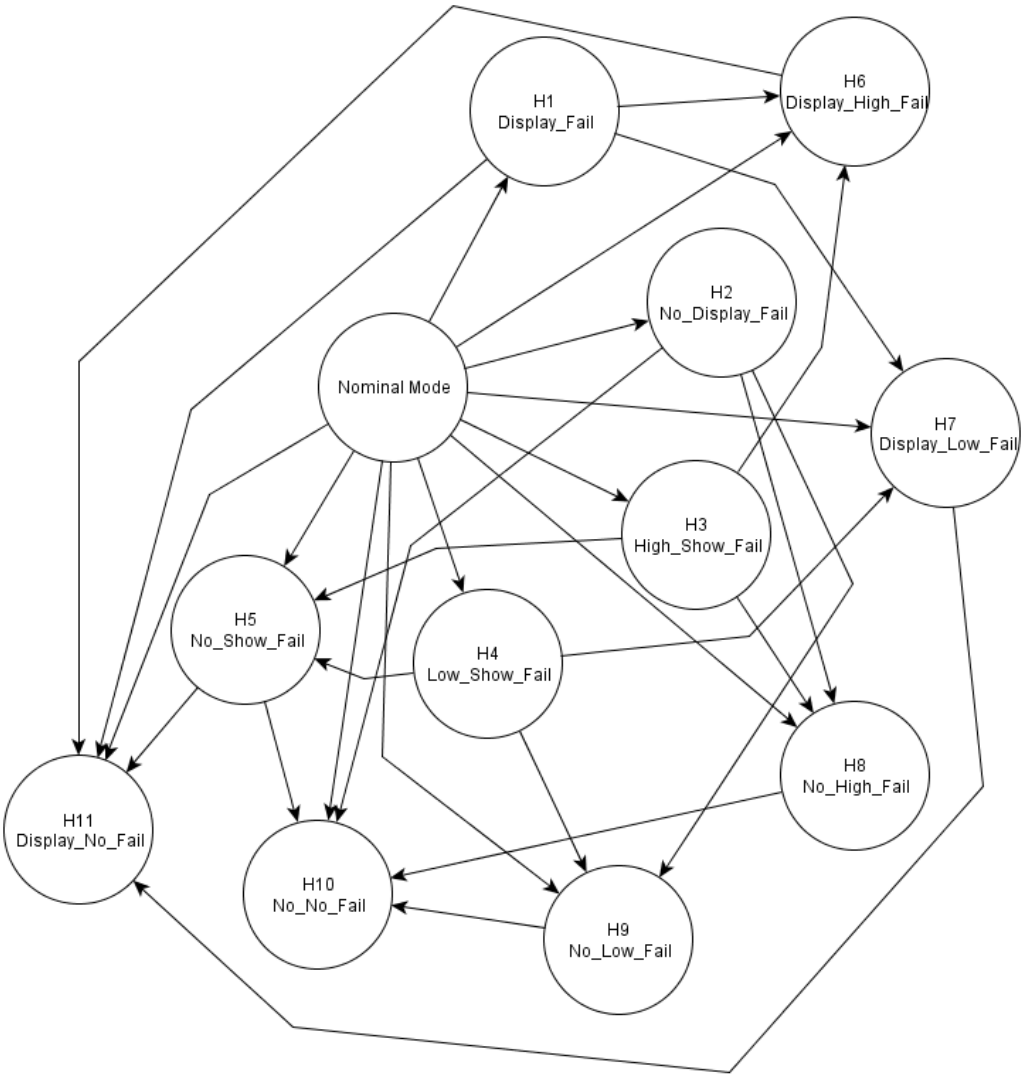
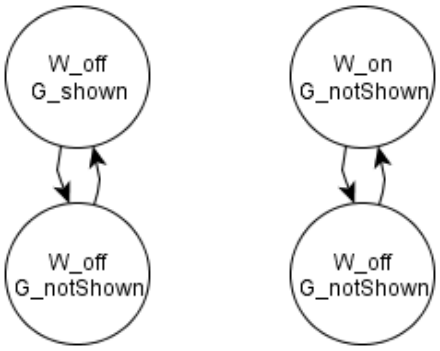


Figure 8.2.1: State machine with nominal mode and those failure modes assessed as hazards for the top level system.

A comprehensive investigation over the function of the lower levels of the fuel level display is not available. Since this is the case, connections between levels are not made. However,



(a) State Machine over H2.

(b) State Machine over H5.

Figure 8.2.2: Hazard H2 and H5 as state machines.

the second level of the system is known to consist of three basic components; estimation (estimates the fuel level based on system input), threshold (sets warning parameter based on output from estimation) and icl (sets lamp and gauge based on estimation and threshold). These components and their failure modes are given as state machines in Figure 8.2.3. Even lower levels of abstraction in the fuel level display would have lead us to specific hardware and software parts. It is these parts that have to be individually tested to meet failure rates.

For simplicity, the system is assumed to have only two levels, hence failures at the second level can directly be mapped to software and hardware failure probabilities. By giving the transitions between nominal and failure modes at the second level, a transition matrix for the second level system can be calculated. This matrix will give the probabilities that the system currently is in a faulty state. These probabilities are then mapped to transitions in the level above, in this case the top level. When this is done, a transition matrix can be calculated and hazard probabilities are given.

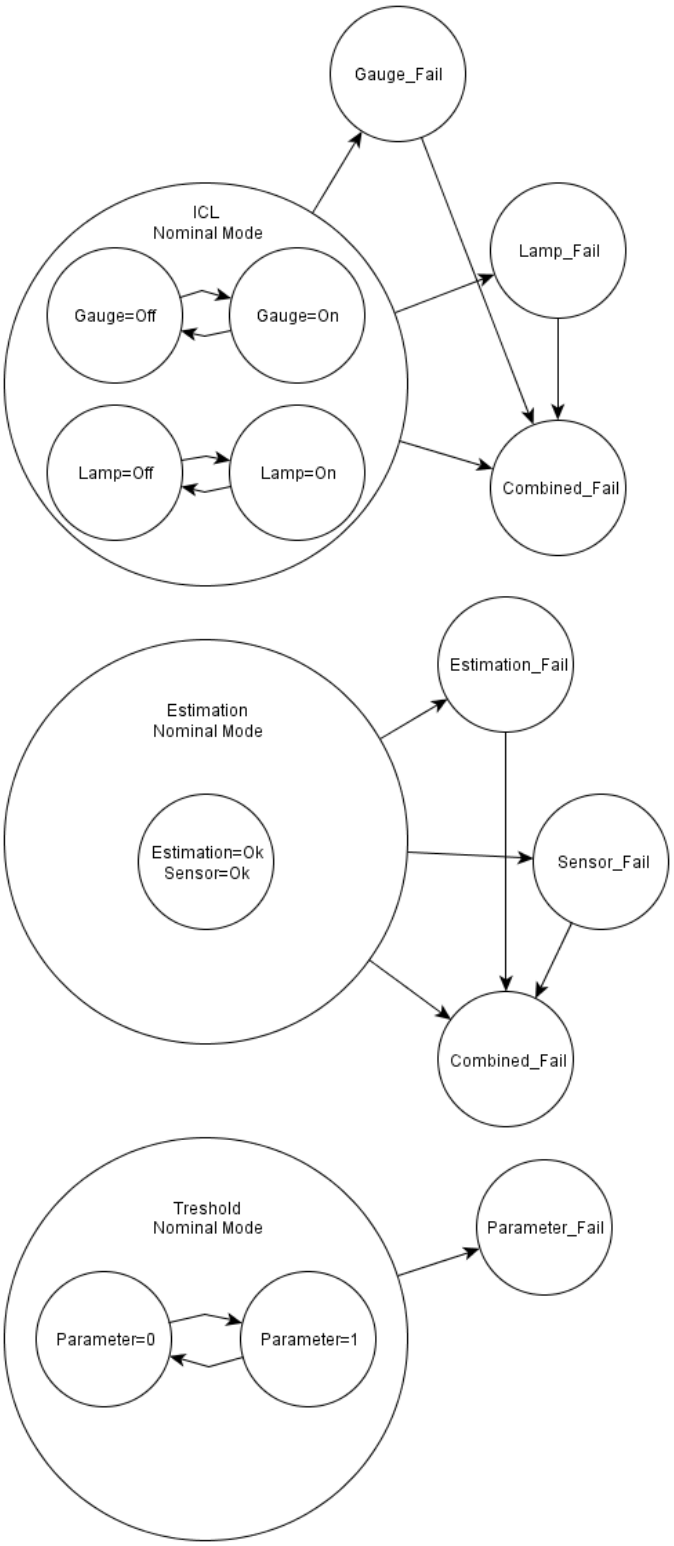


Figure 8.2.3: State machine with nominal mode and failure modes for the second level system. Note that the internal states of the failure modes of the three components are not illustrated.

8.3 Hazard Probability Calculations

Using the required failure rates in Table 8.1.5 and methodology discussed in Section 6.3.1, the probabilities of the hazards can be calculated. These probabilities give the likelihood of being in a specific hazard after 1 hour of driving.

First we derive the P matrix for the top level system. Since failure do not occur at specific time steps, but rather in a stochastic fashion, we use a continuous time markov chain. The P matrix derives to

$$P = \begin{pmatrix} 10^{-0}dt & \lambda_{0,1}dt & \lambda_{0,2}dt & \lambda_{0,3}dt & \lambda_{0,4}dt & \lambda_{0,5}dt & \lambda_{0,6}dt & \lambda_{0,7}dt & \dots \\ 0 & 10^{-0}dt & 0 & 0 & 0 & 0 & \lambda_{1,6}dt & \lambda_{1,7}dt & \dots \\ 0 & 0 & 10^{-0}dt & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 10^{-0}dt & 0 & \lambda_{3,5}dt & \lambda_{3,6}dt & 0 & \dots \\ 0 & 0 & 0 & 0 & 10^{-0}dt & \lambda_{4,5}dt & 0 & \lambda_{4,7}dt & \dots \\ 0 & 0 & 0 & 0 & 0 & 10^{-0}dt & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 10^{-0}dt & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10^{-0}dt & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \end{pmatrix}$$

$$\begin{pmatrix} \dots & \lambda_{0,8}dt & \lambda_{0,9}dt & \lambda_{0,10}dt & \lambda_{0,11}dt \\ \dots & 0 & 0 & 0 & \lambda_{1,11}dt \\ \dots & \lambda_{2,8}dt & \lambda_{2,9}dt & \lambda_{2,10}dt & 0 \\ \dots & \lambda_{3,8}dt & 0 & 0 & 0 \\ \dots & 0 & \lambda_{4,9}dt & 0 & 0 \\ \dots & 0 & 0 & \lambda_{5,10}dt & \lambda_{5,11}dt \\ \dots & 0 & 0 & 0 & \lambda_{6,11}dt \\ \dots & 0 & 0 & 0 & \lambda_{7,11}dt \\ \dots & 10^{-0}dt & 0 & \lambda_{8,10}dt & 0 \\ \dots & 0 & 10^{-0}dt & \lambda_{9,10}dt & 0 \\ \dots & 0 & 0 & 10^{-0}dt & 0 \\ \dots & 0 & 0 & 0 & 10^{-0}dt \end{pmatrix} \quad (8.3.1)$$

where failure rates connecting two failure modes, e.g. $\lambda_{1,2}$, are given the same value as its direct failure rate, hence $\lambda_{1,2} = \lambda_{0,2}$. As seen in the matrix, a lot of transition probabilities equals zero. A zero means that no transition between states is possible, where the use of non-repairable components implies an upper triangular matrix. The possible transitions are illustrated graphically in Figure 8.2.1. As for the lower levels of the fuel level display, their failure rate requirements can be calculated in a similar way, by inheriting requirements given in higher levels. However, since no comprehensive investigation about the lower level function is at hand, these calculations are not covered here.

Based on the P matrix, the individual probability of being in a specific hazard is calculated using Equation (6.3.5), where λ_{in} and λ_{out} for each hazard is their respective incoming and

outgoing transition rate. The probabilities becomes

$$P(H = H1) = Q_1(1) = 9.999984 \cdot 10^{-7} \tag{8.3.2}$$
$$P(H = H2) = Q_2(1) = 9.999984 \cdot 10^{-7} \tag{8.3.3}$$
$$P(H = H3) = Q_3(1) = 9.999989 \cdot 10^{-7} \tag{8.3.4}$$
$$P(H = H4) = Q_4(1) = 9.999980 \cdot 10^{-7} \tag{8.3.5}$$
$$P(H = H5) = Q_5(1) = 1.000004 \cdot 10^{-6} \tag{8.3.6}$$
$$P(H = H6) = Q_6(1) = 1.000001 \cdot 10^{-7} \tag{8.3.7}$$
$$P(H = H7) = Q_7(1) = 1.000001 \cdot 10^{-6} \tag{8.3.8}$$
$$P(H = H8) = Q_8(1) = 1.000001 \cdot 10^{-7} \tag{8.3.9}$$
$$P(H = H9) = Q_9(1) = 1.000001 \cdot 10^{-6} \tag{8.3.10}$$
$$P(H = H10) = Q_{10}(1) = 1.000003 \cdot 10^{-6} \tag{8.3.11}$$
$$P(H = H11) = Q_{11}(1) = 1.000004 \cdot 10^{-6} \tag{8.3.12}$$

where hazards H5, H9-H11 and H7 appear to have a higher probability after 1 hour of usage, than their given failure rates indicates.

8.4 Estimated Loss Associated with the Fuel Level Display

By using the model derived in Section 4.2, where the loss of a system is calculated based on values gathered from a ISO 26262 hazard analysis, the combined loss associated to the fuel level display system. The probability of exposure, severity and controllability classifications is translated into probabilistic terms and used as in the mentioned section. The loss calculates to

$$\begin{aligned}
\mathbb{E}_{fld}(S) &= \sum_H \sum_O S(C = c_1, H, O) P(C = c_1, H, O) = \\
&= \sum_H \sum_O P(\text{fatal injury} | C = c_1, O, H) P(C = c_1 | H, O) P(H) P(O) = \\
&= \ldots + P(\text{fatal injury} | C = c_1, O12, H8) P(C = c_1 | H8, O12) P(H8) P(O12) + \ldots = \\
&= \ldots + 10^{-0} \cdot 10^{-0} \cdot 10^{-7} \cdot 10^{-1} + \ldots = \\
&= 10^{-8} h^{-1}
\end{aligned} \tag{8.4.1}$$

This means that if the requirements on failure rates are fulfilled, the fuel level display do meet the required maximum probability of fatal injury given in ISO 26262.

8.5 Possible Implementation in Modelica

In Chapter 7 the Modelica language was discussed. This discussion described what parts of the method that is currently supported. Based on the supported functionality, the fuel level display is modeled in Modelica.

Figure 8.5.1 illustrates the ordering of classes. By using this way of ordering classes, new systems can easily be added later, keeping it manageable. In more detail, a package is created for the fuel level display (UF18). Inside this package, the main class called Fuel_Level_Display is defined together with a package consisting of its underlying classes. These classes are discussed in detail in the following pages.

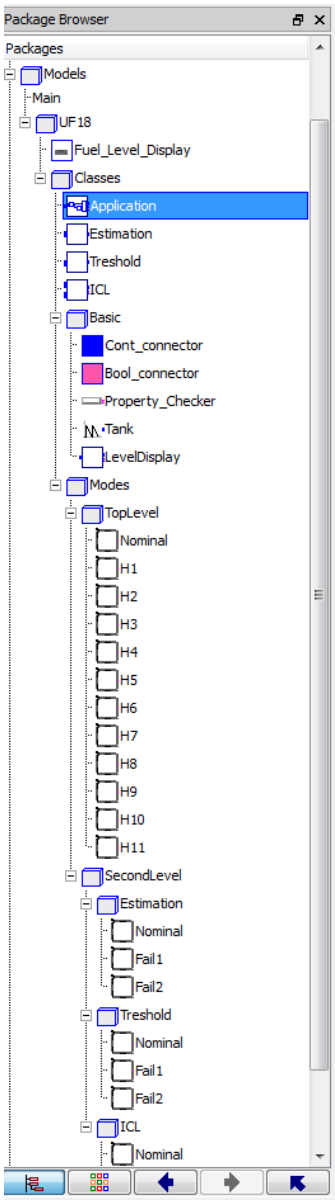


Figure 8.5.1: Fuel Level Display class overview.

The graphical representation of the Fuel_Level_Display class is shown in Figure 8.5.2. This class defines the basic input and outputs of the system, which are the fuel volume, light from the lamp and level given by the gauge. The component called application (see figure) is defined in the Application class. Note that this class could be extended with more accurate models of the tank and gauge, in order to get a more realistic model.

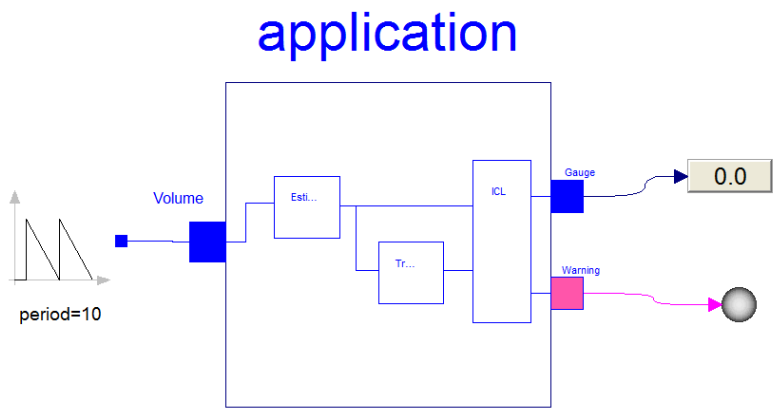


Figure 8.5.2: Fuel Level Display input and output.

Looking into the application class, shown in Figure 8.5.3, a bisectional view is modeled. Here the upper part of the model illustrates the physical components of the system, whereas the lower part illustrates the abstract view incorporating nominal and failure modes. The physical components are defined in classes Estimate, Threshold and ICL. As for the abstract view, every transition from nominal mode to a failure mode, or between failure modes, is connected to a property. These properties are shown in the lower left corner and either gives a true or false value, in order to fire a transition. These properties are based on current physical properties of the system.

Furthermore, every mode, nominal or failure, is a hierarchical state, meaning they consist of internal functional states. Figure 8.5.4, Figure 8.5.5 and Figure 8.5.6 show the internal states of the nominal mode, hazard H1 and hazard H2 respectively. In the nominal mode, the system function as it is supposed to function. Again, the state transitions are controlled by properties.

State machines active when a failure occurs are defined in the same way as for the nominal mode, differing in possible states and their transitions. Recalling the definition of hazard H1; ”Fuel Level Warning displayed when it should not and Fuel Level Gauge indicates a correct fuel level” and hazard H2; ”Fuel Level Warning not displayed when it should and Fuel Level Gauge indicates a correct fuel level”. This means that in the case of hazard H1, the system can only be in either state Off_noShow or On_Show. As for hazard H2, the states Off_Show and Off_Show are the only possible ones.

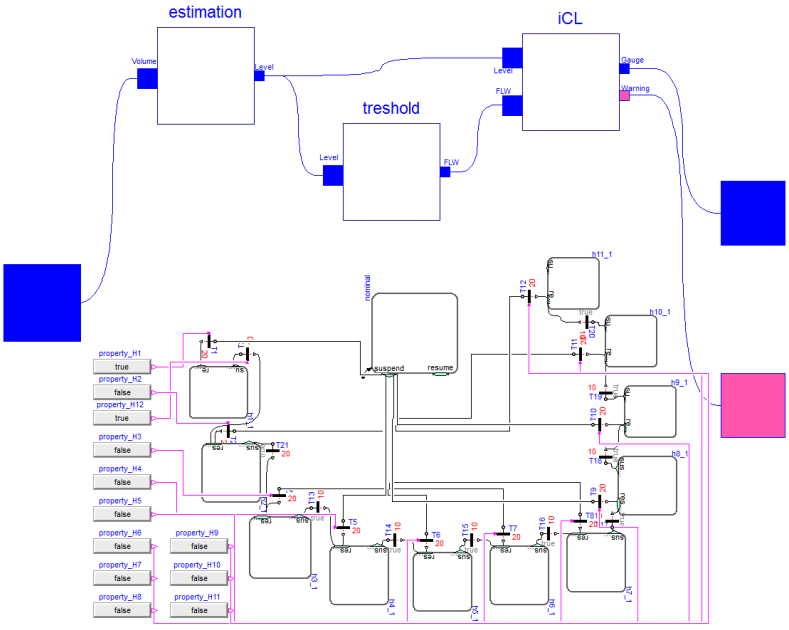


Figure 8.5.3: Fuel Level Display internal view. Physical and states illustrated simultaneously.

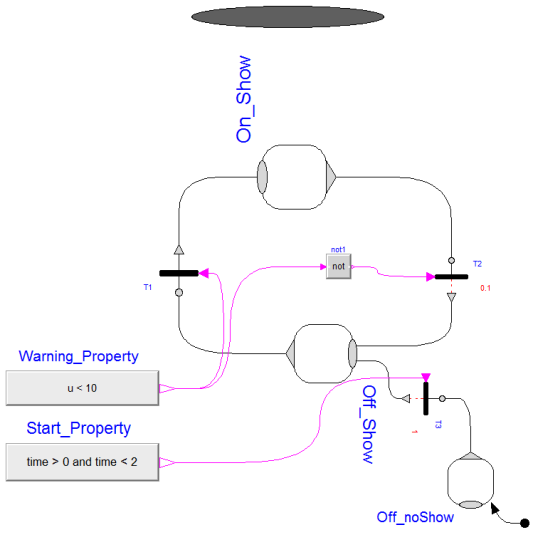


Figure 8.5.4: Nominal mode of the top level.

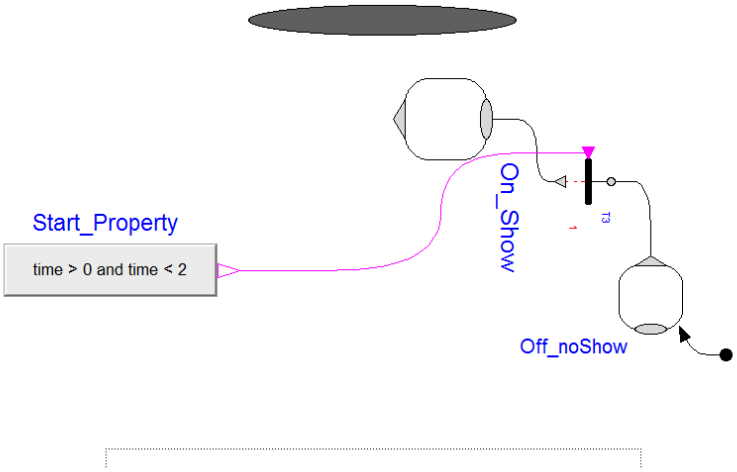


Figure 8.5.5: Hazard H1.

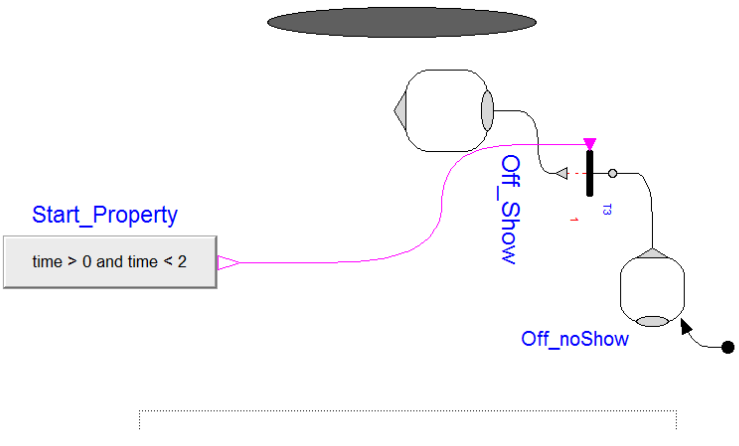


Figure 8.5.6: Hazard H2.

As stated before, the fuel level display consists of three separate components in a second level. These were called Estimation, Threshold and ICL. The component Estimation is illustrated in Figure 8.5.7, where only the abstract part has been modeled due to contingency about the actual software and hardware components and connections. However, the nominal mode and failure modes are present, illustrated using hierarchical states. Similar to previous discussion about these hierarchical states, the nominal mode internal state is shown in Figure 8.5.8.

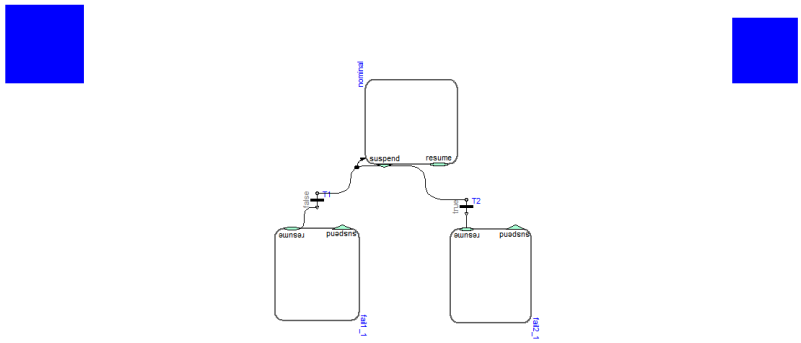


Figure 8.5.7: The estimation component of the fuel level display.

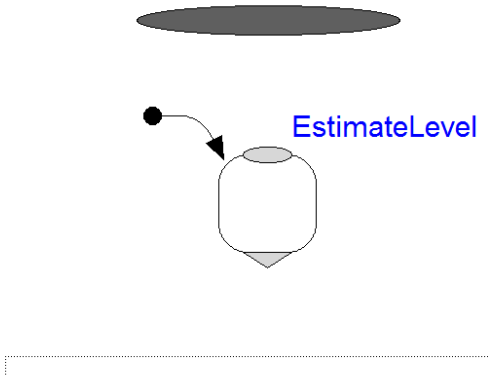


Figure 8.5.8: Nominal mode of the estimation component.

In Figure 8.5.9 the component, in which a parameter for the warning is set, is shown. This component uses the output of the Estimation component to decide if this parameter is to be set. Again, the physical representation is lacking, for the same reason as for the Estimation component. The nominal mode and failure modes are once again represented in accordance with the other components. Figure 8.5.10 illustrates the internal states of the nominal mode.

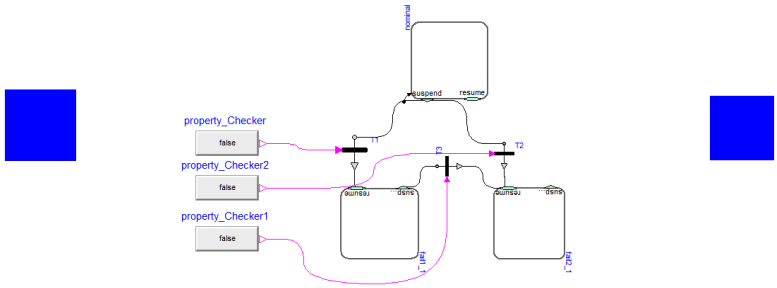


Figure 8.5.9: The threshold component of the fuel level display.

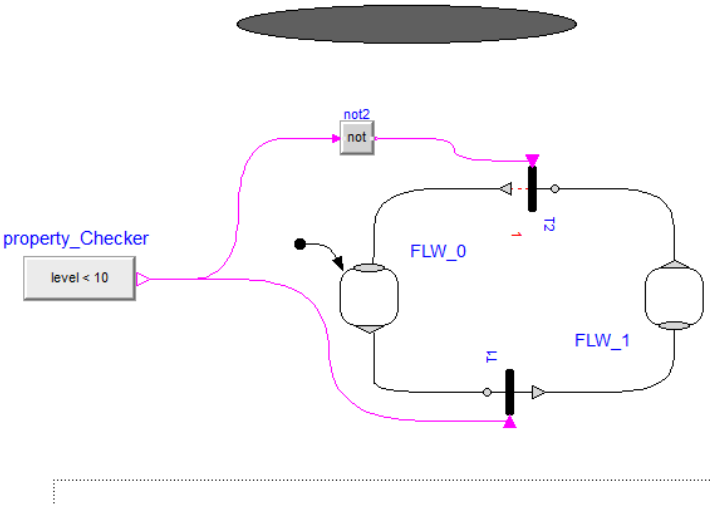


Figure 8.5.10: Nominal mode of the threshold component.

Finally, Figure 8.5.11 and Figure 8.5.12 illustrates the ICL component and the internal states of its nominal mode.

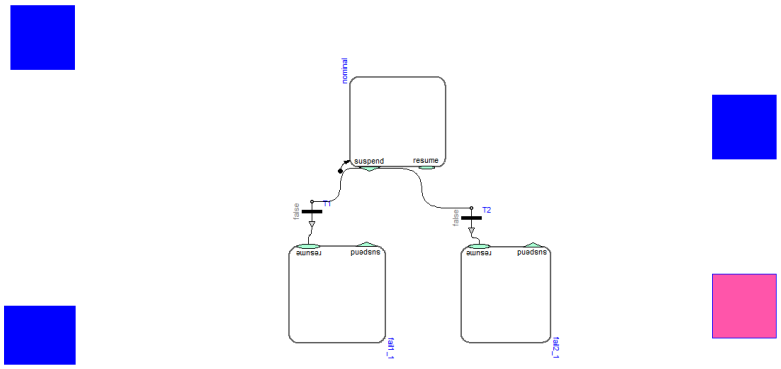


Figure 8.5.11: The icl component of the fuel level display.

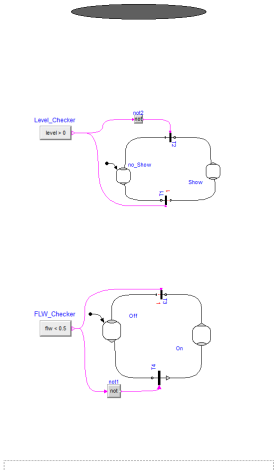


Figure 8.5.12: Nominal mode of the icl component.

8.6 Experience from the Fuel Level Display Example

The fuel level display is considered as one of the least complex systems in a heavy truck. Although, this system introduced some very interesting properties and do a great job as a benchmarking example. The basics of this system, described in state machines and assessed in a hazard analysis are easily extended in more complex implementations. The CTMCs are also well suited to be extended in more complex implementations. However, by merely considering a smaller system, the needed assessments and loss calculations become relatively big. In a more complex implementation these things will grow even bigger, illustrating a great need for software tools to automatise as much as possible. Furthermore, the fuel level display system consists of two ”parallel” subsystems. This parallelism creates problems in various places, such as in hazard identification or basic system description. If this parallel nature of a system can be avoided, a determination of the probability of fatal injury becomes less complex. Avaoiding parallel systems would also avoid multiple point failures, hence the hazards would be defined diferently.

The probability of fatal injury associated with the fuel level display system is $10^{-8}h^{-1}$, which corresponds to the maximum value given in ISO 26262. This value should not be taken too literary, but gives an indication about the safety of the system compared to requirements and other systems. The Modelica language, using the Dymola tool, does cover a lot of the things of interest when implementing the method developed in this thesis. However, there are, as discussed in Section 7.3, some essential functionality missing.

Chapter 9

Discussion and Overview of Proposed Method

This thesis investigates the possibilities of safety assessment using quantitative techniques such as Bayesian networks, influence diagrams and continuous time Markov chains. Bayesian networks are found useful when modeling causal accident processes, including extension with utilities defined in influence diagrams, and calculating total expected loss, as a probabilistic value, for an entire model. These calculations are based on assessments of severity, controllability and probability of exposure as described in ISO 26262. Probabilities associated with failures or hazards are not covered in the ISO 26262 assessments. Here requirements gained from identifications of hazards gave failure rate requirements. Based on these failure rates, CTMCs are discussed as a possible way of modeling the interdependencies between failure modes mapped to probabilities, resulting in a top level transition matrix, giving individual probabilities of hazards. Finally, Modelica is introduced as a tool for practical implementation of the derived models. However, Modelica appeared to lack some functionality to fully implement the complexity of the models.

The quantitative approach of safety assessment given in this paper does not give a complete understanding of every consequence and causing process. It merely provides a tool to use in combination with other more qualitative methods. However, one of the advantages of this quantitative approach is that it provides a straight-forward way of comparing different systems using a single value. This enables comparisons between functions and how changes in a function affect safety. The value itself might be hard to translate into real world events and has to be seen as a way to somehow benchmark safety properties of a system.

Furthermore, the question if it is better to describe safety through natural language or by numbers is another area of interest, since safety assessments, even when using numbers as in this thesis, are always at some point dependent upon human judgment. There are currently no alternatives as how to asses severity, controllability and probability of exposure, but these assessments should, to a greater, extent be based on actual accident statistics, in order to secure the accuracy. Another problem regarding safety assessments is the need for consistency, meaning all assessments need to be made in the exact same way.

As seen in previous sections, all required functionality is not supported by Modelica. Lack of

support for probabilities is the most problematic. If this functionality takes too much effort to actually implement, other modeling software should be considered. Additionally, a way to integrate calculations for transition matrices, which then are used in loss calculations, need to be investigated.

Based on the findings discussed in this thesis, a working process when estimating safety quantitatively follows:

1. **Modelica**
   The Modelica language, incorporated in the Dymola software, is used, where applicable, throughout the method in order to support the assessment process.

2. **State Machine**
   As a basis for the understanding of a system, or function, some kind of system description is essential. State machines serve this purpose well and are useful in the further analysis; hence top-down state machines are derived of a system.

3. **Identify Possible Failure modes**
   When having defined the system using state machines, possible failure modes, relevant for the specific system, are identified. Starting with a state machine the failure modes are identified using qualitative techniques, such as brainstorming.

4. **Consequence Analysis of Failures**
   The previously identified failure modes are now related to possible accidents. These accidents can be anything from crashing into a tree or having to stop at the side of a road.

5. **Identify Hazards**
   Based on the consequence analysis, those failure modes associated with potential of injury to a human, driver or other involved people, are identified as hazards.

6. **Hazard Analysis**
   Now, when having identified relevant hazards for the system, a hazard analysis as described in ISO 26262 is performed. This hazard analysis will assess probability of exposure, severity and controllability. This implies identification of operational situations, as described in Section 8.1.3.

7. **Map Required Failure Rates**
   Required failure rates are gathered from the highest ASIL associated with every hazard, based on the result from the hazard analysis.

8. **Continuous Time Markov Chains**
   Derive CTMCs for the system. The CTMCs are defined as hierarchical states where the top level system description incorporates nominal and failure modes, while the internal states model the function of the system.

9. **Subsystems**
   Repeat the process for associated subsystems, with the difference that required failure rates are inherited from the level above. In the subsystems, all failure modes are taken into account since no hazards can be identified. At the lowest level the failure requirements are mapped to specific software and hardware components.

10. **Probabilistic Safety Assessment**
    Check or calculate safety requirements. The top level transition matrix gives hazard probabilities needed in the Bayesian networks, which calculates the expected loss associated to a system. Bayesian networks connecting the different levels are also derived, in order to get required failure rates on the individual hardware and software components.

The derived method implies that, to truly understand the nature of safety, complete knowledge of all possible combinations of hazards and operational situations combined with their respective consequences is needed. Only the models derived in this thesis is not enough for a complete understanding, but does, as stated, provide a way to compare different systems or functions to each other, combined with possibilities to illustrate how a system is affected by changes, in terms of safety. However, the value itself might be hard to translate into real world events, which is one of the major disadvantages when considering if a quantitative analysis should be used in the development process of new functions.

In this thesis the derived method is only implemented using one example system, the fuel level display. To conclude the actual usefulness of the method, several examples should be carried out. Only then, the expected advantages can be thoroughly discussed. Another problem with the derived method is the lack of experience at Scania concerning quantitative safety analysis. Finally, at companies such as Scania, the calculation of a value called loss, or probability of fatality, might lead to problems from a legal point of view.

9.1 Compilation of Desired Modelica Extensions

If the modeling language Modelica is to be used in implementations, such as the ones discussed in this thesis, the language needs to be extended. The needed extensions are summarized as:

- To be able to implement CTMCs in Modelica, the Modelica_StateGraph2 library offers some basic functionality. However, full support for properties incorporating probabilities and extra ports to connect physical and abstract views are needed.
- Improved graphics and code coordination. In present Modelica versions, it is hard to create good looking graphics. By improving graphics generation, combined with better code and graphic correlation, the impression and application handiness would improve.
- Some way to represent calculations. This might be done by adding support for other software, to be used together with Modelica, or by adding support directly in Modelica.
- A library adding functionality for modeling Bayesian networks. Here support for graphical representations, as well as conditional probability tables, are needed. Ideally, changes in individual probabilities should be shown in real time during a simulation.

Finally, in ISO 26262 and in system design work at Scania, a lot of various documents are generated in order to describe functionality and architecture of a system. This has not been discussed previously, but Modelica does support functionality to add text based descriptions to a class. If Modelica is to be used in safety assessments, this text based documentation functionality in Modelica might as well be used to generate some of the system documents as well.

Chapter 10

Conclusions

This thesis examined the possibilities to derive and implement a method for safety assessment based on quantitative techniques. Techniques such as Bayesian networks and CTMCs are used in combination with information given in common functional safety standards. In order to make this method usable in practice, Modelica was discussed as a potential alternative. Based on findings throughout this thesis, the following conclusions are made:

- By using Bayesian networks, a general accident model was successfully derived. When deriving such a model, incorporating causal relationships and conditional probabilities, the use of Bayesian networks is a modeling tool well suited. However, to get a manageable model, simplifications are imminent. If no simplifications are made, the amount of probabilities to consider easily becomes very large.

- In the increasingly safety aware heavy automotive industry, implementation of the functional safety standard ISO 26262 is likely to become necessary in the near future. By successfully adjusting the general accident model to use information gathered when implementing the standard, the derived accident model will damand less work to implement. However, this thesis has shown several different ways to incorporate classifications made in the ISO 26262 standard, resulting in the same value of expected loss assigned to a system.

- As for modeling faults and failures, the most promising method is the use of Continuous Time Markov Chains. Many alternatives do exist, and this area of research experience increased interest in academia. By using CTMCs combined with hierarchical state machines, nominal and failure modes and function oriented states are modeled together. Resulting in probabilities of hazards, which are used in the Bayesian networks.

- The Modelica language does hold some interesting properties allowing physical and abstract views of a system to be modeled together. However, the present version of Modelica does not support all necessary functionality to fully implement the entire method for quantitative safety assessment derived in this thesis. Hence, further research need to be made concerning implementations, if the derived method are to be used in future system safety assessments. Main extentions needed are probability property modeling with markov models and Bayesian network modeling.

- In more general terms, if the functional safety standard ISO 26262 is to be used in the heavy automotive industry, some changes are preferable. Relevant changes found throughout this thesis are; a need for expanding the available levels of severity to differ between few and many persons getting hurt, reconcidering the accuracy of the maximum required value of risk, clearer definitions of controllability, hazards and operational situations, and adding of environmental and property loss factors.
- When using quantitative analysis techniques, the resulting value should not be interpreted as a definite measure of the safety of a system. Since a quantitative analysis is dependent on qualitative assessments, such as the ones made in an ISO 26262 hazard analysis, such assessments are likely to give rise to faulty values. However, if the qualitative assessments are made in similar ways for every system, then the resulting values from quantitative analyses are of great use when comparing systems or what impact a system change has on safety.

As for the complete derived method, a direct implementation at Scania is not likely in the near future. However, the results will hopefully function as a basis for further research and inspiration in later assessments.

10.1 Considerations and Further Research

It should be pointed out that the method developed in this thesis merely gives a proposition on how a quantitative analysis method might be used in principle. Further research will have to be done mainly focusing on creating Modelica libraries incorporating the identifyed lacking functionality, but also questions concerning model checking as a way to determine the correlation between a system description of a model and various function requirements. Also implementations where repairs are allowed, and modeled accordingly, should be further investigated.

Concerning more general areas of research, alternatives to Markov models in implementations, such as the ones discussed in this thesis, should be considered further. Here mainly fault tree analysis is of interest. As for the software implementation, Modelica did not support all desired functionality. Investigating alternatives to Modelica, to find a better suited tool that offer a complete functionality, should be considered.

Furthermore, another unanswered question is if it really is better to describe safety through numbers compared to natural language? By using natural language, it is easy to understand and interpret the level of safety, but instead these descriptions can be interpreted in different ways, for good or for worse? Since safety assessments, even when using numbers as in this thesis, are always at some point dependent upon human judgment, maybe qualitative analysis is the most relevant approach.

Finally, this thesis has covered faults and failures based on a top-down system view, where failure rates are assigned starting at the top, resulting in required failure rates on specific software and hardware components. A future area of research is to approach the safety assessment based on a bottom-up view. This way, the modeling process starts with statistical analysis of the relevant components of a system, giving failure rates. These failure rates are then related to upper level descriptions, hopefully leading to probabilities of hazards. Note

that in a bottom-up view, as well as in a top-down view, the problem concerning common cause failures should be further investigated.

Appendix A

Vocabulary

ISO 26262 is a standard for functional safety used mainly in the automotive industry. The standard describes several words and phrases. In this appendix those words and expressions that are used to a great extent throughout the thesis are presented. Furthermore some changes and additions are made in order to simplify for the reader.

A.1 Basic Vocabulary in ISO 26262

Element
Part of a system. Consists of hardware and/or software [19, part 1].

Exposure
State of being in an operational situation that can be hazardous if coincident with the failure mode under analysis [19, part 1].

Failure, F (also referred to as System Failure)
Termination of ability of a system to perform function as required [19, part 1].

Failure Mode
Manner in which an element or an item fails [19, part 1].

Hazard, H
Potential source of harm through vehicle level failures, caused by malfunctioning behavior of the item [19, part 1].

Hazardous Event
Combination of a hazard and an operational situation [19, part 1].

Item
Individual item or array of systems that implements a function at the vehicle level which is being investigated, e.g. the fuel level display [19, part 1].

Operational Situation, O
Scenario that may occur during the life of a vehicle, e.g. driving, parking, and maintenance [19, part 1].

System
Set of elements that relates at least a sensor, a controller and an actuator with one another [19, part 1].

A.2 Changes used in the Context of this Thesis

System
The item described above will be referred to simply as a system, i.e. the previously notations of item and system have been merged to simply system. This to simplify the notation towards a more wide spread expression. A system is an element or collection of elements whose properties are to be studied.

A.2.1 Additions used in the Context of this Thesis

Accident, A
An undesired and unplanned event or circumstance.

Accident Management
Actions taken to reduce the impact of an accident on life, property or the environment.

Accident Prevention
Actions taken to prevent an accident to happen.

Environment Factors
Characteristics of the environment in which the system operates.

Fatality
An accident where one or several people disease due to system malfunctioning.

Human Error
Action taken by a human with the potential to lead to an accident.

Human Preventive Action
Action taken as a response to failure and operational situation to prevent an accident to happen.

Loss, L
Damage to life, property or the environment.

Nominal Mode
When the system works as it is supposed and state transitions happens as they are meant to, the system acts in nominal mode.

A.3 Vocabulary Associated with Classification of Hazardous Events in ISO 26262

Controllability, C
The involved persons' (drivers, passengers or persons close to the exterior of the car) ability to avoid specific harm or damage through preventing reactions with or without external measures. This controllability is defined by one out of four classes where the fourth class is defined as uncontrollable [19, part 3].

Table A.3.1: Classes of controllability according to ISO 26262.

	C0	**C1**	**C2**	**C3**
Description	Controllable in General	Simply Controllable	Normally Controllable	Difficult to Control or Uncontrollable

Probability of Exposure, P(O)
This is the probability of exposure of a specific operational situation and is estimated based on a five level scale from incredible to high probability [19, part 3].

Table A.3.2: Classes of probability of exposure according to ISO 26262.

	E0	**E1**	**E2**	**E3**	**E4**
Description	Incredible	Very Low Probability	Low Probability	Medium Probability	High Probability

Severity, S
The severity of potential harm for hazardous events are estimated through four different classes spanning from no injuries to fatal or life-threatening injuries [19, part 3].

Table A.3.3: Classes of severity according to ISO 26262.

	S0	**S1**	**S2**	**S3**
Description	No injuries	Light and moderate injuries	Severe and life-threatening injuries (survival probable)	Life-threatening injuries (survival uncertain), fatal injuries

Automotive Safety Integrity Level, ASIL
An approach to determine risk classes in specific systems or elements. Levels span from $\mathcal{A}$ to $\mathcal{D}$, where $\mathcal{D}$ represents the most stringent safety requirements for achieving an acceptable residual risk. The levels are determined using the parameters severity, probability of exposure and controllability as in table A.3.4. $\mathcal{QM}$ (quality management) classed combinations means very low risk. Every hazardous event is ASIL classified and allocated to an element or system. The ASIL classifications based on severity, controllability and probability of exposure are shown in Table A.3.4 [19, part 3].

Table A.3.4: ASIL classification in ISO 26262 based on the S, E and C classes. Any combination containing S0, E0 or C0 are classified as $\mathcal{QM}$.

Severity Class	**Probability Class**	**Controllability Class**		
		C1	**C2**	**C3**
S1	**E1**	$\mathcal{QM}$	$\mathcal{QM}$	$\mathcal{QM}$
	E2	$\mathcal{QM}$	$\mathcal{QM}$	$\mathcal{QM}$
	E3	$\mathcal{QM}$	$\mathcal{QM}$	$\mathcal{A}$
	E4	$\mathcal{QM}$	$\mathcal{A}$	$\mathcal{B}$
S2	**E1**	$\mathcal{QM}$	$\mathcal{QM}$	$\mathcal{QM}$
	E2	$\mathcal{QM}$	$\mathcal{QM}$	$\mathcal{A}$
	E3	$\mathcal{QM}$	$\mathcal{A}$	$\mathcal{B}$
	E4	$\mathcal{A}$	$\mathcal{B}$	$\mathcal{C}$
S3	**E1**	$\mathcal{QM}$	$\mathcal{QM}$	$\mathcal{A}$
	E2	$\mathcal{QM}$	$\mathcal{A}$	$\mathcal{B}$
	E3	$\mathcal{A}$	$\mathcal{B}$	$\mathcal{C}$
	E4	$\mathcal{B}$	$\mathcal{C}$	$\mathcal{D}$

List of Figures

List of Tables

Source Reference

[1] N. G. Leveson, *Safeware - System Safety and Computers*. USA: Addison-Wesley Publishing Company, Inc., 1995.

[2] L. B. Neil, M. and N. Fenton, "Applying bayesian belief networks to systems dependability assessment," in *Proceedings of Safety Critical Systems Club Symposium*, (Leeds), Springer-Verlag, 1996.

[3] J. Börcsök, *Functional Safety - Basic Principles of Safety Related Systems*. Heidelberg, Germany: Hütig GmbH & Co, 2007.

[4] International Council on Systems Engineering, INCOSE, *What is Systems Engineering?*, 2011. Retrieved September 6, 2011 from `http://www.incose.org/practice/whatissystemseng.aspx`.

[5] H. Kumamoto and E. Henley, *Probabilistic Risk Assessment and Management for Engineers and Scientists*. New York: IEEE Press, 2nd ed., 1996.

[6] Scania, "Scania group," 2011. Retrieved June 14, 2011 from `http://scania.com/scania-group/`.

[7] Fordonsstrategisk Forskning och Innovation, Vinnova, *BeSafe*, 2010. Retrieved June 14, 2011 from `http://www.vinnova.se/sv/Resultat/Projekt/Effekta/BeSafe/`.

[8] N. Friedman and D. Koller, *Probabilistic Graphical Models - Principles and Techniques*. Cambridge, Massachusetts, USA: The MIT Press, 2009.

[9] F. Jensen and T. Nielsen, *Bayesian Networks and Decision Graphs*. London: Springer-Verlag, 2007.

[10] C. Grinstead and J. Snell, *Introduction to probability*. Providence, RI, USA: American Mathematical Society, 1997.

[11] W. K. Ching and M. K. Ng, *Markov chains: models, algorithms and applications*. New York: Springer, 2006.

[12] P. Popov and G. Manno, "The effect of correlated failure rates on reliability of continuous time 1-out-of-2 software," in *Computer Safety, Reliability and Security; 30th International Conference, SAFECOMP 2011, Naples, Italy September 2011*, (Berlin Heidelberg, Germany), Springer-Verlag, 2011.

[13] P. Norvig and S. Russell, *Artificial Intelligence - A Modern Approach*. Upper Saddle River, New Jersey, USA: Pearson Education, Inc., 2010.

[14] F. Felger and G. Frey, "Multi-phase markov models for functional safety prediction," in *3rd International Workshop on Dependable Control of Discrete Systems (DCDS)*, (Saarbrucken, Germany), IEEE Press, 2011.

[15] T. Murata, "Petri nets: Properties, analysis and applications," *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541–580, 1989.

[16] Merrian-Webster, 2011. Retrieved June 19, 2011 from `http://www.merriam-webster.com/`.

[17] A. Onisko, M. Druzdzel, and H. Wasyluk, "Learning bayesian network parameters from small data sets: Application of noisy-or gates," *International Journal of Approximate Reasoning*, vol. 27, pp. 165–182, 2001.

[18] M. Güdemann and F. Ortmeier, "Probabilistic model-based safety analysis," *Electronic Proceedings in Theoretical Computer Science (EPTCS)*, vol. 28, pp. 114–128, 2010.

[19] International Organization for Standardization, ISO, Geneva, Switzerland, *Road Vehicles - Functional Safety, ISO/FDIS 26262*, 2011. ISO Copyright Office.

[20] International Electrotechnical Commission, IEC, Brussels, Belgium, *Functional safety of electrical/electronic/programmable electronic safety-related systems, IEC 61508*, 2010. CENELEC.

[21] T. Tengs, M. Adams, J. Pliskin, D. Safran, J. Siegel, M. Weinstein, and J. Graham, "Five-hundred life-saving interventions and their cost-effectiveness," *Risk analysis*, vol. 15, no. 3, pp. 369–390, 1995.

[22] K. Kingsbury, "The value of a human life: $129,000," *Time Health*, 2008. Retrieved October 17, 2011 from `http://www.time.com/time/health/article/0,8599,1808049,00.html`.

[23] J. Tångring, "Iso ska trygga din bil," *Elektroniktidningen*, 2009. Retrieved August 16, 2011 from `http://www.etn.se/index.php?option=com_content&view=articleid=50285`.

[24] Trafikverket, "Analys av trafiksäkerhetsutvecklingen 2010, målstyrning av trafiksäkerhetsarbetet mot etappmålen 2020," tech. rep., Trafikverket, Borlänge, Sweden, 2011.

[25] M. Xie, Y. Dai, and K. Poh, *Computing Systems Reliability - Models and Analysis*. Hingham, MA, USA: Kluwer Academic Publishers, 2004.

[26] A. Jardin, D. Bouskela, T. Nguyen, N. Ruel, E. Thomas, L. Chastanet, R. Schoenig, and S. Loembé, "Modelling of system properties in a modelica framework," in *Proceedings of the 8th International Modelica Conference, March 20th-22nd, Technical Univeristy, Dresden, Germany*, (Linköping, Sweden), Linköping University Electronic Press, Linköpings universitet, 2011.

[27] UPPAAL, "About," 2011. Retrieved October 19, 2011 from `http://www.uppaal.org/`.

[28] M. Güdemann and F. Ortmeier, "A framework for qualitative and quantitative formal model-based safety analysis," in *Proceedings of the 12th International Symposium on High Assurance Systems Engineering*, (San Jose, CA, USA), Conference Publishing Services, 2010.

[29] J. Rouvroye and E. G. van den Bliek, "Comparing safety analysis techniques," *Reliability Engineering & System Safety*, vol. 75, no. 3, pp. 289–294, 2002.

[30] J. Blanquart, E. Armengaud, P. Baufreton, Q. Bourrouilh, G. Griessnig, M. Krammer, O. Laurent, J. Machrouh, T. Peikenkamp, C. Schindler, and T. Wien, "Towards cross-domains model-based safety process, methods and tools for critical embedded systems: The cesar approach," in *Computer Safety, Reliability and Security; 30th International Conference, SAFECOMP 2011, Naples, Italy September 2011*, (Berlin Heidelberg, Germany), Springer-Verlag, 2011.

[31] F. Flamini, S. Bologna, and V. Vittorini, eds., *Computer Safety, Reliability and Security; 30th International Conference, SAFECOMP 2011, Naples, Italy September 2011*, (Berlin Heidelberg, Germany), Springer-Verlag, 2011.

[32] A. Hildebrandt, "Calculating the probability of failure on demand (pfd) of complex structures by means of markov models," in *Electrical and Instrumentation Applications in the Petroleum and Chemical Industry, 2007. PCIC Europe 2007. 4th European Conference*, (Mannheim, Germany), Pepperl+Fuchs GmbH, 2007.

[33] P. Fritzon, *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*. USA: John Wiley & Sons, Inc., 2004.

[34] Modelica Association, *Modelica Standard Libraries - Modelica_StateGraph2*, 2011. Retrieved October 4, 2011 from `https://modelica.org/libraries`.

[35] D. Bouskela, A. Jardin, Z. Benjelloun-Touimi, P. Aronsson, and P. Fritzson, "Modelling of uncertainties with modelica," in *Proceedings of the 8th International Modelica Conference, March 20th-22nd, Technical Univeristy, Dresden, Germany*, (Linköping, Sweden), Linköping University Electronic Press, Linköpings universitet, 2011.

[36] C. Schallert, "Inclusion of reliability and safety analysis methods in modelica," in *Proceedings of the 8th International Modelica Conference, March 20th-22nd, Technical Univeristy, Dresden, Germany*, (Linköping, Sweden), Linköping University Electronic Press, Linköpings universitet, 2011.