NTNU – Trondheim
Norwegian University of
Science and Technology

Hazard boilerplates in safety analysis

Aspects of hazard identification using
boilerplates and ontologies

Christian Overvåg Hjorth

Master of Science in Computer Science
Submission date: June 2013
Supervisor: Tor Stålhane, IDI

Norwegian University of Science and Technology
Department of Computer and Information Science

Problem description

In the specialization project (TDT4520) we studied the use of boilerplates for requirement elicitation and combining this with ontologies to check the requirements for completeness and consistency. By adding generic failure modes for equipment, we are able to perform an early safety analysis on a system under development. Based on the use of boilerplates templates and ontologies, we found three areas of interest to investigate further: human failure modes, boundary objects and cause-consequence chains.

We want to investigate how including human failure modes will affect the safety analysis. It is our belief that this will make it possible to examine certain aspects of the system, which is not possible by only using failure modes for the equipment. Further, we want to be able to infer global hazards starting from local system hazards. In order to do this, we will need a method for moving between these domains. This will be accomplished by either using a list/table or, if there is time available, cause-consequence chains.

We will run an experiment comparing the safety analysis procedure with and without this new additional information in form of human failure modes and global hazards. This will be achieved by using two different cases within the same system, where one test the human failure modes and the other the global hazards. The data collected will be analyzed, modifications to the procedure will be made, if needed, and advantages and disadvantages will be highlighted.

Assignment given: 15. January 2013

Supervisor: Tor Stålhane, IDI

Abstract

In the Specialization Project, we looked at methods of performing safety analysis in the early stages of development based upon the use of boilerplates and ontologies. Based on our work, we suggested two approaches for performing safety analysis: global hazards using can-cause chains and human failure modes. The method of global hazard focus on identifying events in a system that can cause hazards which affects the environment it operates in. The method of human failure modes introduces generic failures for human, in order to identify hazards related to the operator of the system.

We were interested in assessing how good our suggested methods were in identifying hazards during the safety analysis. To do this, we chose to create two research questions to be answered in this thesis:

RQ1: Is it easier to discover possible environment threatening hazards with global hazards and can- cause chains?

RQ2: Is it easier to discover possible operator hazards with human failure modes?

To answer our research questions, we chose perform an experiment with students using the suggested methods for safety analysis of two systems. The experiment gave us a good illustration of how the procedure would work in a real hazard analysis project.

The results for global hazards with can-cause chains indicate that the method is not in a state where it can be used for safety analysis as of yet. There are still too many ambiguities as too how the chains should be created, and the feedback from the students indicates that it is difficult to learn and use the method. The algorithm needs to be further structured and we must obtain better documentation of how to perform it.

The data from the experiment indicate that human failure modes have proven to be efficient at identifying operator related hazards. The method was given overall favorable feedback from the students, and appeared to identify many of the hazards in the test case. Our hypothesis was that it would be better than the method of system diagrams at identifying operator related hazards. The results from the experiment support this hypothesis.

Sammendrag

I Fordypningsprosjektet, studerte vi metoder for å utføre sikkerhetsanalyser i tidlige stadier av systemutviklingen basert på bruken av ”boilerplates” og ”ontologier”. Med utgangspunkt i dette arbeidet, forslo vi to framgangsmåter for å utføre sikkerhetsanalyser: globale farer ved å bruke ”kan-forårsake” kjeder og menneskelige feilmoduser. Metoden med globale farer fokuserer på å identifisere hendelser i et system som kan forårsake farer som kan påvirke omgivelsene systemet operer i. Metoden med menneskelige feilmoduser innfører generiske feil for mennesker, for å identifisere farer knyttet til operatøren av system.

Vi ønsket å vurdere hvor gode våre foreslåtte metoder var for å identifisere farer i løpet av sikkerhetsanalysen. For å oppnå dette, valgte vi å stille to forskningsspørsmål som skulle besvares i denne avhandlingen.

RQ1: Er det lettere å oppdage mulige trusler i omgivelsene med globale farer og kan-forårsake kjeder?

RQ2: Er det lettere å oppdage mulige farer knyttet til operatøren ved hjelp av menneskelig feilmoduser?

For å besvare forskningsspørsmålene, valgte vi å utføre et eksperiment med studenter der vi brukte de foreslåtte metodene for sikkerhetsanalyse av to systemer. Eksperimentet ga oss en god illustrasjon på hvordan prosedyren ville fungere i en virkelig risikoanalyse.

Resultatene for globale farer med kan-forårsake kjeder indikerer at metoden ikke er tilstrekkelig utviklet for å benyttes til risikoanalyse ennå. Det er fortsatt for mange tvetydigheter med hensyn til hvordan kjedene bør utformes, og tilbakemeldingen fra studentene indikerer at det er vanskelig å lære og bruke denne metoden. Algoritmen må struktureres ytterligere, og det må lages enda bedre dokumentasjon på hvordan den skal benyttes.

Dataene fra eksperimentet indikerer at menneskelige feilmoduser har vist seg å være effektiv til å identifisere farer knyttet til operatøren. Metoden fikk gjennomgående god tilbakemelding fra studentene, og identifiserte mange av farene i testen. Vår hypotese var at menneskelige feilmoduser ville være bedre enn metoden med systemdiagram for å identifisere farer knyttet til operatøren. Resultatene fra eksperimentet understøtter denne hypotesen.

Preface

This master thesis was written as part of the MSc at Norwegian University of Science and Technology (NTNU) for the Department of Computer and Information Science. The thesis is a continuation of the work done in the Specialization Project during the Fall of 2012.

I would like to thank my supervisor professor Tor Stålhane for his help throughout this thesis. He has contributed with his insight and experience in safety analysis, as well as functioning as a invaluable discussion partner.

Trondheim, June 18, 2013

Christian O. Hjorth

Contents

List of Figures

List of Tables

1 Introduction

1.1 Background & motivation

In the Specialization Project (TDT4520) prior to this thesis, we studied how hazard analysis are performed and possible improvements to this process. Our motivation was to reduce the time spent on hazard analysis and making the process easier to perform early in the project. We recognized the importance of keeping the method as simple and efficient as possible, thereby making it possible to perform early when we have limited amount of information and still have the ability to implement possible changes at a low cost.

The approach we choose to accomplish this was the use of templates in the form of boilerplates along with an ontology for describing the environment and system components. By using these methods, we were able to presented two different approaches to improving the hazard analysis; the use of can-cause chains to infer global hazards and human failure modes for identifying operator related hazards. These methods will be described in depth in chapter 2. Since the Specialization Project was purely based on theory, we have in this thesis chosen to create an experiment in order to gather information and feedback on using these methods.

1.2 Goals and research questions

We need to define a research goal to pursue in the master thesis to ensure that the important aspects of the experiment are defined before we start planning and execution. To make it possible to see if we reached our goal, we will also define some research questions to answer with the experiment performed in this thesis.

Main goal

Make it easier to perform early safety analysis

Our main research goal is to identify methods for making it easier to perform early hazard analysis. Performing hazard analysis can be a resource demanding, time consuming and work intensive process requiring a diverse team of people in order to find and describe possible hazards for a safety critical system. If we want to make the process easier, we will have to cut down on the needed resources, time used, people needed or the complexity of the work.

In order to measure the success or failure of our main goal, we will need to formulate some research question. These questions will make it possible to gather quantifiable data from the experiment. They will also be used to formulate hypothesis and create the design of the experiment along with its test cases and questions.

Research question 1

Is it easier to discover possible environment threatening hazards with global hazards and can-cause chains?

One of the proposed methods for hazard analysis in the Specialization Project was the use of can-cause chains to discover global hazards. It was our belief that this would make it possible to identify the events that can cause a chain reaction leading to an environmental hazard. In order to answer this question we will have to create a case and provide the participant performing the hazard analysis with possible events that may occur in the system and environment hazards that may threaten the system. We will then have the participant create chains of events that may lead to an environmental hazard and see if they actually finds it easier finding these with our proposed method.

Research question 2

Is it easier to discover possible operator hazards with Human Failure Modes?

Another proposed method for hazard analysis was the use of human failure modes for discovering hazards and threats related to the operator of the system. By creating failure modes specifically for the operator we believed that these hazards would be made easier to discover. To answer this question we will create a case along with requirements specified with the boilerplate template. For each requirement, the participant will be given a table with the generic failure modes related to the operator.

1.3 Research method

In order to answer the research questions listed, we need to perform an experiment. The structure of the experiment is based on Wohlin [2], where four research methods are presented, shown in table 1.1.

Method	**Description**
The scientific method	The world is observed and a model is built based on the observation, for example, a simulation model.
The engineering method	The current solutions are studied and changes are proposed, and then evaluated.
The empirical method	A model is proposed and evaluated through empirical studies, for example, case studies or experiments.
The analytical method	A formal theory is proposed and then compared with empirical observations.

Table 1.1: Research methods in the field of software engineering

Since we will not be studying an already existing system, neither the scientific nor the engineering method will be relevant for our experiment. Both the analytical and empirical methods are relevant for the experiment we want to perform. They will both produce quantifiable data in an controlled environment. We are mostly interested in creating a model of a real world problem and try to use our suggested method for solving them. This means that we will be using the empirical method to answer our research question, since this method will enable us to control most of the conditions and ensure that the data we collect are reliable.

The experiment to be performed will consist of six parts. First a pre-experiment questionnaire where the participants will fill in information about themselves. Then there will be a tutorial which explains the methods they are about to use and give a simple example of how to use them. The next parts will be two different cases to be solved with different methods, followed by a post-experiment questionnaire for each. More details concerning the experiment will be given in chapter 2.

1.4 Context

This report work is a continuation of the work done in the CESAR (Cost-efficient methods and processes for safety relevant embedded systems) project and at NTNU. It will mainly be based on the work done in the TDT4501 - Specialization Project [3]. CESAR is a European project started in 2009 that focus on software development for reliable embedded systems in four different domains; Avionics & Space, Automotive, Rail and Automation. The aim of the project is to bring significant innovations to the industry in terms of how to improve the development of safety critical embedded systems. It tries to achieve this by addressing

- Requirements engineering in particular through formalization of requirements
- Component based engineering applied to design space exploration comprising multi view/multi criteria architecture trade-offs.

1.5 Thesis structure

The remainder of the thesis is organized into the following parts:

Preliminary studies
The second chapter gives necessary background information for understanding the topics discussed in the later chapters and findings from the Specialization Project. This includes the methods and theories used to create the experiment. Most of the material will be from the Specialization Project [3].

The experiment
The next three chapters discuss how the experiment was planned and executed. Chapter 3 discuss the purpose and focus for the experiment, and how the experiment was designed. Chapter 4 evolves around the execution of the experiment and chapter 5 show the data collected and analysis.

Evaluation and conclusion

Chapter 6 evaluates the data collected and summarize our findings from the experiment. Lastly, chapter 7 will conclude our work and answer the research question stated in chapter 1 and recommend further work to be done on the subject.

2
Preliminary studies

This chapter will give the necessary background information for understanding topics discussed in the later chapters. It will mainly evolve around the work done in the Specialization Project, and give a quick introduction to the material. For a more complete review of the topics discussed, it is advised to refer to the Specialization Project.

2.1 Boilerplate

Usually requirements are stated in natural language - unstructured plain text. This gives the author freedom in how he wants to express and formulate the requirements. On the other hand, it can lead to ambiguous and inconsistent requirements that can be troublesome to maintain. The concept of boilerplates is using templates to formalize and structure the requirements, while still being able to write them in natural language. This creates a good way of standardizing the language for requirements.

The basic building block of a boilerplate is a clause which expresses some aspect of the requirement. Each clause has a clause type that specifies what kind of requirement that is being expressed. Examples of clause types can be *capability, function or constraints*. Further, each clause must have a goal type that indicates whether the goal has been reached. Examples of goal types can be *minimize, maximize, exceed some value, etc.* Boilerplates are created by combining multiple clauses expressing multiple aspects of a single requirement

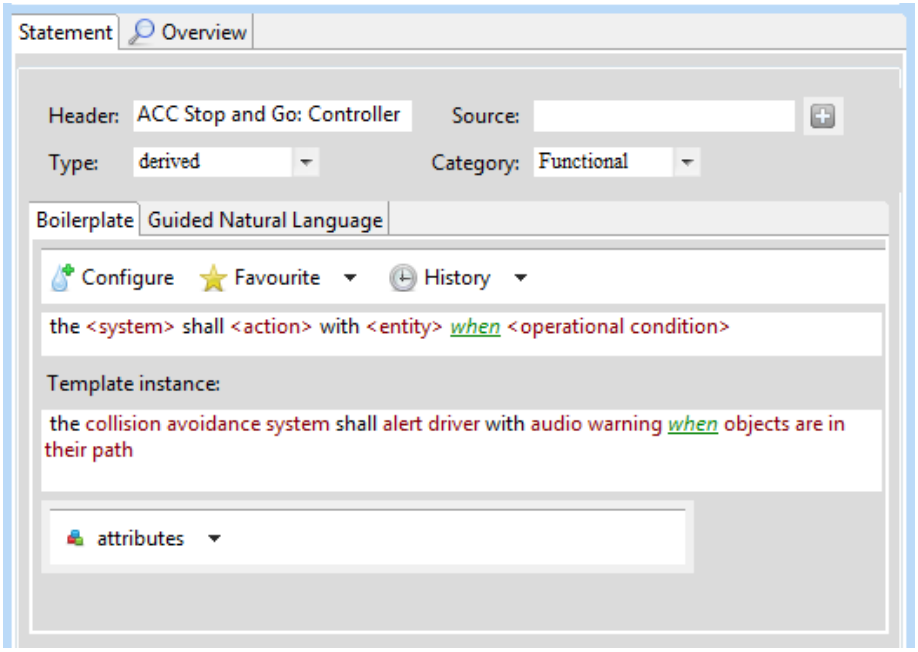


Figure 2.1: Boilerplate template used in the tool GNLQ[1]

Through the work done in the CESAR project[1], the initial set of 32 boilerplates proposed by [4] has been further extended due to the need for more classes of requirements in the CESAR boilerplate repository. This includes classes such as capability, function, timeliness, mode, operational constraint and sustainability. The requirements are formulated using three types of boilerplates:

- **Pre-condition** (prefix) - one or more operational conditions that need to be true in order to move on to the action part of the requirement.
  e.g. ”While <state >...” or ”If <event >...”

- **Active part** (main) - one or more actins perfromed or abilities made available if the pre-condition is true.
  e.g. ”... <system >shall <action >” or ”... <system >shall allow <entity >to be <state >”

- **Action condition** (suffix) - one or more operational conditions or restrictions that applies to the action or the way the action is performed.
  e.g. ”... without <action >” or ”... within <number ><unit >”

[1] http://www.cesarproject.eu/

2.2 Ontologies

Ontology is a shared formal conceptualization of a domain that allows definition of semantic relationships between entities, and inference of knowledge through reasoning at run-time [5]. Simply put, an ontology defines the terms we use to describe and represent some kind of knowledge within a domain. In information science, an ontology can be seen as a dictionary of terms formulated in a canonical syntax and with commonly accepted definitions designed to yield a lexical or taxonomical framework for knowledge representation which can be shared by different information systems communities [6]. Since concepts may have different meanings and use in different domains, we need different ontologies for different domains. We may, for example, need one ontology for the equipment and another for the system domain. Some of the reasons for developing an ontology are:

- Share common understanding of the structure of information among people
- Enable reuse of domain knowledge
- Make the domain assumptions explicit
- Be able to analyze the domain knowledge

An ontology has three parts - a thesaurus, a set of relationships between terms in the thesaurus and rules for making interferences based on these relationships. An ontology is mainly based on standards and glossaries. The problem with creating ontologies is that it involves a lot of manual work, since there is no automatic process available. This work includes ensuring that sentences have clearly marked boundaries, replace information contained in figures, diagrams and tables with sentences and lastly having a domain expert do a review of the ontology before it is put to use [7]. The components of an Ontology usually include individuals, classes, attributes and relations.

2.3 Safety analysis

From the Specialization Project, we have chosen to focus on two methods for performing safety analysis; hazard and operability study (HazOp) and failure modes and effects analysis (FMEA) using generic failure modes.

2.3.1 HazOp

Hazard and operability study (HazOp) is the process of identifying potential hazards in order to plan for, avoid or mitigate the impact they will have. This

is a high-level process that is usually performed early in the project and can be seen as the first step of a risk assessment. The two purposes in identifying hazards can be either to obtain a list of hazards for evaluating using other assessment techniques (failure case selection), or to perform an evaluation of the significance of the hazards and identifying measurements for reducing the risks from them (hazard assessment).

During the Hazard identification phase, criteria for hazards to be evaluated will be established and possible hazards and accidents will be reviewed. The hazards will then be classified into critical and non-critical hazards, where also the non-critical hazards must be well documented in order to demonstrate that they can be safely disregarded.

The failure case selection will typically be performed by executing a hazard and operability study (HazOp). A HazOp study is a structured and systematic examination of a planned or existing process or operation in order to identify and evaluate problems that may represent risks to personnel or equipment, or prevent efficient operation [8]. By performing HazOp you attempt to identify the systematic safety weaknesses in the system design by considering the potential effects of variation of the system parameters beyond the intent of the design. When performing the HazOp process, a team which contains individuals with different background and expertise is assembled. The team then performs the following basic procedure[9]:

1. Select a representation of the system (logical, pictorial, etc)
2. Identify the objects (including subsystems) in the representation
3. Identify the attributes (properties) of those objects
4. Systematically consider the potential effects of extending the values of those attributes, individually or multiply, beyond design intent
5. Record any safety-related consequences from Step 4

The HazOp is focused on specific points of the process or operation, called study nodes. These are usually points in the system where the system interacts with its environment - e.g. user input, or where two or more parts of the system exchange information, - e.g. a network connection. In order to identify deviations from the design intent, a set of guide words are applied to each identified study node. The guide words are used to focus the attention of the participants. The standard set of guide words are presented in table 2.1, taken from [10]. The guide words may, however, not be particularly well matched for software. The solution to this can be to either create new guide words related to software, or give a new software-related meaning to the original guide words [11]. For each guide word, the discussion start with the question: what does this guide word mean in our system and for this study node?

Guide word	Meaning
No or not	Complete negation of the design intent
More	Quantitative increase
Less	Quantitative decrease
As well as	Qualitative modification/increase
Part of	Qualitative modification/decrease
Reverse	Logical opposite of the design intent
Other than	Complete substitution
Early	Relative to the clock time
Late	Relative to the clock time
Before	Relating to order or sequence
After	Relating to order or sequence

Table 2.1: Guide words for the HazOp

In table 2.2 we show a simple example of relating guide words to study nodes. The guide words and where it makes sense to combine them with a study node is marked in the table. After doing this, the HazOp table is created with the guide words and study nodes that can be combined, and the consequences, causes and possible solutions are entered. See table 2.3 for a simple example of a HazOp table.

Guide word	Study node			
	Flow	Pressure	Level	Temperature
No	X			
Less	X	X	X	X
More	X	X	X	X
Part of				
...				

Table 2.2: Correlation between guide-words and process parameters

Guide word	Study node	Consequence	Causes	Possible solution
Less	Flow	Lower pressure than normal	Blockage, defective pump	Regular inspections, Warning through sound
...				

Table 2.3: Example of a simple HazOp table

The strength of the HazOp is that it divides a complex system into smaller and more manageable study nodes and perform a systematic identification of process

deviations, thereby making it a thorough method for identifying system failure modes. The guide words makes the participant consider how the guide words can apply in the context of the function being examined [12]. This also makes it harder to be lazy, because the list is standard, and the omission of a guide word becomes obvious, even to the inexperienced. The weakness of the method is that you cannot find hazards that are not present in one of the participants heads when the process start [11]. It does not generate new knowledge, and is therefore dependent on the participants' experience and knowledge.

2.3.2 FMEA and generic failure modes

A FMEA can be described as a systematic way of identifying failure modes of a system, item or function, and evaluating the effects of the failure modes on the higher level [13]. The objective is to identify the effects of the failure modes, and what can be done to remove or reduce the probability of failure happening by finding countermeasures. FMEA is a bottom up analysis starting from a component-level failure and evaluating its effects. When performing FMEA, one seeks to answer questions like: what could go wrong with the system, how badly might it go wrong and what can be done to prevent it. FMEA has been widely adopted and become the practice in Japanese, American and European manufacturing companies. Since it has been widely adopted in the industry, there exists a lot of documentation, experience and knowledge of how to use it. In [13] the purposes of FMEA is listed as follows:

- Identify the potential failure modes related to the design and process of the system. If problems are found, the design or process should, ideally, be changed
- Find the effects of the failure modes
- Find the root cause of the failure modes
- Prioritize recommended actions to handle the failure modes
- Identify, implement and document the recommended actions

FMEA is performed by creating a worksheet containing rows with fixed column headings. What these columns contain depends on the system, and how much information is of interest. Examples of columns can be; component, failure mode, possible cause, system effect, preventive actions and severity. In the worksheet, note a function or process (component) of the system. Then list ways that the component can fail, and list the effects of the failure. For each failure mode, list one or more causes for the component failing. Then, for each cause, list at least one method of preventive action, and enter the severity of the failure. A small example of a FMEA worksheet is shown in figure 2.4. FMEA is

a bottom-up method, meaning that the analysis start at the low level component, and works its way up towards higher level components in systems. The failure effects of the low level component constitute the failure mode of the upper level components.

Component	**Failure mode**	**Possible cause**	**System effect**	**Preventive actions**	**Severity**
Pressure valve	Too much flow through valve	Internal pressure valve fail to open	Steam boiler rupture	Have two internal pressure valves	High
	Too little flow through valve	Internal pressure valve failed to close	Flooding	Have sensors detecting pressure	High
...					

Table 2.4: Example of a simple FMEA table for a steam boiler

The strong features of the FMEA are that it offers a systematic walk-through of the system components. The method gives us an easy-to-use list of hazards. By studying and "scoring" based on the parameters in the worksheet, a thorough understanding of the failure mechanism and insight in determining effective preventive actions is achieved. This also assists us on prioritizing failure modes for applying resources more effectively. Conversely, FMEA has no standard set of keywords to start off from, leaving it up to the practitioner to decide what to consider. This requires that we have domain knowledge and experience of typical hazards in order to be able to perform it effectively. The size of an FMEA also presents a challenge. The method will produce large amounts of repetitive and redundant data, and the scale of the worksheet will often be several levels deep, and sometimes dozens or hundreds of pages long [13].

Generic failure mode is a failure mode containing a group of more detailed, specific failure modes that all have the same high level manifestation [14]. For example, all failure modes that will lead to a motor stopping can be included in the generic failure mode "motor stops". Generic failure modes are used in several industrial domains, and the failure modes will vary depending on the industry type [7]. Most of what is published on generic failure modes concerns hardware, while a few publications published focus on software.

For hardware components, it is more or less straight forward to find failure modes. These can either be based on experience of same or similar component, or information from the manufacturer. For software components, such information does not exist and failure modes are unknown, since any known errors

Failure mode	Meaning
Omission	Something that should have happened did not happened Something that should have been implemented was not
Commission	Something happened that should not have happened Something was implemented that should not have been
Stuck	Nothing changes for an output signal Nothing changes for the position of a valve Nothing changes for an output parameter
Incorrect	The system gives a wrong value - passive element The system performs a wrong action - active element The system contain a wrong realization of some functionality
Too early	The event happens too early An action is taken too early or too quickly
Too late	The event happens too late An action is taken too late
Loss	No output Wrong output
Erroneous	Wrong output Wrong action

Table 2.5: Failure modes for FMEA

in the code would be corrected and released as an update to the software [13]. Therefore, failure modes for software components are largely dependent on the knowledge of the analyst. A problem with generic failure mode is to choose the right level. Selecting a failure mode of too high level, will result in too low granularity, while choosing a too low level will result in a too long list of possible failures [7].

[15] present a list of generic failure modes. These generic failure modes need to be interpreted by someone with experience in safety analysis, and will work well as a starting point. On the other hand, it can create problems for users without this experience and domain expertize. The failure modes should therefore be made more concrete and specialized according to their use. Table 2.5 list the generic failure mode, along with a short description for each element:

2.3.3 Safety analysis based on components and generic failure mode

As noted in section 2.3, there are multiple methods of performing safety analysis on a system. The methods that have been explored is HazOp and FMEA. These

methods focus on different part of the system, and how to perform the analysis. FMEA is a bottom up analysis starting from a given component view, and works its way forward (up) to evaluate the effects of component failure [16]. The method starts with a component error, then identify the effect of this failure and the error it causes and eventually the consequence that leads to a potential system failure. The HazOp, on the other hand, study the deviation from the design intent. Once a deviation has been identified, the possible causes and consequences of that deviation is explored. If a cause leads to a consequence (see figure 2.2), we can see the different approaches to identifying faults and hazards in the two methods in figure 2.3.

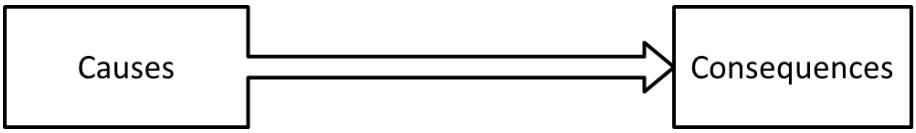


Figure 2.2: Simple causal model

In FMEA we focus on one cause, a component failure, which may lead to multiple consequences. HazOp, on the other hand looks at the deviation from the design, which might both have multiple causes and consequences.

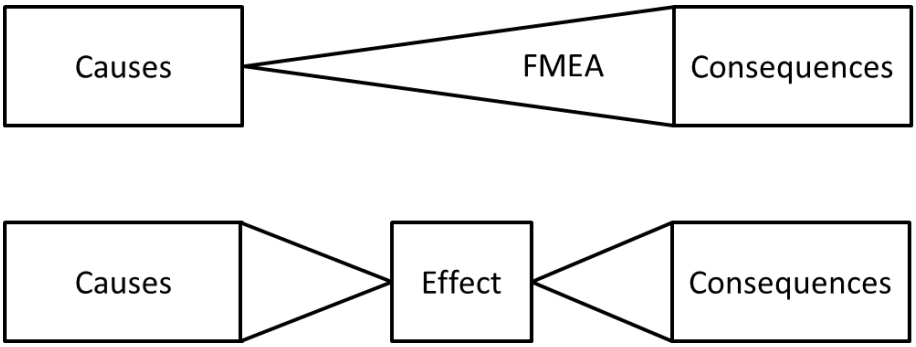


Figure 2.3: Causal model for FMEA and HazOp

For the safety analysis, we want to be able to combine the strengths of these methods into one. In other words, we want to create a FMEA table with guide words or generic failure modes instead of component failure modes. This will create a partial filled in FMEA table with generic failure modes, hereby referred to as a HazId table. The creation of the HazId table can be done automatically based on requirements stated using boilerplates, along with an ontology that contain failure modes for the components. It is important to remember that the use of this HazId table will, in the general case, not add any new information. What it will do is to [7]:

- Make sure that all identified components or functions are included in the analysis

- Save a large amount of effort by giving the analyst a flying start
- Enable developers and customers to do a large part of the HazId themselves, which again will enable us to use available domain knowledge more effectively

There are several reasons for basing the HazId on generic failure modes and an FMEA table. First of all, it will be possible to perform the analysis early in the process, since we can infer from the requirements what kind off components that are needed. Furthermore, it is possible to perform the analysis as soon as we know the services the system will supply by basing the FMEA table on the services and functions off the system. The use off FMEA is already a preferred method for achieving hazard identification early in the product development process.

The generic failure modes presented in section 2.3.2 is the starting point for the analysis. These need to be interpreted and specialized according to their use and domain, which should be done by people with experience in safety analysis. By doing this, the terms will become more concrete and thus work better for non-expert like domain experts and users. The following is a simple example of the safety analysis output from the information by a single requirement R1:

R1: <control system> **shall** <control><water level>
with <feeding pump>

For this example, there has been created a simple ontology with generic failure modes that has been specialized according to their use. We have the following failure modes for the components:

- Sensor
  - No output
  - Wrong output
- Actuators
  - No action
  - Wrong action
- Computer system
  - Omission - function not present when required
  - Commission - function present when not required
  - Incorrect
  - Too late

There are two parts to this requirement; a software component called ”control system” and a physical component called ”feeding pump”. This means that the physical component should get the failure modes related to actuators, while the software component should get the failure modes related to computer system.

With this set up, the partially filled HazId table seen in table 2.6 will be generated. The rest will have to be filled in by the analyst. It should be noted that this is only a simple example of the HazId table, it can be expanded with more columns as seen needed.

Req. id	**Element**	**Failure mode**	**Cause**	**Effect**	
				System Local	**Environment Global**
R1	control system	omission			
		commission			
		incorrect			
		too late			
	pump	no action			
		wrong action			

Table 2.6: Example of a HazId table with generic failure modes

2.4 Results from Specialization Project

In this section we will present new methods of safety analysis based on the material shown in the previous sections. This represent the work we did during the Specialization Project.

2.4.1 Global hazards

In the previous sections, the creation of a partial and complete HazId table based on boilerplates and equipment ontology was presented. We want to investigate how we can build upon this, to create a new method to identify global hazards. The procedure for identifying global accidents is the cause-consequence chain for the equipment in a given environment. This is a simple, transitive chain e.g.:

```
A can cause B, B can cause C
=> A can cause C.
```

With this procedure as a starting point, I have identified several areas of interests to explore further. These are the cause-effect algorithm, how to move between an domain ontology and environmental ontology, making the hazard boilerplates more precise as well as including generic operator failure modes.

2.4.2 Cause-effect algorithm

We want to find the end-effects of a cause-consequence chain. To do this, we need an algorithm that describes a structured way of doing this. We want to relate a failure mode, with a local effect, to a global effect. A local effect is an effect that is present within the system under control, while a global effect is related to the environment the system is operating within. The starting point is that a failure mode can cause an effect on the local system.

```
<Failure mode> can cause <effect>
```

The effect itself may then cause new effects on the local system, thereby possibly starting a chain of effects.

```
<effect> can cause <effect>
```

Eventually we will come to the point where the chain of effects can't be further deduced. When reaching this point, we will have to decide whether this is a boundary effect or not. A boundary effect is referring to an effect that can have an impact on the environment and may lead to a global effect (accident), i.e. it is located at the boundaries of our system under control. Figure 2.4 shows how an initiating event (failure mode) can lead to different effects. Some of the effects stops within the control system, meaning that the <effect> won't have an impact on the environment the system is located in. The boundary effects lead out of the local system and thereby have the possibility to affect the environment and causing a global accident. In order to decide if an <effect> is an boundary effect, we need to have some predefined notion of what the boundary effects for our system are. We will discuss this in depth in the next section.

A possible challenge is that we assume that all of the definitions, components, events and effects are using the same names. For example, referring to the <heater> as boiler, heating system or something else, will break the chain of events, making it complicated to deduce global accidents from initiating events.

```
A => B, B' => C
- B and B' are synonyms
```

The solution is to use the ontology to find synonyms and the relationships between them. This will ensure that even though we use different names for a heater, it will not break the algorithm if we use the ontology to check the relationships between them

Below is the algorithm discussed above stated in a more structured and formal way

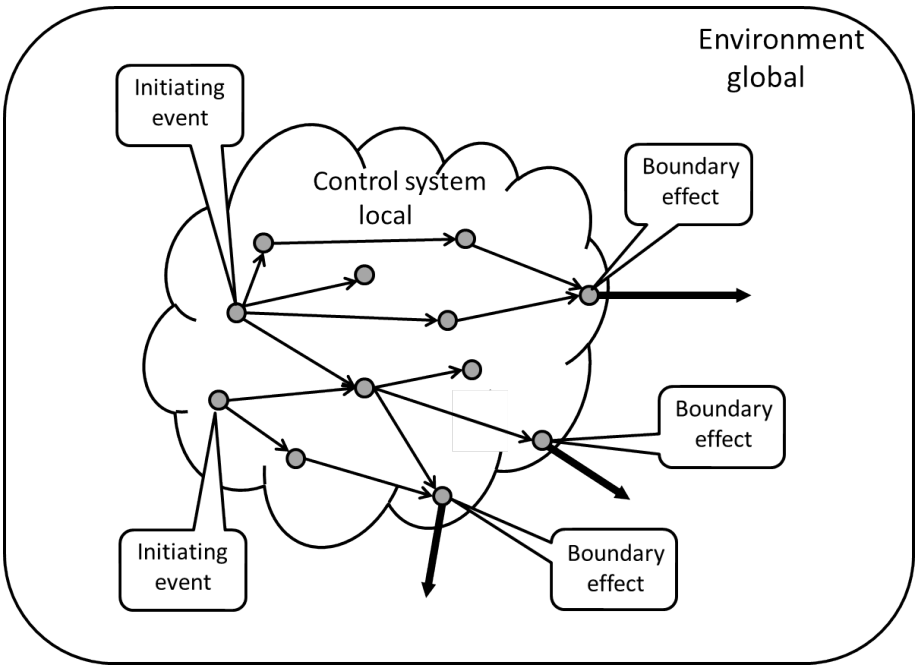


Figure 2.4: Chain of events

1. First select a representation, R, of the full system, S.
   - R will contains all components identified from the requirements
2. Use the ontology to list (L)
   - All objects in R
   - All the properties to said objects
   - The relationships between the objects
3. For all hazardous events, E, in system S which can be expressed in the component ontology (L)
   - Perform cause-effect analysis on E
   - When elements in E can not be deduced further - check if E $\in$ <boundary effects>
   - if E $\in$ <boundary effects>, retrieve global hazard from environment ontology

A small example of an event leading to a global accident is shown in figure 2.5. The generic failure mode ”sensor error” is the initiating event, with two possible effects - Wrong temperature or no sensor data. We see that wrong temperature have a chain of effects that lead towards a global accident - Fire. ”Too high

temperature" is in other words a boundary effect, while "No sensor data" and "Too low temperature" are limited to having only local effects.

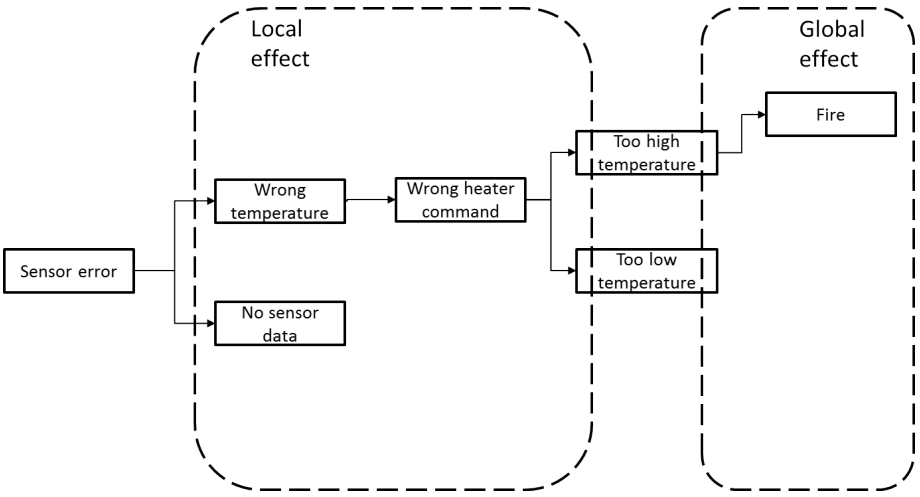


Figure 2.5: Example of a local failure causing a global failure

2.4.3 Human failure modes

The creation of a partial HazId table is based on generic failure modes for the components in the system. This makes sure that all the identified components are included in the safety analysis. What has not been included is the operator that will use the system. This is a rather important part of the analysis since humans some times make mistakes. If we were to add the operator, we would need to identify some generic failure modes to be associated with them. A starting set of failure modes could be:

- Forget - forget to perform an operation
- Wrong - wrong operation performed
- Too late - operation performed too late

In addition to the generic failure modes, we would need a separate ontology for the operator. This ontology will be used to control that each entity that is linked to an operator has a set of generic failure modes for an operator.

It should be rather straightforward to add human failure modes when performing the safety analysis based on generic failure modes. With this new knowledge we would be able to identify barriers against human error early in the project. Below is an example of the generated HazId table with generic failure modes for an operator. The table is based on the following requirement:

The <operator > **shall be able to** <start the control system> **to** <pump out water>

Req. id	Element	Failure mode	Cause	Effect	
				System Local	Environment Global
R1	control system	omission			
		commission			
		incorrect			
		too late			
	pump	no action			
		wrong action			
	operator	forget			
		wrong			
		too late			

Table 2.7: HazId table with generic failure modes for an operator

3

Definiton and planning

In this chapter the foundation of the experiment is determined, describing why the experiment is conducted. Then the plan for the experiment is described. This chapter follows the structure described for an experiment in [2]

3.1 Definition

Before we can plan how to perform our experiment, we need to analyze whether the problem at hand is suitable for an empirical study. Our experiment is motivated by a need for making hazard analysis as easy and cheap as possible to perform, while also making sure it is complete with regards to detecting all hazards that are relevant.

3.1.1 Object of study

The object of study is the entity that we will be studying in our experiment. For this experiment we will study the process of early hazard analysis on a system. We will be focusing on two methods for performing the hazard analysis, Global hazard by using can cause chains and human failure modes.

3.1.2 Purpose

The purpose for our experiment is to evaluate what can be gained from using the above mentioned methods when performing hazards analysis. We want to see whether these methods will be able to make it easier to perform a hazard analysis on a system, than doing it without these aids.

3.1.3 Quality focus

For our experiment we will have two methods that we will test. Each of these methods will have their own quality focus. For global hazards using can-cause chains we haven chosen to focus on the participants ability to find chains that leads to global hazards in the environment the system operates in. For human failure modes we will be focusing on how many operator related problems the participant are able to find a set of requirements for a system.

3.2 Context selection

The context sets the environment in which we will be executing our experiment. There are four different dimension which will describe how our experiment will be done:

- Off-line vs. on-line
- Student vs. professional
- Toy vs. problems
- Specific vs. general

En experiment that is run on-line means that the investigation is executed in the field under normal conditions, in other word we would need a real project. Running our experiment on-line will produce results with better validity, but at a greater cost and risks connected to it [2]. Since our research is in a early phase with a lot of uncertainties, it will be better to perform our experiment off-line, where we are able to control the environment and variables better. Given that our findings are significant, it would be interesting to perform an on-line experiment to validate that our findings are the same in a real world situation.

The test subjects for our experiments will be students. The reason for this is more or less the same for performing it off-line, students are cheaper than professionals and there will be less risks involved with using them. Further, we will have less problems with getting a sizable group of students and scheduling

a time, than with professionals. The downside of using students are the lack of relevant experience with the subject, and that we may end up with a homogeneous group, making the results less applicable to the general safety analysis domain. However, based on results from earlier experiments with students, it seems that a lot of safety analysis is down to common sense. As explained in later chapters, we are more interested in the distribution of the answers, than the quality of them.

The experiment will consist of two cases, a steam boiler and a train scheduling system. Both of these cases has been used in earlier experiments and are therefor well documented and understood from our perspective. They are both simplified system from larger real world system, which means the requirements and diagrams follow industry standards. Given this, we can state that we are dealing with ”toy system”, meaning that it is not real problems, but problems of toy size. Given the time constraint and lack of domain knowledge from the participants along with cost constraints, this is to be expected.

Whether our findings are specific for the current context or are valid to the general domain depends on the cases and participants we have chosen. Even tough both cases are problems of toy size, they present different complexities, focus and scale. We believe that this ensure that they will be generally applicable.

3.3 Hypothesis formulation

We need to formally state what we intend to evaluate in the experiment. Our hypotheses will be based on our two research questions stated in section 1.2. Since we have chosen to divide the experiment into two parts with different methods and cases, we have also found it necessary to have two different approaches to the hypothesis.

For global hazards using can-cause chains, we have chosen to do an exploratory experiment. Exploratory research is a form of research conducted for a problem that has not been clearly defined [17]. The main reason for doing this is the lack of earlier research around the subject and available data to compare against. The topics we will research are as follows:

Research question 1

Is it easier to discover possible environment threatening hazards with global hazards and can-cause chains?

Since we don’t have a hypothesis formulated for this question, we have instead

identified three areas of interest that we will analyze. We believe that this will be a good way to make an evaluation of the method.

Number of chains that are discovered per possible global hazard.
In our experiment we will describe several global hazards, and present a list of events in the system (see section 3.6). The test subjects will try to find all possible event chains that will lead to a global hazard.

Frequency of the type of chains identified.
There will be several chains that can lead to a global hazard. We will see which of these chains that are identified by the test subjects.

The number of correct chains.
There will be a set of correct chains for each global hazard. We want to see how many of the chains that the test subject finds that are correct.

Research question 2

Is it easier to discover possible operator hazards with human failure Modes?

We want to compare the method of human failure modes against another method for hazard identification- using system diagrams. The data for the method of using system diagrams will be from an earlier performed experiment with the same case. The method of system diagrams is explained in section 5.6. Our hypothesis formulation will then be:

H0 The number of operator related hazards is the same or fewer with human failure modes compared to the method of using system diagrams

H1 The number of operator related hazards will be higher with human failure modes than with the method of using system diagrams

3.4 Variables selection

The independent variables are the method of hazard identification, which in our thesis is the method of can-cause chains using global hazards and human failure modes method. Other independent variables are the test subjects earlier experience, education and working experience.

The dependent variables will be different for the two cases. For case 1, regarding global hazards with can-cause chains, we will be registering the chains that are discovered and how many of these that is identified per global hazard. For case 2, regarding human failure modes, we will measure how many and what kind of operator and non-operator related problems that are discovered.

3.5 Selection of subjects

The test subjects for the experiment are chosen based on convenience. We have chosen to recruit second year students from the Computer Science studies at NTNU. The volunteers for the experiment will be compensated with NOK 250 towards their third year class trip. The trip committee receives the money when the participants have signed off after turning in the experiment material. There were expected to be around 50 students participating.

The participants will most likely not have much experience with safety analysis or requirements engineering. They were currently taking the course TDT4140 - Software Engineering - which will give them some relevant experience in regards to the requirements engineering. However, they were only half-way through the course. Based on this, we can say that we expected that the test subjects have little relevant experience. Given that we cases are designed to be simple, we believe that this will not be a serious problem, but should, and have been taken into consideration when designing the problems and cases.

3.6 Experiment design

In order to be able to draw meaningful conclusions, we must ensure that the design of the experiment fit our goals and plans stated earlier. We will here describe how the tests are organized and run. The complete experiment is described in Appendix A.1. The experiment is divided into four main parts; pre-experiment questions, first case (test), second case (test) and lastly a post-experiment questionnaire.

In the pre-questionnaire the participants are first given a short introduction consisting of the purpose of the experiment and a table of contents with an estimated time used for each part. After this they are asked to give general background information about themselves including previous experience, relevant courses taken and number of completed semesters of study. We also asked for a ”person-ID”, which will be used to contact the person regarding the results, if necessary. Lastly, we provided a short tutorial on safety analysis using can-cause chains and human failure modes along with examples.

The next part of the experiment is the first case to be solved using can-cause chain on a steam boiler system. The goal of the steam boiler is to deliver steam while maintaining the water level and pressure within set boundaries. There are two control loops, one adjusting the water level and one adjusting the heat and pressure. An automated system tries to keep the water level and pressure in the tank constant. If necessary, an operator can manually override either of the two control loops. See figure 3.1 for a complete system diagram.

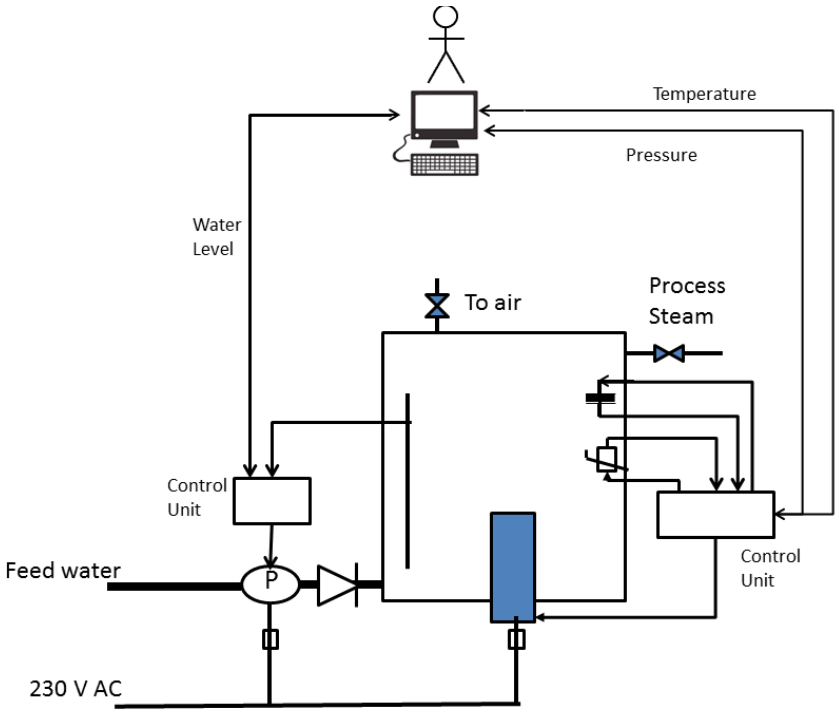


Figure 3.1: System diagram of steam boiler

The test participant will be given a list of possible events that can occur in the system. These events are based of generic failure modes for the components that would occur in a Hazard identification table discussed in section 2.3.3. Examples of these are sensor errors and incorrect behaviour of the control unit. They will also be given a list of hazards that will be a threat to the environment the system operates in (global hazards). Examples of these threats are Explosion and Fire. The test subject is then asked to identify all events that may lead to a global hazard. Since the events in the list are based of components failure modes, it is not a complete list of events for the system. This will make the task more true to a real system, since we rarely will have such a list available.

The second case is a train control system to be solved using human failure modes. The track is built up of multiple zones, and only one train may be in a zone at the same time. The train is operated by a train engineer who can regulate its speed. He receives scheduling information from the control system, and can send updates about his own schedule. An operator is stationed in the control room and oversees all schedules and can manually update or change them if necessary. If the control system fails, there is a mobile network i.e. cell phones that will be used in backup. See figure 3.2 for the complete system diagram.

The test participant will be given a set of system requirements written in boiler-

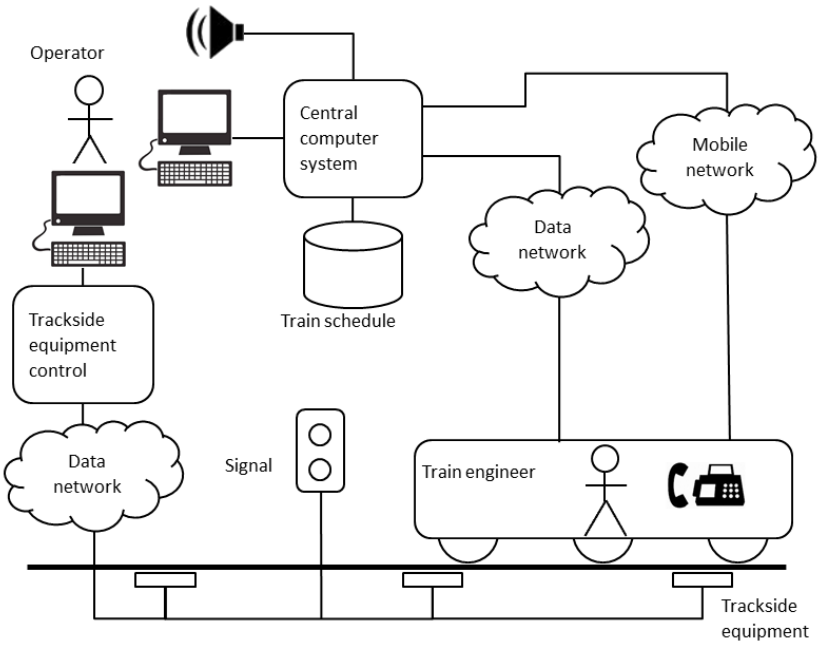


Figure 3.2: System diagram of train control system

plates. One set is for the operator of the control room, while the other is for the train engineer. All of the requirements are related to a human interacting with the system. They are then asked to fill in the effects of the failure modes in a hazards identification table. This table contains the requirement identification, along with three generic failure modes: ”forget”, ”wrong” and ”too late”.

The last part of the experiment is the post-experiment questionnaire. This should be completed after each case and asks for the test subject to evaluate the methods he just used as well as his own performance.

A possible problem with having them perform two different methods after each other is that they can learn from the first case and thereby influence the result of the second case. In order to reduce that threat, we chose to create two versions of the experiments where the cases are alternated. We will then distribute the participants in two groups that are as equal as possible. We should then be able to see any possible learning effects by comparing each case result from the different groups.

3.7 Instrumentation

Before the experiment starts, the test subjects will be given a 10 minutes oral presentation of the experiment. This will include a PowerPoint presentation and will present the two approaches to safety analysis and the test cases to be solved.

The experiment will be printed out on paper and handed out after the oral presentation. As stated in section 3.6, the experiment itself will contain a short introduction to the safety analysis to be performed.

4

Experiment operation

4.1 Preparation

The experiment was executed on Wednesday 17th of April at 14:00. The expected time used for the experiment was 2 hours. The date and time of the experiment had to be scheduled taking into consideration the test participants own schedule in regards to different courses time slots. The location was an auditorium which can seat up to 250 persons and has a projector available. This enabled us to be sure that we had enough seating available and that we could use PowerPoint to show an introduction to the experiment and cases to be solved.

Based on communication with the class representative, we expected approximately 60 students to show. We printed out 70 examples of the two different versions of the experiment, 35 for each, respectively named Experiment 01a and 01b.

4.2 Execution

The attendance was at a total of 59 students, all from the Computer Science studies at NTNU. We ensured that the students were evenly distributed throughout the room to prevent them from influencing each other when performing the experiment. The participants sat at alternating rows, with a minimum of two seats in between each participant.

After the initial distribution of the students throughout the auditorium, the students were given a short 10 minutes oral introduction with PowerPoint slides. The presentation started with an introduction to safety analysis. After this, the main features and intents behind the two hazard identification methods global hazards and human failure modes were presented. Lastly we showed the system diagrams, seen in section 3.6 and explained how the system was designed. After the presentation, we asked if anyone had questions before starting the experiment, which no one had.

The two versions of the experiments were handed out to the students sitting on alternating rows, to further ensure that none of the students would influence each other answers. There were 30 student taking the 01a version, and 29 with the 01b version. From here on, the students was given 1.5 hour to answer the questions and tasks in the experiment. Throughout the time of the experiment, they were able to ask questions . I was stationed at the front of the auditorium at all times.

There were only one question during the exercise, which was a student asking for a new copy of the experiment, due to a wish for writing the answers more understandable. The first participant to deliver did so after a little more than 1 hour, while the last two students delivered after about 2 hours. After they had delivered the experiment, they had to sign the participant list, which would be used as proof that they had performed the experiment, and therefore should receive their payment.

4.3 Data validation

There were 59 students participating in the experiment: 30 in group 1, which was using the 01a version and 29 in group 2 using the 01b version. A possible source of error was participants misunderstanding the text given for the test cases and participants not understanding how to use the methods.

Most of the students used 1.5 hours in total to answer the experiment. As stated above, the first participant delivered the survey after roughly 1 hour. This indicates that the time table we set up and stated in the introduction of the experiment was well estimated. Further, it shows that the participants tried to do their best when answering the task and questions at hand and to do a good effort, even though they were able to leave whenever they wanted.

After an initial review of the collected data, I found two data sets that had to be removed. One student had commented that he mistakenly thought that ”train engineer” and ”operator” meant the same thing. His data from the train control system was therefore removed. Another student had not answered any of the

tasks except for the pre- and post-questionnaire, giving both methods very poor assessment. Given that he didn't ask for any help during the experiment or seem to have tried to answer the tasks, all his data was removed. Aside from these two cases, all of the answers seemed to be valid.

5 Analysis and interpretation

We will in this chapter present the data we collected from the experiment, and how these data were handled and analyzed in order to extract information from them.

5.1 Descriptive statistics

The data we collected from the experiment came in many forms. In the pre- and post-experiments questionnaire we asked the test subject to use a Likert scale to grade their agreement or disagreement with a set of statements. This makes these data easily quantifiable without any further analysis from our side.

The data from the train control system, where we analyzed the human failure modes method where not directly quantifiable, and had a wide range of answers. These answers would have to be interpreted, analyzed and, to some extent, quantified in order to be usable in comparing the method against the results from the experiment with diagrams.

The data from the steam boiler system, where we analyzed can-cause chains, was somewhat more structured, since the test subjects had a limited number of choices that they could use. Still, these data would also have to be interpreted in order to make them quantifiable.

5.1.1 Combining the result

As was mentioned in section 3.6, we divided the test participants into two groups. Each group received a different version of the experiment. The two versions differed in the order of the experiment cases. Group 1 had the steam boiler as the first case, while group 2 had the train control system first. Apart from this, the versions were identical. The reason for doing this was to prevent the two methods from influencing each other's results. After having analyzed and categorized the data we compared them to see if there was any observable learning effect. We used a combination of a t-test and generally comparing the results from the two groups to decide whether we could combine the two groups into one. Since the students were from the same "group" and performing the same case, we used a paired t-test, which is used when data has been measured at two time points.

	Global hazards		**Human failure modes**	
	paired t-test	t-test	paired t-test	t-test
P-value	0,86	0,44	0,66	0,33

Table 5.1: Result from paired t-test on group 01a and 01b

Our conclusion was that there was no indication that the ordering had any significant impact on the results. We have therefore chosen to combine the two group's data into one dataset. All of the following data presented will be from the combined set of data from both groups.

5.2 Previous experience

Before solving the experiment test cases, the participants were asked to give information about them-selves on previous experience.

We asked the participants how many number of semesters they had completed after high school, as seen in figure 5.1. 85% answered 3-4 semesters while 12% answered more than 5 semesters. It is rather peculiar is that 2 persons answered having completed less than 2 semesters. We believe this is due to the student misunderstanding the question and rather answering number of years completed.

When asked for their experience with IT work, 78% answered that they had no experience, 19% had a limited experience and 3% with a lot of experience. This indicates that the participants will probably not have had the possibility to learn any relevant information or experience in regards to the safety methods they

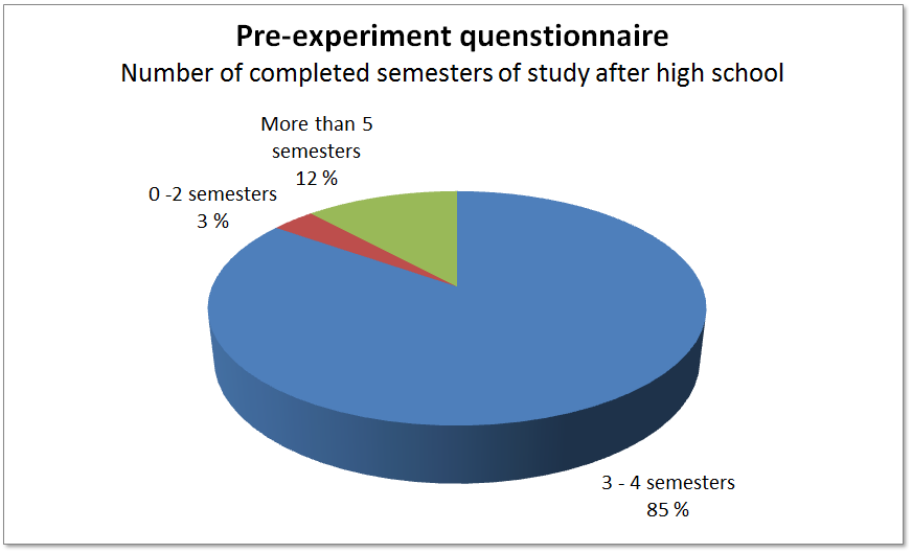


Figure 5.1: Number of completed semesters

will use to solve the test cases. Figure 5.2 shows the distribution of working experience.

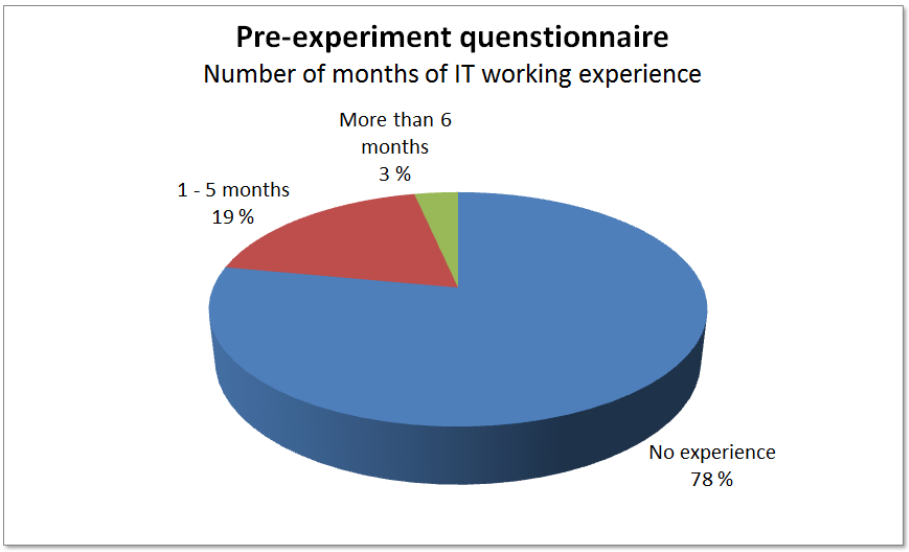


Figure 5.2: Relevant work experience

In regards to relevant experience from courses at NTNU, we asked if they had taken the course TDT4242 - Requirements and Testing, or any other similar courses. As can be seen in table 5.2 none of the students had taken the mentioned course, while four participants answered that they had taken some other relevant course. All of these had answered TDT4140 - System Development, which teaches requirements specification. Since this is a course mandatory to

the Computer Science program in the third semester, it is reasonable to assume all of the students have this course.

Statement	Yes	No	Other
Have you had TDT4242 or similar courses?	0,0%	93,2%	6,8%

Table 5.2: What kind of relevant courses the participants has completed

Lastly, we asked if they had any kind of experience with safety analysis, reliability analysis or boilerplates. Table 5.3 show the same trend as the other answers, none of the students had any relevant experience in regards to the methods to be performed in the experiment. More than 96% indicated that they did not have any relevant experience with safety or reliability analysis, while 85% stated that they had no experience with boilerplates. We see that some of the students are neutral or disagree with the statement in regards to boilerplates. We believe that they may confuse requirement boilerplates, with boilerplates that are used as a template in programming, often called boilerplate code. Boilerplate in computer programming is a section of code that has been included in many places with little to no alteration [18]. It is, however, possible that they have experience with boilerplates from either an earlier experiment or a course they have taken at NTNU, but the vast majority still had no experience.

Statement	Agree strongly	Agree	Neutral	Disagree	Disagree Strongly
No previous experience with safety analysis	72.9%	23.7%	3.4%	0.0 %	0.0%
No previous experience with reliability analysis	74.6%	23.7%	1.7%	0.0 %	0.0%
No previous experience with boilerplates	69.5%	15.3%	13.6%	0.0 %	1.7%

Table 5.3: Previous experience with safety and reliability analysis and boilerplates

5.3 Finding can-cause chains

For case 1, the test participants were asked to find all events that may lead to an environmental catastrophe, in other words a global hazard. It was quickly discovered that it would be difficult to categorize the cause-effect chains because of the many ways to arrange the events, and optional routes that may be taken. What we mean with optional routes is that many of the chains may include

sub can-cause chains, or optional events, leading to an hazardous event (global hazard). This means that it is not a necessity that an event, E, happens to produce the chain leading to the global hazard, H. But the event will not hinder or make the chain invalid either. This makes it hard to categorize the chains, without getting too many categories with minor differences.

We did not want to make a separate category for each such chain due to the amount of work and time it would be to set up and later- categorize the answers. It would also make the data harder to analyze. Therefore, we chose to create diagrams with chains leading to possible hazards, based on the system diagram. After this, we used the events listed in the experiment task (E01 - E14) and found the initiating events for all of the global hazards. The diagrams can be seen in Appendix A.2, with the document describing our approach in Appendix A.3.The initiating events we found per global hazards is shown in table 5.4.

Global Hazard	**Initiating event**
H01: Explosion	E01 - Temperature sensor fails E02 - Pressure sensor fails E06 - Thermostat fails E13/E14 - Incorrect behaviour from CU E08/E11 - Operator reacts wrongly
H02: Fire	E01 - Temperature sensor fails E02 - Pressure sensor fails E03 - Water level sensor fails E06 - Thermostat fails E13/E14 - Incorrect behaviour from CU E08/E11 - Operator reacts wrongly
H03: Leak from water pipe	E01 - Temperature sensor fails E02 - Pressure sensor fails E03 - Water level sensor fails E06 - Thermostat fails E13/E14 - Incorrect behaviour from CU E08/E11 - Operator reacts wrongly
H04: Leak from steam pipe	E01 - Temperature sensor fails E02 - Pressure sensor fails E03 - Water level sensor fails E06 - Thermostat fails E13/E14 - Incorrect behaviour from CU E08/E11 - Operator reacts wrongly

Table 5.4: Initiating events per global hazard

We have chosen to categorize the answer based on the initiating event for each global hazard. Each answer has then been compared to the diagrams in Ap-

pendix A.2, and if it corresponds to one of the defined chains, it has been marked as a correct answer. For example, a valid can-cause chain for H01: Explosion with an initiating event E01 would be:

```
E01 - Temp. sensor fails
   E01 can cause E04 - Heater generate to much heat
     E04 can cause E12 - High pressure in tank
       E12 can cause H01: Explosion
```

The above example follows the highlighted route in figure 5.3. The events in the diagrams and the events presented in the steam boiler case do not agree in all aspects. For example, the diagrams has more events that are better specified than those we have made available for the test subjects in the experiment. The main reason for this, is that the events in the experiment was based upon possible generic failure modes for the equipment is the system, while our diagrams are based upon events happening in the system. However, we feel that this was the best approach to categorize the data and make them quantifiable.

The category E08/E11 Operator reacts wrongly is a somewhat special case. This event is not actually an initiating event, but a hazardous reaction on an initiating event by the operator. This will typically occur when an alarm goes off, like when a sensor starts reporting wrong values. Lastly, the event E13 and E14 have been combined to one event related to a control unit failing.

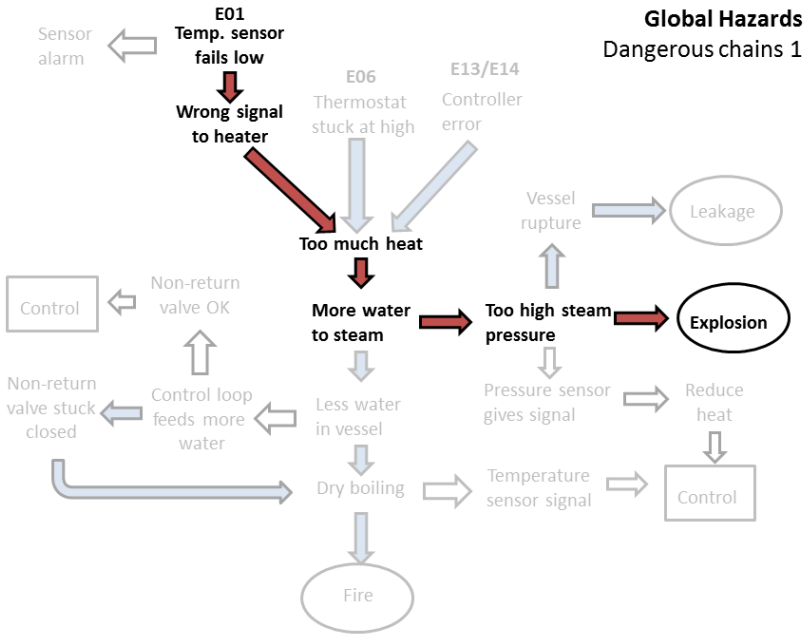


Figure 5.3: Initiating event E01 leading to an explosion

We see in figure 5.4 that the most common chain found for H01: Explosion was

”Pressure sensor failing”, which was correctly identified by 70% of the test subjects. A close second was ”Operator reacts wrongly”, which 66% identified. On the other hand, ”Thermostat fails” was only found by 20% of the test subjects. Overall, it seems that the frequency of the events the test subjects found for H01 was rater evenly distributed

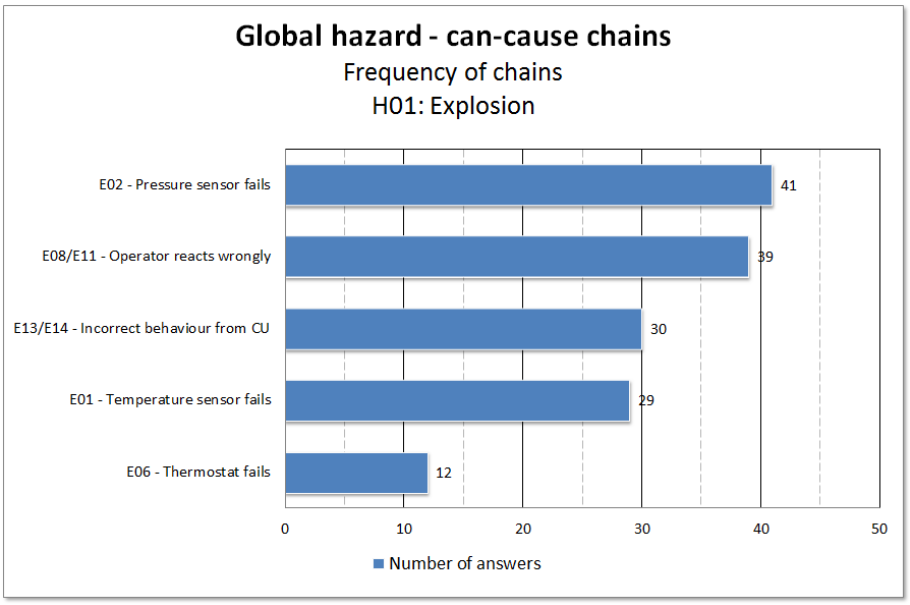


Figure 5.4: Can-cause chains leading to an explosion

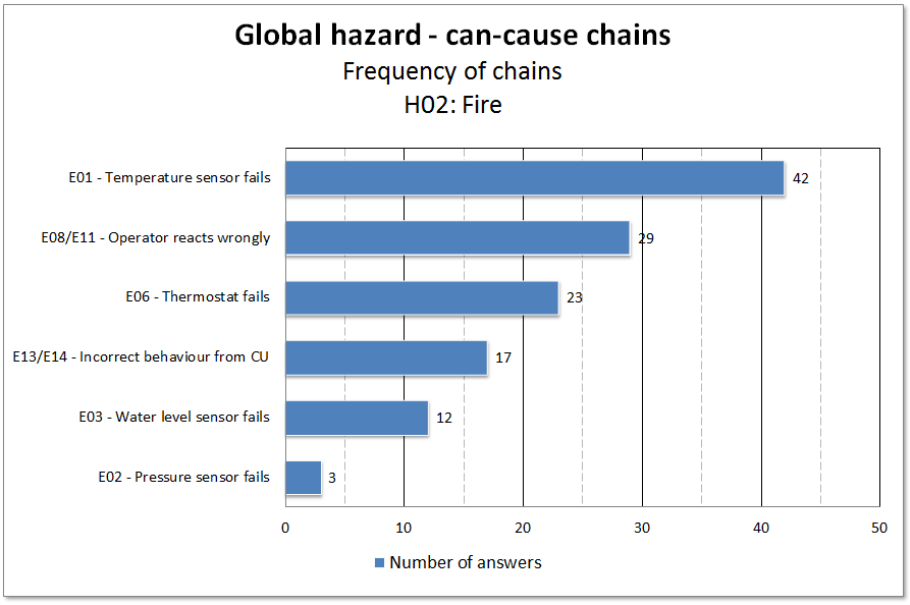


Figure 5.5: Can-cause chains leading to a fire

In figure 5.5, we see that ”Temperature sensor fails” was the most common

initiating event found for all of the global hazards. 42 persons -72% of the test subjects - correctly identified it. ”Pressure sensor fails” was only identified by 5% of the test subjects. We see that the frequency of the chains found for H02 have a much higher variance than H01.

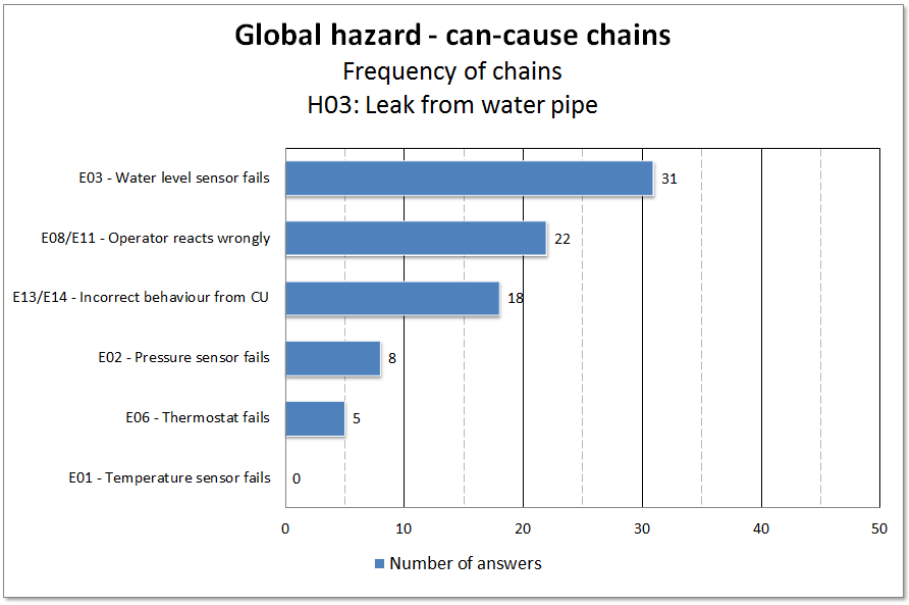


Figure 5.6: Can-cause chains leading to water leakage

For H03, figure 5.6, and H04, figure 5.7, shows that both of them have a low number of chains identified. For H03, the most common initiating event discovered was E03 - identified by 52% of the test subjects. For H04, the most common event was E02 - identified by 45% of the test subjects for H04. Further, we see that none of the participants was able to identify the chain with initiating event E01 for H03.

The last data we have collected for the can-cause chains are the average number of correct and incorrect chains found for each global hazard, seen in figure 5.8. We see a trend towards finding most chains as well as correct chains for the first global hazards. We also note that for H03, we have the second lowest number of identified chains as well as having the most incorrect, or invalid, chains identified by all of the global hazards. In total, the test subjects found on average 2.3 chains with 0.4 being incorrectly formulated

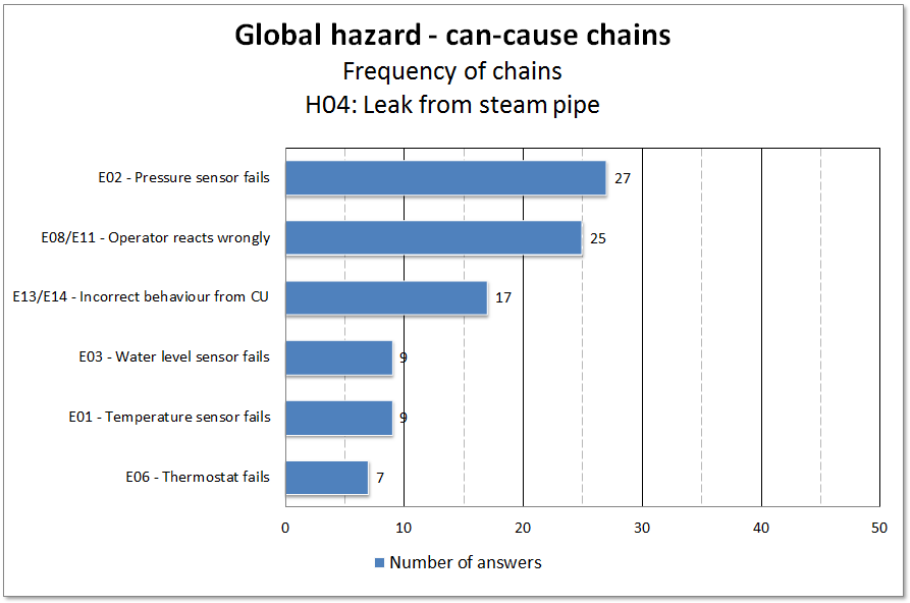


Figure 5.7: Can-cause chains leading to steam leakage

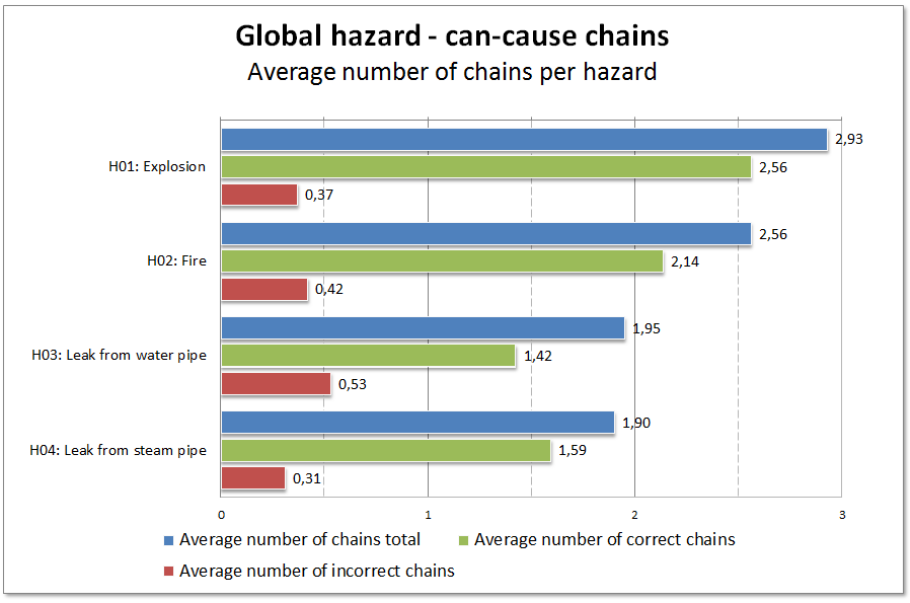


Figure 5.8: Different types of chains created per global hazard

5.4 Identifying operator problems

The train control system has been used in several student experiments at NTNU. Because of this, we already have a list which should cover all problems that can occur in this system. For our experiment, we have chosen to reduce the set of problems, because certain parts of the problems are not relevant for our system. The list of problems can be seen in Appendix A.4. The categories we have chosen to remove are related to maintenance personnel. We have also chosen to remove the problems OP301 and EP301 - wrong maintenance scheduling. Since we have not mentioned maintenance anywhere in the requirements in the experiment, we feel that OP302 and EP302 - wrong train scheduling - are good enough for describing the scheduling problems. Table 5.5 show the top-level problems.

Top-level problem	**Frequency range**
OP-100 Incoming messages	0.00 - 0.22
OP-200 Operator action	0.07-0.92
OP-300 Operator scheduling	0.10 - 0.85
OP-400 Operator equipment problems	0.00
OP-500 Operator knowledge	0.0 - 0.25
OP-600 Operator overload, e.g. due to panic	0.00
EP-100 Incoming messages	0.00 - 0.39
EP-200 Operator action	0.02 - 0.85
EP-300 Operator scheduling	0.78
EP-400 Operator equipment problems	0.00
EP-500 Operator knowledge	0.00 - 0.27
EP-600 Operator overload, e.g. due to panic	0.02
CS-100 Does not save or deletes info	0.07
CS-200 Demands a wrong action	0.00
CS-300 Reacts wrongly to command or not at all	0.17
CS-400 Shows wrong info, including false alarms	0.25
CS-500 Unavailable – e.g. due to network problems	0.03
CS-600 Shows no info or no alarms	0.05
CS-700 Other software errors	0.03
TC-100 Technical problems with telecom equipment	0.02
TC-200 Bad signal coverage or poor radio signal	0.00
TC-300 Busy line	0.00
TC-400 Other technical communication problem	0.02
TE-100 Wrong signal set	0.02
TE-200 Signal equipment fails	0.00

Table 5.5: Problem categories for train control system

The problems focus on four parts of the train control system, two related to human interaction and two related to the computer system and communication. The OP-problems are focused on the operator, who schedules the trains and updates the database. The EP-problems are related to the train engineer, whose tasks include adjusting the train speed and updating his status to the operator.

In order to see the most commonly identified problems, we have taken all problems which more than 30% of the test subjects was able to find. This leaves us with 15 problems, which is shown in figure 5.9. First of all, we see that all of the problems identified are either related to the operator (OP) or the train engineer (EP). The most common problems for the operator was ”Forget to perform action” (0.92), ”Responds too late” (0.90), ”Enters wrong info” (0.85) and ”Wrong train scheduling” (0.85). The most common problems for the train engineer is almost identical, with ”Forget to perform action” (0.85) and ”Responds too late” (0.85) being most popular. The small difference is that ”Enters wrong info” (0.73) is mentioned three times less than ”Wrong train schedule” (0.76).

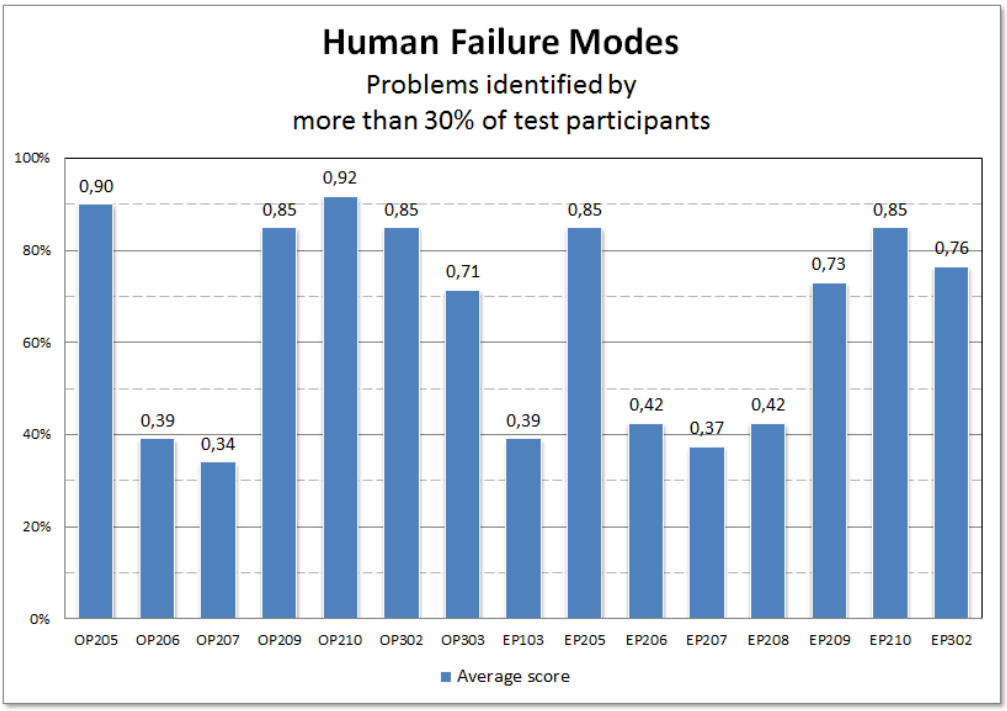


Figure 5.9: Problems identified by more than 30% of test participants

The most commonly identified problems are similar to the generic failure modes presented by the hazards identification table; ”forget”, ”wrong” and ”too late”. In addition, the problems related to a wrong train schedule are also an often identified problem. This is to be expected, since seven out of ten requirements are related to the train schedule.

With human failure modes, we are mostly interested in the number of human related problems the experiments participants are able to identify. Because of this, we have chosen to categorize the problems as whether they are human- or non-human related. Just because a problem is related to the operator (OP) or train engineer (EP), does not necessarily mean it is human related. We are more interested in the problems created or initiated by the person interacting with the system, than problems occurring as the person react to a erroneous system behavior. An example of this is OP103 - ”Do not receive message”, which is the related to the underlying technology, in this case the computer system, not being able to send the message, thereby causing the operator to not receive it. We are more interested in the problems related to the human interacting with the system in a wrong way or not at all. The categorizes we have chosen as human related, are shown in table 5.6.

If we only take into consideration the human related categories and once again chose to show the problems that more than 30% of the test subjects manages to find, there is only one change to the chart - EP103 ”Do not receive message” is removed. This is a good indication that the method is only discovering human related failures, and not the other parts of the system, like the errors related to the computer system. The total number of human related problems found is shown in figure 5.10. We note that operator actions (OP200 and EP200) and operator scheduling (OP300 and EP300) are the most commonly found problems. On the other hand, problems related to incoming messages (OP100 and EP100), operator knowledge (OP500 and EP500) and operator overload (OP600 and EP600) are all poorly covered in comparison.

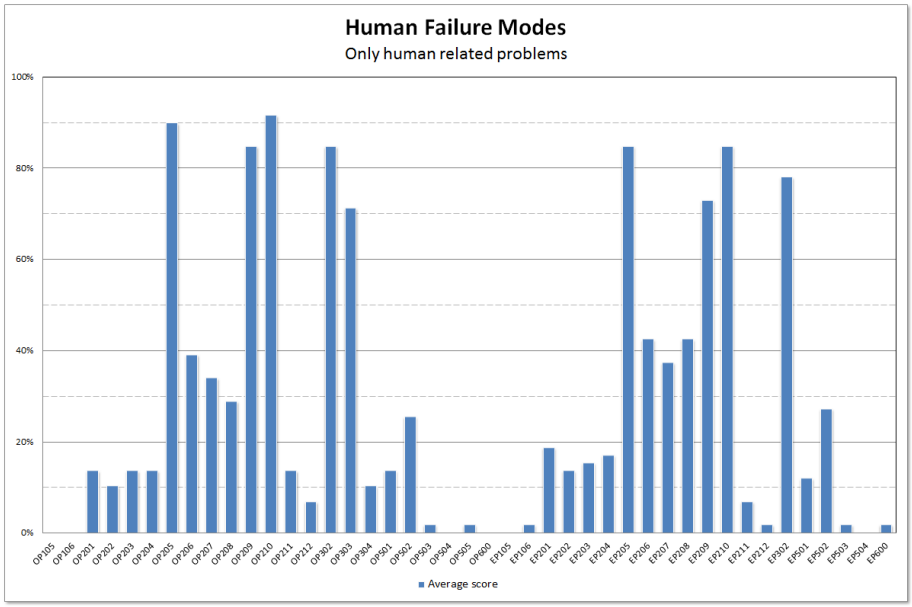


Figure 5.10: Problems categorized to be human related

Operator	Train engineer
OP105 Incomplete message received from other person	EP105 Incomplete message received from other person
OP106 Ignores message or alarm	EP106 Ignores message or alarm
OP201 Wrong action or non-existing or wrong command	EP201 Wrong action or non-existing or wrong command
OP202 Unable to perform action	EP202 Unable to perform action
OP203 Fails to ack	EP203 Fails to ack
OP204 Wrong ack	EP204 Wrong ack
OP205 Responds too late	EP205 Responds too late
OP206 Fails to respond	EP206 Fails to respond
OP207 Fails to perform	EP207 Fails to perform
OP208 Wrong response	EP208 Wrong response
OP209 Enters wrong info	EP209 Enters wrong info to operator
OP210 Forget to perform action – whole of part of	EP210 Forget to perform action, do not notice the need to perform action
OP211 Do not save info	EP211 Do not save info
OP212 Do not save changes to schedule etc.	EP212 Do not save changes to schedule etc.
OP302 Wrong train scheduling	EP302 Wrong train scheduling
OP303 Fails to schedule	-
OP304 Fails to report	-
OP501 Wrong situation analysis	EP501 Wrong situation analysis
OP502 Fail to discover dangerous situation	EP502 Fail to discover dangerous situation
OP503 Wrong interpretation of system's functionality	EP503 Wrong interpretation of system's functionality
OP504 Lack of training	EP504 Lack of training
OP505 Lacking info – all or in part	-
OP600 Operator overload	EP600 Engineer overload

Table 5.6: Human related problems

5.5 Evaluation of methods

When the participants had finished answering one of the test cases, they were given a post-experiment questionnaire and asked to evaluate the method and their own performance. The questionnaire consisted of 10 statements where they answered to which degree they agreed, ranging from ”disagree strongly” to ”agree strongly”. When calculating the score for the methods, each answer has been given a score from 1 to 5, where 1 = ”Disagree strongly”, and 5 = ”Agree strongly”. An average score has then been calculated, to summarize the

feedback given to each of the methods. Figure 5.11 shows the average feedback.

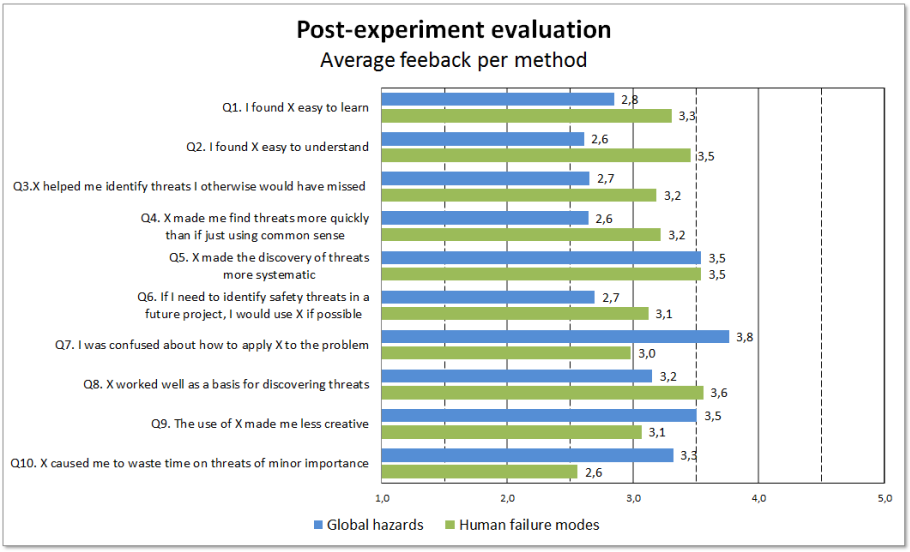


Figure 5.11: Average feedback of both methods

Given that the score 3.0 is neutral, we should be pleased with any feedback that has a score above this value. Note however, that not all the statements are stated in a positive way, like Q7, Q9 and Q10. On these answers we want a score of below 3.0. We see that human failure modes have in general received favorable feedback from the test subjects. Most of the participants found the method easy to learn and understand, and that they felt it helped them with the discovery of threats. The participants did, however, have some problems with how they should use the method to solve the problem. Global hazards on the other hand, received an overall poor feedback, where most of the participants had problems with learning and using it as well as being confused with how to apply it to the problem.

In order to visualize the interesting parts of the data, we show the distribution of answers. With this, we remove the ”neutral” answer, and focus on the answers that either agree or disagree with the statement. By doing this, we can see how large a portion of the test subject that answered positively (agree) or negatively (disagree) for each statement. This should help us find statements where the test subjects are divided in their opinion.

First we will look at the evaluation of the method of global hazard with can-cause chains, shown in figure 5.12. Overall, we see the same trends as we saw in figure 5.11. However, there are some additional results. For Q1, we see that the feedback is rather contrasting with 31% finding it easy to learn and 41% finding it difficult to learn. We also see that there is some disagreement for Q8, with 36% finding it as a good basis for discovering threats, while 20% disagrees.

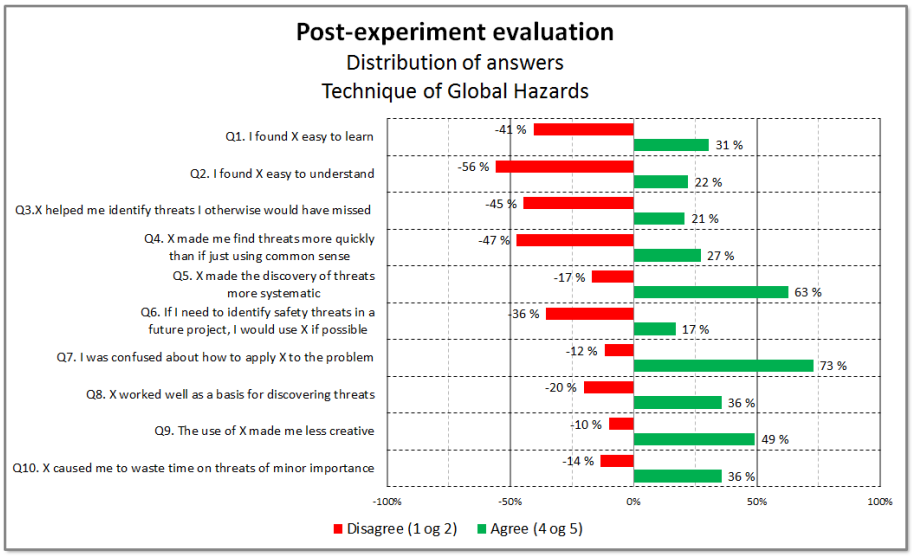


Figure 5.12: Evaluation of using global hazards with can-cause chains

For Human failure modes, we have multiple statements that the test subjects does not agree upon, as seen in figure 5.13. For Q9, there is about equal amounts of participants saying the method made them less creative - 34%, as those saying it did not affect their creativity - 36%. We also note that feedback on Q7 is split into two groups, those that were confused of how to apply the method to the case - 36%, and those that did not find it difficult - 41%. The last thing to note is that for statement Q3, there is also a disagreement as to whether the method helped them identify threats otherwise missed.

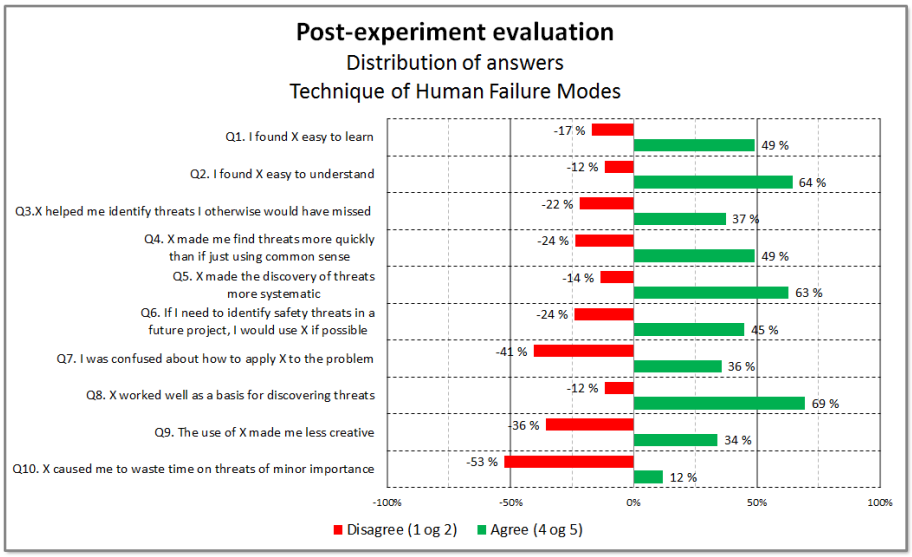


Figure 5.13: Evaluation of human failure modes

5.6 Hypothesis testing

As mentioned in section 3.3, we will compare the human failure modes method against the method of using system diagrams. The method of system diagrams is a rather simple approach to hazard identification. The user is shown a system diagram, from which he will mark the part of the system that will be inspected, and write down the possible hazards that can occur in that part of the system. The data from using this methods stems from an earlier student experiment performed at NTNU. The experiment used the same case, train control system, as in our experiment and has been analyzed and coded using the same problems we have.

In order to get an idea of the differences between the data collected in the two experiments, we have taken the average score for human failure modes, and subtracted the average score for the method of using system diagrams. Thus, the problems code with a value above 0.0 imply that the test subjects using human failure mode found the problem more often than those using the system diagrams, and vice versa for problems code with negative score. We have also chosen to only use the human related problems, since that is what we want to compare. Figure 5.14 show the result.

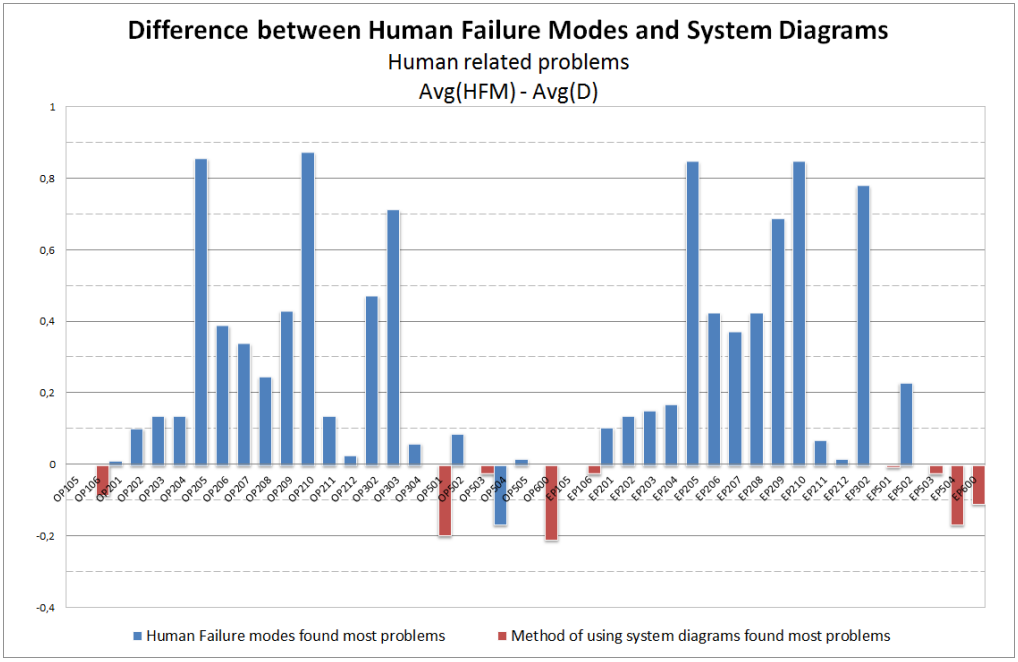


Figure 5.14: Comparing the average score of human failure mode, against Diagram

We see that human failure modes appear to be able to find more problems related to the humans operator in most of the cases. On the other hand, the method of using system diagrams seems to have found more problems for five of the categories. By looking at the chart, we would expect there to be a significant difference between the two methods.

As we have stated earlier in this chapter, we have obtained a set of quantifiable data from our experiment, and now want to compare this set with another set from an earlier performed experiment. The two cases are similar, but have been solved with different methods and by two different groups of students. Because of this, we have chosen to us a t-test to compare the two sets of data and answer the hypothesis stated in chapter 3.

The datasets we have chosen to compare is the average number of human related problems identified by the two methods. We have chosen a confidence level of 95%, meaning that in order to reject the null hypothesis, we will need a p-value, the probability of a false positive, of less than 0.05 in order for the results to be statistical significant. Since our null hypothesis is that the method of human failure modes will not be different, or worse than system diagrams at detecting operator related problems, we use a one-tailed t-test. We will be looking for a significant difference as well as that the mean values of human failure modes are higher than the method of system diagrams. The results from our t-test is shown in table 5.7

	Human Failure Modes	**System Diagram**	**Difference**
Mean	0.281	0.065	0.216
St. deviation	0.309	0.104	0.204
St. error mean	0.047	0.016	0.031

Table 5.7: Results from the t-test

From of the one tailed t-test, we calculated our t-value to be 4.36 and the p-value to be 0.00003. By conventional criteria, this difference is statistically significant. From this we can conclude that the two methods produce different results. Further, we see that the difference in mean is 0.216, meaning that the mean value is greater for human failure modes which implies that it on average finds more hazards. Our alternative hypothesis was that human failure modes would be able to find more operator related problems than system diagrams method, which our t-test confirm. We therefore chose to **reject the null hypothesis** in favor of the alternative H1 hypothesis, which state that human failure modes identifies more operator related threats than using only system diagrams.

5.7 Data interpretation

We have just shown all the data that has been collected throughout the experiment in the previous sections. We will here try to interpret at the data that has been presented and ask why we get these results.

For global hazards there are certain aspects of the data that should be further discussed. There is a large variation in the number of times an initiating event is identified for the different hazards. If we take E01 as example, we see that it was identified by 72% of the test subjects for H02: Fire, but none of them was able to identify the chain for H03: leak from water pipe. We are not completely sure as to why there is such a large variation, but we see that it appears to be regarding all of the sensor failure (E01, E02 and E03), but not the other initiating events (E06, E13/E14 and E08/E11). We believe it may be related to the test subjects feeling a need for identifying new hazardous can-cause chains, and not repeat the same ones. Further, the name of the sensors can impact which global hazards the test subjects feel they ”belong” to, e.g. - water level sensor belongs to leakage of water, and pressure sensor belongs to leakage of steam.

The event E06 - Thermostat error, was the least commonly identified initiating event. After looking through the answers, we saw a trend towards not using it as an initiating event, but rather as a second event in the can-cause chains, typically after the event E01 - Temperature sensor fails. We believe that this show that the test subjects might feel that this event has been covered, since it already it part of a chain. Our approach to categorize the can-cause chains has not taken this into consideration, since we only enter the initiating event of each chain, thereby ”hiding” information and making the displayed data somewhat skewed.

The most commonly mentioned chains were the one related to E11/E08 - the operator. As we stated in section 5.3, this is a somewhat special case, given that it is not an initiating event, but rather a hazardous reaction by the operator on an initiating event by the operator. Because of this, many of the chains we found, typically the operator reacting to an error or event, would be categorized as E11/E08. An event that might start with an E01 - temperature sensor fails, which also contain the operator reacting to the event, would not be categorized as E01, but E11/E08.

When interpreting the feedback on the can-cause chains, we see that many of the test subjects were confused as to how to use the method. If we also look at the average number of chains identified, regardless of being valid or not, we see that the test subjects typically found a little more than 2 chains (2.3). For each global hazard, there were six (five for H01) chains to be identified. We feel that the number of chains identified by the students should be higher, preferably finding more than half of the chains. When looking over the answers, we saw that some of the participants had crossed out correct chains, or stopped half-way through

them. This shows us that we have probably not explained and presented how to use the method good enough, and should have had more relevant examples of how to use the method. With the test case and events we had, the chains were typically short and simple. This might have made the test subjects second guess whether they had misunderstood the method, e.g. - not believing that it could be that simple. This becomes even more relevant when comparing the answers given in the other test case for human failure modes, which was typically longer and more work intensive to write/ and identify.

When interpreting the data collected for human failure modes, we have found some interesting aspects that need to discussed further. The first trend in the answers, was that the experiment participants usually was rather similar to the generic failure modes presented in the hazards identification table, e.g. - ”forget”, ”wrong” and ”too late”. This ensures that important parts of the hazards related to the operator is identified, but can also remove focus and hide other hazards that are not as easily identified. By answering the hazards related to the generic failure modes, the person performing the hazard analysis can get a false sense of safety, by filling in the whole table while still not having identified all threats. It is difficult to evaluate whether this is a real problem with human failure modes or not based on our data.

In regards to identifying hazards, we see that operator action problems for the operator have not been identified as often for the train engineer, even though they are similar. For example, we see that the most commonly identified problem, OP210 ”forget to perform action” has been identified by 92%, while EP210 ”only” has been identified by 85%. When working through the answers, we saw that many of the test subjects appeared to have much shorter answers, or not answer at all for the last tables. We believe that this might be related to the students becoming bored or tired of writing somewhat similar answers multiple times. We had ten requirements, five each for the operator and train engineer. Given that each requirement will have three failure modes, this makes the test subjects fill in thirty tables. This reveals a weakness with the method: the person performing the hazards analysis can become tired of repeating somewhat similar answers and feel it to be an unnecessary large amount of work.

When analyzing the answers the test subjects had given, there were some of the students that had written some feedback in regards to the experiments. It was in total four persons that had chosen to do this, and they all said, more or less, the same thing - ”The train control system case was a lot more difficult than the steam boiler system”. If four persons had written this as feedback, we feel it is safe to assume that more of the test subjects felt the same way. This may have influenced the feedback we got in the post-experiment questionnaire, since the test subjects may have felt the task difficult to solve, and thereby giving the method a bad review. It is therefore interesting to see that the feedback on

human failure modes still was overall favorable.

Lastly, we will take a look at the comparison of human failure modes to using system diagrams. When looking at the number of hazards identified related to the operator, students using human failure modes did identify a lot more hazards on the average, even though the use of system diagram did identify more of some of the hazards. The difference does, however, also show that the use of human failure modes and hazards identification table can hinder creative thinking and take focus away from certain part of the system. For the students using human failure modes, they had both the system diagram and hazard identification table available, but the other group did still identify some threats better. However, when comparing the number of human related problems found, the students using human failure modes did on the average identify more threats. For human failure modes 14 hazards where identified with a probability larger than 0.30, while students using diagrams had 3 hazards identified.

6 Evaluation

6.1 Global hazards

Our research question was whether it was easier to discover environment threatening hazards, using can-cause chains. After having collected and analyzed the answers from the experiment, we have found some interesting data with regards to using can-cause chains for global hazards. The first thing to mention is that because of the lack of similar methods and work done on this subject, it quickly became inherent that it would be difficult to compare this method to any other method. Since we had a limited amount of time and resources available, we decided to treat this part of the experiment more as a try-out, and see what data we collected and possible problems we would identify. It is clear that the method needs more work before it will function as we described in our Specialization Project, described in chapter 2.

From the data we have collected, we have seen that the test subject have had problems with learning the method and understanding how to use it. We think that this came from not having presented the method thoroughly enough, and not providing them with enough relevant examples to understand how to use it. We saw that the students had different approaches to how to build the can-cause chains. Some created many small and simple chains, while other had few complex chains that had almost every event in the system included. The large variation in how the chains were constructed showed that the students had very different opinions on how they should be built.

We experienced difficulties when describing what a valid chain would look like, in contrast to an invalid. Even though our test case was rather small, and had few events, we saw that there were a lot of different ways to describe the chains leading to an environmental hazard. Many of these chains were almost identical, with just the chain of events re-arranged. We chose to overcome this by categorizing the chains according to what the initiating event was and creating more complete can-cause chains directly from the system diagram. This made us able to collect the data and analyze them, but also revealed that there was more work needed on the definition of the can-cause chains.

If we are to overcome these problems, we will have to define clearer and more strict rules as to what defines relevant can-cause chains, and what should be done with events that are optional or when events that can split into different new can-cause chains. When discussing this in our Specialization Project, we said that most of this work would be done by an accompanying ontology. However, after having tried to use the method with test subjects, on a system, we see that we need more than an ontology in order to overcome these problems.

Since this has been an exploratory experiment, we can say that we have learned from it, and identified aspects of the method that will need further work and research. The feedback and data from the experiment will hopefully be a step toward creating a working method of identifying environmental hazards like we had envisioned in our Specialization Project. We can, however, not give a definitive answer to the question we asked our selves when starting the master thesis, if it is easier to identify environment hazards with can-cause chains

6.2 Human failure modes

The experiment shows that the hazards the test subjects identified using the method of human failure modes was mainly human related hazards. As well as finding the type of hazards we were interested in, the data also show that most of the participants were good at identifying a large specter of human related hazards, with 14 human related hazards identified with a probability larger than 0.30. The test subjects have also given the method overall favorable feedback, even though we got some feedback saying the test case was too complicated.

There were, however, some tendencies towards that the students answerers were getting shorter and less specific with the later requirements. Typically the last answers said to look at earlier answers (”see answer for R02”). This can indicate that filling in the hazard identification table can be a repetitive, which can make the person performing the hazard analysis skip part of it. This is not favorable, as we have stated that we need the method to be easy and quick to use, since this is typically thought to be for an early hazards analysis for a system. However,

when looking at the answers given in the post-experiment questionnaire, we see that only 12% felt that the method made them waste time on threats of minor importance (Q19) and 49% said that it made them discover threats more quickly than with only common sense (Q4).

When comparing the method of human failure modes to using system diagrams to identify threats, we saw that even though human failure modes found more threats, it did not cover all parts of the system. There were some hazards that were easier to discover when only using the system diagram. The hazards found by system diagrams with a probability larger than 0.2 than human failure modes were OP501 ”Wrong situation analysis” and OP600 ”Engineer overload”. This is important to note that using human failure modes will not cover all human related hazards, so to gain full coverage you would need more than one method. But, the human failure method did cover much larger part of the human related hazards, with 16 hazards found with a probability larger than 0.2 than the method of system diagram. Given how easy it should be to create such a table if the boilerplates and ontology already is available, we consider the method of human failure modes to be a good method for covering the human related hazards in a system.

The research question for human failure modes was whether it made it easier to discover operator related hazards. With the data collected and analyzed, we can now answer our research question in regards to the hypothesis we asked. The results are that human failure modes do make the process of identifying operator hazards easier.

7 Conclusion and further work

7.1 Conclusion

The purpose of this thesis have been to investigate how we can use the method of can-cause chains with global hazards and human failure modes in safety analysis and if they are better than other methods. The work presented in this report is a continuation of the work done in TDT4501 - Specialization Project, where we first identified these new methods for performing early safety analysis, based on the use of boilerplates and ontologies.

We have chosen to perform an experiment to test the two methods on students to see if they will enable a easier hazard identification. For the experiment, we crated two test cases, one for each method. A steam boiler system for testing global hazards with can-cause chains, and a train control system for testing human failure modes. After each test case, the students were asked to evaluate the methods.

From analyzing the data from the experiment, we have been able to evaluate how well the two methods work for safety analysis. The results from global hazards with can-cause chains indicate that the method is not in a state where it can be used for safety analysis as of yet. There is still too much ambiguities as too how the chains should be crated, and the feedback from the students indicate that it difficult to learn and use the method. The algorithm needs to be further structured and better documentation of how to perform it must be created.

The data from the experiment indicate that human failure modes have proven to be efficient at identifying operator related hazards. The methods was given overall favorable feedback from the student, and appeared to identify many of the hazards in the test case. The methods was compared against another hazard identification methods - using system diagrams, from an older student experiment. We concluded with that human failure modes was better at identifying human related hazards, and can **reject the null hypothesis** in favor of the alternative H1 hypothesis, which state that human failure modes identifies more operator related threats than using only system diagrams.

Lastly, we need to answer the research questions formulated before we started on the experiment. We have not found data that implies that it becomes easier to identify environmental hazards with global hazards and can-cause chains (RQ1). The results from the experiment indicates that it is easier to identify operator hazards when using human failure modes (RQ2).

7.2 Insights from the experiment

After working through our results and writing this thesis, we have gained some insights in regards to the two safety analysis methods presented in this report. We have listed the most important ones below:

Global hazards

- Need to define clearer and more strict rules for the chains. In our Specialization Project, we said that most of this work would be done by an accompanying ontology. However, after having tried to use the method with test subjects, on a system, we see that we need more than an ontology in order to overcome these problems.

Human failure modes

- The person performing the hazard analysis can get a false sense of safety, by filling in the whole table while still not having identified all threats.
- The person performing the hazards analysis can become tired of repeating somewhat similar answers and feel it to be an unnecessary large amount of work.
- The use of human failure modes and hazards identification table can hinder creative thinking and take focus away from certain part of the system
- Using human failure modes will not cover all human related hazards, so to gain full coverage you would need more than one method

7.3 Further work

The following present what we consider to be important aspects to investigate further either as experiments or refinements of the methods presented in this thesis.

Create a more structured definition of the can-cause chains
When analyzing the experiment, we saw that the can-cause chains was not well understood and varied a lot from student to student. We think that the definition of global hazards should be improved and better structured. There need to be a well developed structure that will take into consideration aspects like optional sub-chains, the ordering of events and

Implement the can-cause algorithm in GNLQ
Since we already have the requirement elicitation tool GNLQ available, it would be natural to continue developing this tool. With this prototype we would be able to identify strong and weak aspects of the the can-cause algorithm, as well as identify new functionality that we may want to further investigate and possibly add to the tool.

Combine human failure modes with generic failure modes for components
It would be of interest to test the coverage and efficiency for a method that contain both human and equipment failure modes. It should be simple to add human failure modes to the already existing hazId tables with generic failure modes.

Experiments to compare it to other safety analysis methods
In this master thesis we have compared human failure modes against a method of using system diagram to identify hazards. It would be interesting to compare the method against other safety analysis methods, to see if we will arrive at the same conclusions as in our experiment.

Identify measures to minimize the negative aspects of human failure modes
In the previous section, we identified three problem areas where human failure modes have potential to improve - false sense of safety, repeating similar answers and hinder creative thinking. It would be of interest to test reassures to counter them, and thereby hopefully improve of the method.

8

References

[1] Inah Omoronyia. GNLQ description. http://sourceforge.net/projects/gnlq/, 2010.

[2] Martin Host Claes Wohlin, Per Runeson. *Experimentation in Software Engineering: An Introduction*. Kluwer Academic Publishers, 1999.

[3] Christian O. Hjorth. Hazard boilerplates in safety analysis: Automated hazard identification using boilerplates and ontologies, 2012.

[4] Jeremy Dick Elizabeth Hull, Ken Jackson. *Requirements Engineering*. Springer, 2004.

[5] Tor Stalhane Olawande Daramola, Guttom Sindre. Pattern-based Security Requirements Specification Using Ontologies and Boilerplates.

[6] Omar Chiotti Graciela Brusa, Ma. Laura Caliusco. A process for building a domain ontology. http://crpit.com/confpapers/CRPITV72Brusa.pdf.

[7] Olawande Daramola Tor Stalhane, Stefan Farfeleder. Safety analysis based on requirements.

[8] Marvin Rausand. Risk Assessment - Hazard and Operability Study. http://frigg.ivt.ntnu.no/ross/risk/slides/hazop.pdf, 2011.

[9] Peter B. Ladkin. Ontological analysis, 2005.

[10] Wikipedia. Hazard and operability study. http://en.wikipedia.org/wiki/Hazard_and_operability_study.

[11] Tor Stalhane. Development of safety ctritical software, 2010.

[12] Michael Ellims. Safety Analysis: Thoughts on Methods and Experience.

[13] Helminen Atte Haapanen Pentti. Failure mode and effect analysis of software-based automation system. Technical report, STUK, 2002.

[14] Vikash Katt Tor Stalhane, Olawande Daramola. Patterns in Safety Analysis.

[15] Cesar Project. Revised Definitions of Improved RE Methods, 2011.

[16] Peter Fenelon, Y Dd, and Barry Hebbron. Applying hazop to software engineering models. In *in Risk Management And Critical Protective Systems: Proceedings of SARSS 1994*, 1994.

[17] Wikipedia. Eploratory research. http://en.wikipedia.org/wiki/Exploratory_research.

[18] Wikipedia. Boilerplate code. http://en.wikipedia.org/wiki/Boilerplate_code.

[19] W.G. Hopkins. A new view of statistics,. University of Queensland, Australia: Brisbane, 2001.

A
Appendix

A.1 Experiment

Hazard boilerplates in safety analysis

Aspects of hazard identification using boilerplates and ontologies

Experiment 01a

Human Failure Modes & Global Hazards

Christian O. Hjorth
Supervisor: Tor Stålhane

Wednesday, April 17, 2013

Experiment introduction

The purpose of this experiment is to test two methods for performing hazard analysis (safety analysis). The methods will be compared by letting the participant perform the following activities:

1. Fill in the pre-experiment questionnaire – approximately 5 minutes.
2. Read the tutorial – approximately 10 minutes.
3. Solve the given tasks – approximately 50 minutes.
4. Fill in post-experiment questionnaire – approximately 5 minutes.

Pre-experiment questionnaire

I have no previous experience with safety analyses	☐ ☐ ☐ ☐ ☐	I have a lot of previous experience with safety analyses
I have no previous experience with reliability analyses	☐ ☐ ☐ ☐ ☐	I have a lot of previous experience with reliability analyses
I have no previous experience with boilerplates	☐ ☐ ☐ ☐ ☐	I have a lot of previous experience with boilerplates

Have you had TDT4242 - Requirements and testing, or any similar courses?

☐ YES ☐ NO

☐ OTHER: ……

Number of completed semesters of study after high school ……

Number of months of IT working experience, including summer jobs ……
(recalculated to full time, e.g., a 10 month 20% part time job should be reported as 2 months)

PERSON-ID: ……

Tutorial on Safety analysis

Throughout this experiment you shall perform early hazard identification for two different system using can-cause chains to detect global hazards and human failure modes to identify possible operator failures.

Can-cause chains

We start with a failure that can cause local effect on the system. This effect may then cause new effects on the local system thereby starting a chain of events. Eventually this may lead to a point where it no longer only affects the local system, but also the environment is operates in, possibly causing an environment hazard. What we are interested in here is to identify this chain of events leading to the environment hazard.

Example

- <Sensor error> **can cause** <Wrong value reported>
  - ↳ <Wrong value reported> **can cause** <Wrong command issued>
    - ↳ <Wrong command issued> **can cause** <Wrong countermeasure initiated>
      - ↳ <Wrong countermeasure initiated> **can cause** <Catastrophic event>

Human failure modes

Human failure modes are generic failure modes that are related to the operator of the system. They are intended to make sure that all tasks related to the operator are included in the safety analysis. The failure modes are used for requirements that include an operator. There are three failure modes available:

- Forget – forget to perform an operation
- Wrong – wrong operation performed
- Too late – correct operation performed too late

Example

R1: **The** <operator > **shall be able to** <display water level>

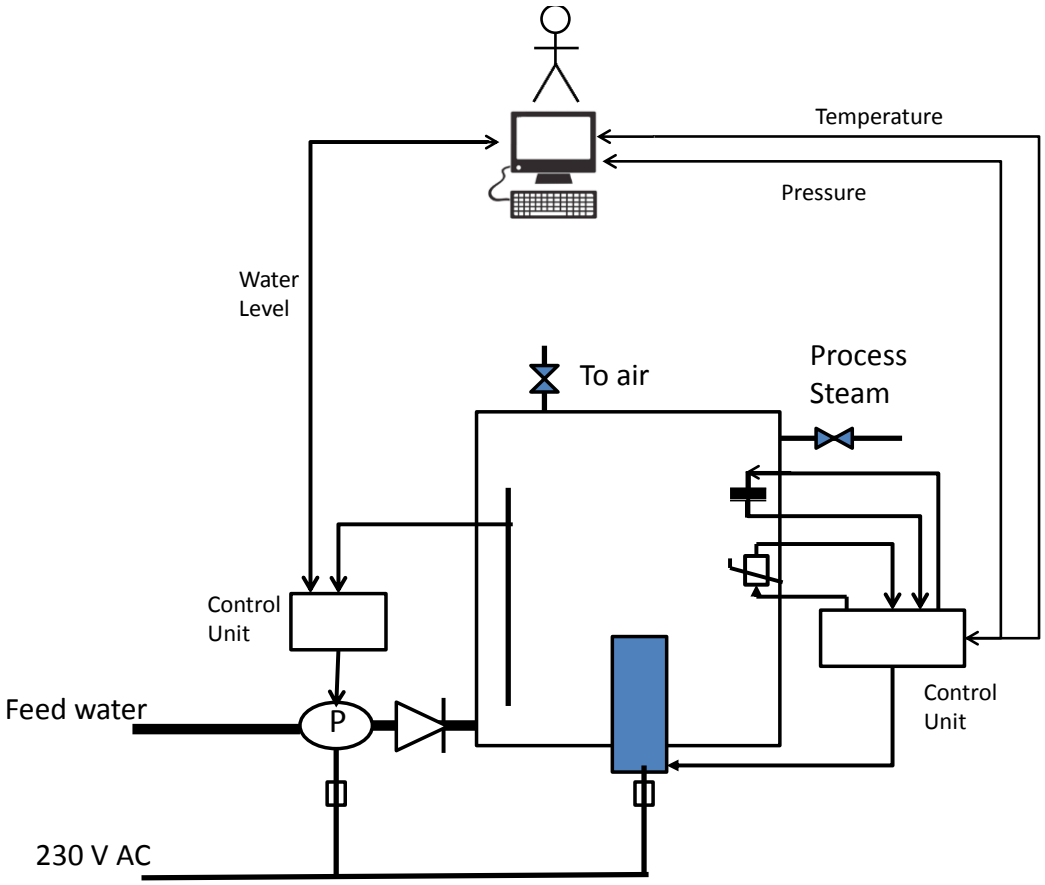
Req. Id	Failure mode	Effect of failure
R01	Forget	May miss the water level getting too high/low, leading too other possible more dangerous events
	Wrong	Fails to open valve to increase water levels when dangerously low
	Too late	Water level is too low. Control unit fails to maintain correct water level, alarm message sent to operator. Operator open valve to fill up with more water, but it might be too late to avoid a mishap

Case to be solved

Below you will find two cases. Each of these cases includes a system overview diagram, followed by a short system description and the problem to be solved. You should start by studying the system diagram thoroughly before you try to solve the problem. Use the available information to find as many safety threats as possible. Note that the information given is not necessarily complete and you are encouraged to add any information you feel is necessary.

CASE 1 – Steam Boiler (Global Hazards)

System diagram



System description

The diagram above shows a simple steam boiler. It is fed water via a pump through a non-return valve. A control mechanism consisting of level measurement equipment, a computerized controller and a pump is used to assure that the water level is kept between defined maximum and minimum values. Sensors that measure pressure and temperature are fed to a computerized controller that starts or stops the heating element in order to keep temperature and pressure approximately constant. If a control unit fails, the operator will get an alarm message on the screen. He should then override the control units with commands from the terminal.

Event list

Event Id	Events	Description
E01	Temperature sensor error	Sensor reports a value below/above the actual temperature
E02	Pressure sensor error	Sensor reports a value below/above the actual pressure
E03	Water level sensor error	Sensor reports a value below/above the actual water level
E04	Heater generate too much heat	Heater generate too much heat causing the water temperature to raise too much
E05	Low pressure in tank	The pressure is below the operating range of the steam vessel
E06	Thermostat error	The thermostat sets wrong or no temperature
E07	Heater does not generate heat	Heater generate too little heat, causing the water temperature to drop too low
E08	Operator give wrong command	Operator give a wrong command, either due to wrong information, or by accident
E09	Feed valve stuck	Feed valve stuck, preventing the ability to regulate water level
E11	Wrong information displayed in control room	Wrong information is displayed, either due to malfunctioning sensors, or errors in the control room
E12	High pressure in tank	The pressure is above the operating range of the steam vessel
E13	Incorrect behavior from the automatic water level component	The automatic water component does no regulate the water level as intended
E14	Incorrect behavior from the automatic temperature/pressure level component	The control unit does not regulate the temperature/pressure level as intended

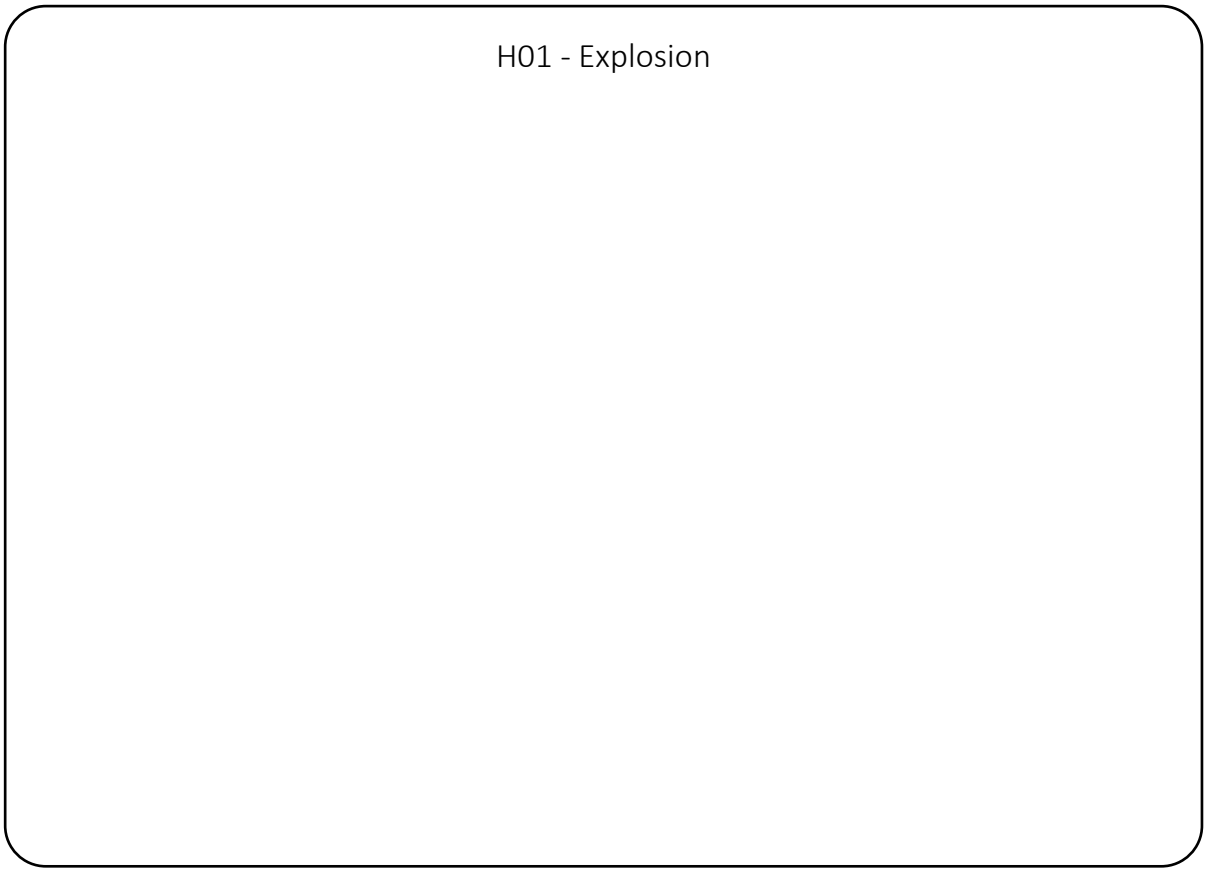
Global Hazard list

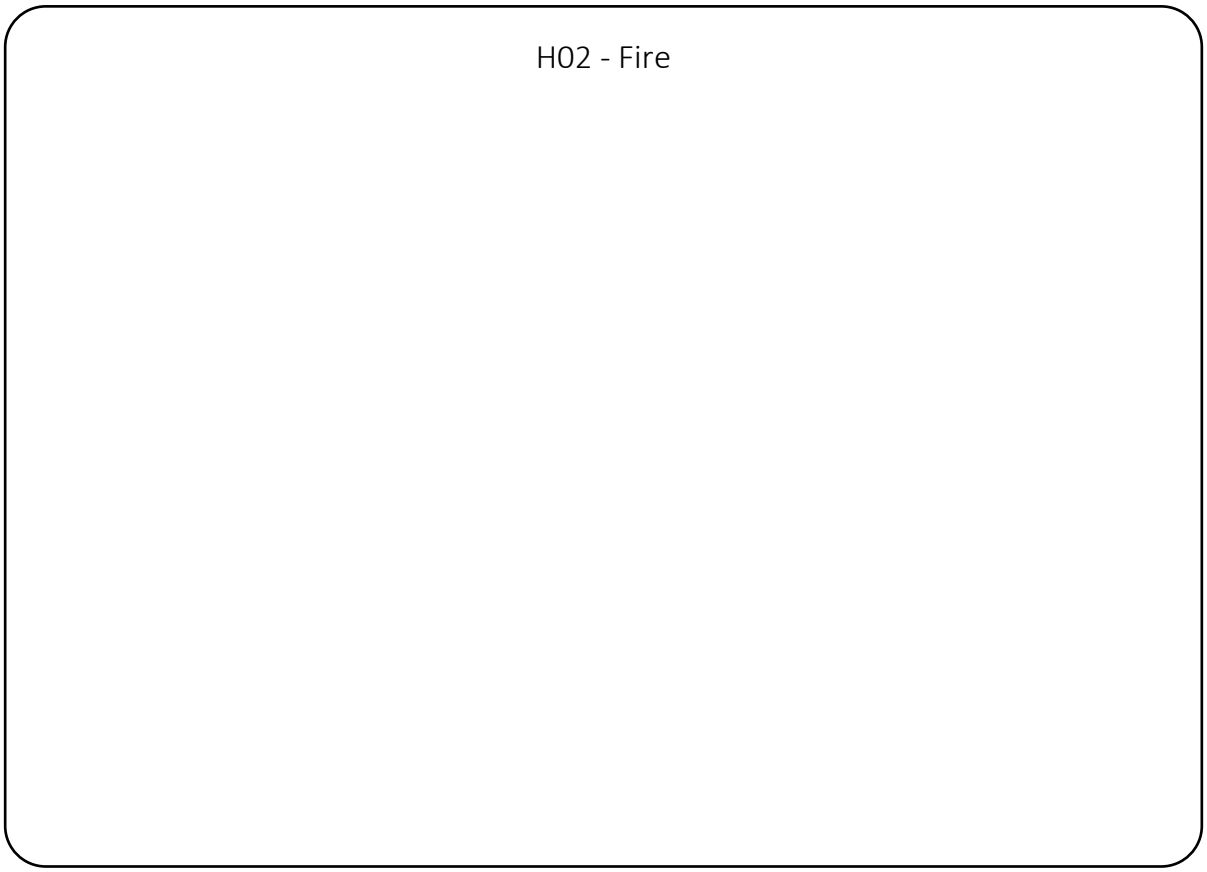
Hazard Id	Global Hazard	Description
H01	Explosion	An explosion will have major impacts on the steam vessel and the environment it operates in, leading to a possible weakening of the structural integrity of the surrounding buildings.
H02	Fire	A fire will have major impacts on the steam vessel and its environment.
H03	Leak from liquid pipe	A leak from the liquid pipe may have large impacts on the surrounding environment of the steam vessel. Blockage of exists and destruction of equipment.
H04	Leak form steam pipe	A leak from the steam pipe may have large impacts on the environment of the steam vessel. Damage to personnel and destruction of equipment.
H05	Vessel rupture	A vessel rupture may have large impacts on the steam vessel and its surrounding environment. Damage to personnel, blockage of exists and destruction of equipment.

Hazard identification

On the following pages you shall identify all events that may lead to a global hazard. A global hazard is an effect that will impact not only the system, but also the environment it operates in. You will use "**can-cause**" chains to show how an event may lead to a global hazard.

Can-cause chains





H03 – Leak from liquid pipe

H04 – Leak from steam pipe

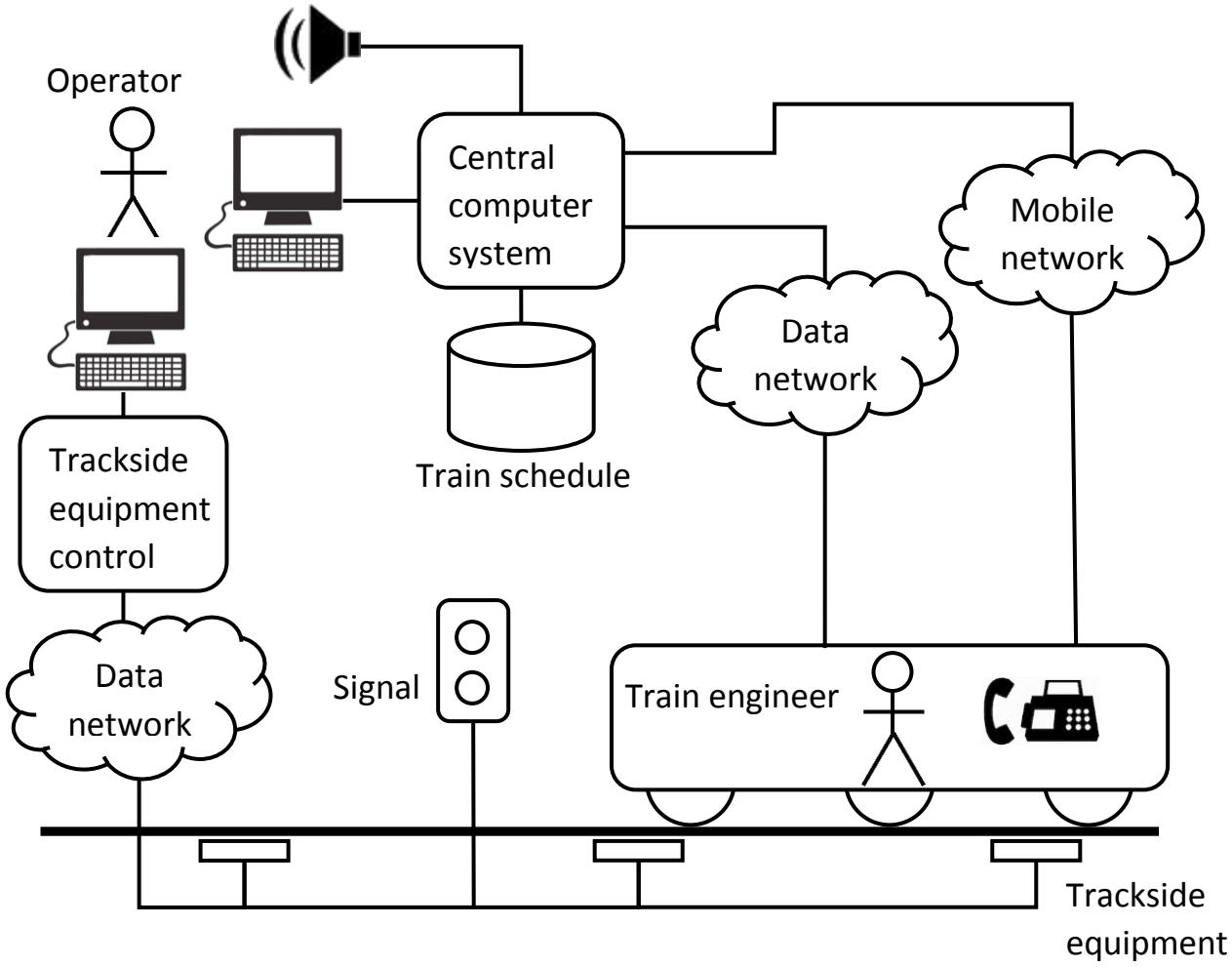
Post-experiment questionnaire

T1 = “The technique of global hazards”

	Disagree strongly	Disagree	Neutral	Agree	Agree strongly
Q1. I found T1 easy to learn	☐	☐	☐	☐	☐
Q2. I found T1 easy to understand	☐	☐	☐	☐	☐
Q3. T1 helped me identify threats I otherwise would have missed	☐	☐	☐	☐	☐
Q4. T1 made me find threats more quickly than if just using common sense	☐	☐	☐	☐	☐
Q5. T1 made the discovery of threats more systematic	☐	☐	☐	☐	☐
Q6. If I need to identify safety threats in a future project, I would use T1 if possible	☐	☐	☐	☐	☐
Q7. I was confused about how to apply T1 to the problem	☐	☐	☐	☐	☐
Q8. T1 worked well as a basis for discovering threats	☐	☐	☐	☐	☐
Q9. The use of T1 made me less creative	☐	☐	☐	☐	☐
Q10. T1 caused me to waste time on threats of minor importance	☐	☐	☐	☐	☐

CASE 2 – Train (Human Failure modes)

System diagram



System description

The diagram above shows a simple train control system. The track is built up of multiple zones. Only one train may be in a zone at the same time, in order to avoid head on collisions. The train is operated by a train engineer that regulates its speed. He receives information about currently running trains from the control system, and the train sends updates about its own schedule. The control system keeps track of all the trains on the track and their times/positions. It updates the database accordingly and sends the information out to the trains. If the control system fails, the operator and the train driver can communicate positions, schedules etc. via cell phone.

System requirements

Req. id	Description
R01	<Operator> **shall be able to** <request scheduling screen>
R02	<Operator> **shall be able to** <request identified schedule>
R03	<Operator> **shall be able to** <enter scheduling info>
R04	<Operator> **shall be able to** <update existing scheduling info>
R05	<Operator> **shall be able to** <confirm schedule>

Req. id	Description
R06	<Train driver> **shall be able to** <request status screen>
R07	<Train driver> **shall be able to** <enter status>
R08	**If** <train driver enters status> **then** <system> **shall** <acknowledge status info>
R09	**If** <train driver cannot accept schedule> **then** <train driver> **shall** <request new schedule>
R10	**If** <system provides new schedule> **then** <train driver> **shall** <acknowledge new schedule>

Hazard identification table

Fill in the effect of the failure modes below.

Req. Id	Failure mode	Effect of failure
R01 Operator	Forget	
	Wrong	
	Too late	
R02 Operator	Forget	
	Wrong	
	Too late	

Req. Id	Failure mode	Effect of failure
R03 Operator	Forget	
	Wrong	
	Too late	
R04 Operator	Forget	
	Wrong	
	Too late	

Req. Id	Failure mode	Effect of failure
R05 Operator	Forget	
	Wrong	
	Too late	
R06 Train engineer	Forget	
	Wrong	
	Too late	

Req. Id	Failure mode	Effect of failure
R07 Train engineer	Forget	
	Wrong	
	Too late	
R08 Train engineer	Forget	
	Wrong	
	Too late	

Req. Id	Failure mode	Effect of failure
R09 Train engineer	Forget	
	Wrong	
	Too late	
R10 Train engineer	Forget	
	Wrong	
	Too late	

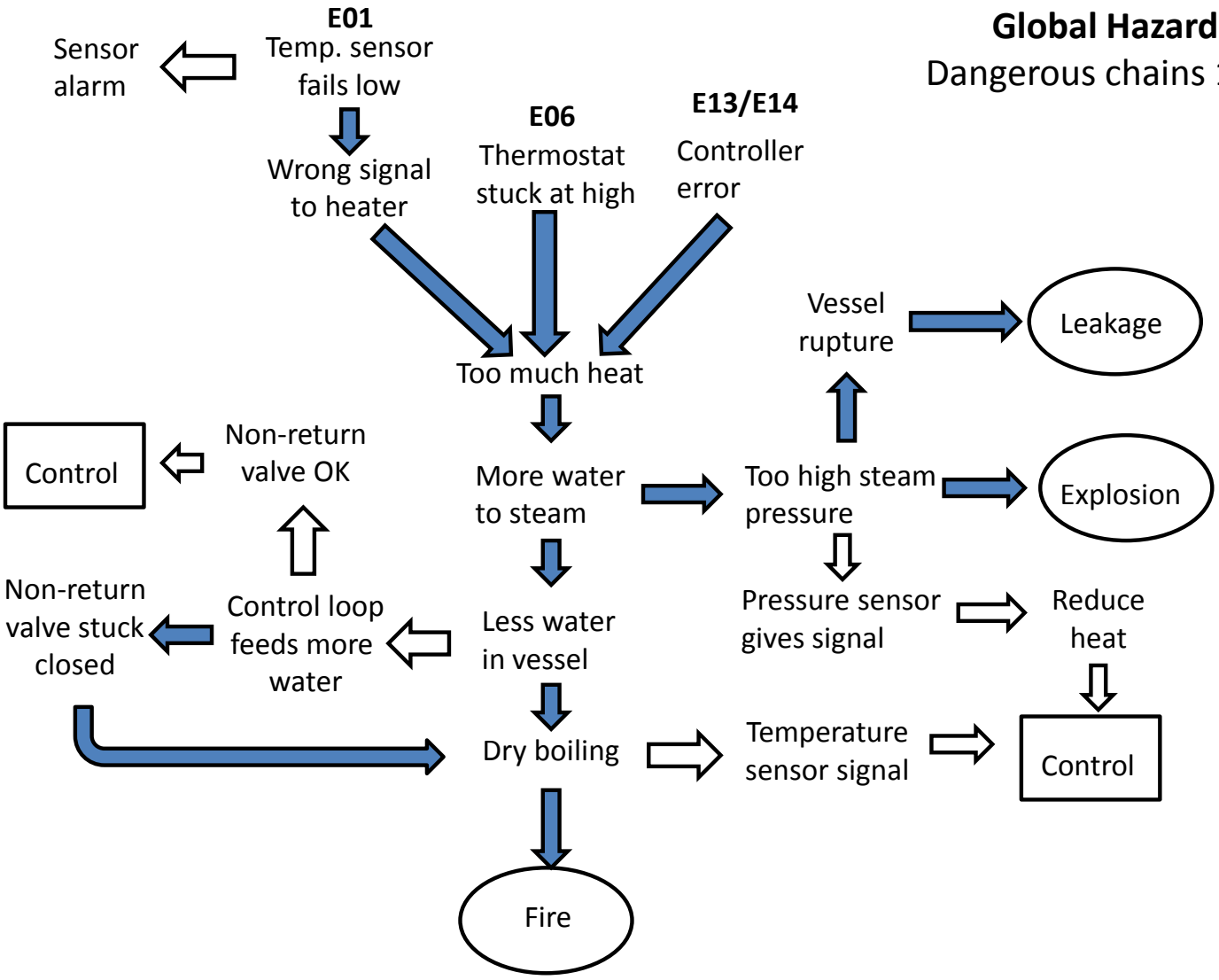
Post-experiment questionnaire

T2 = “The technique of human failure modes”

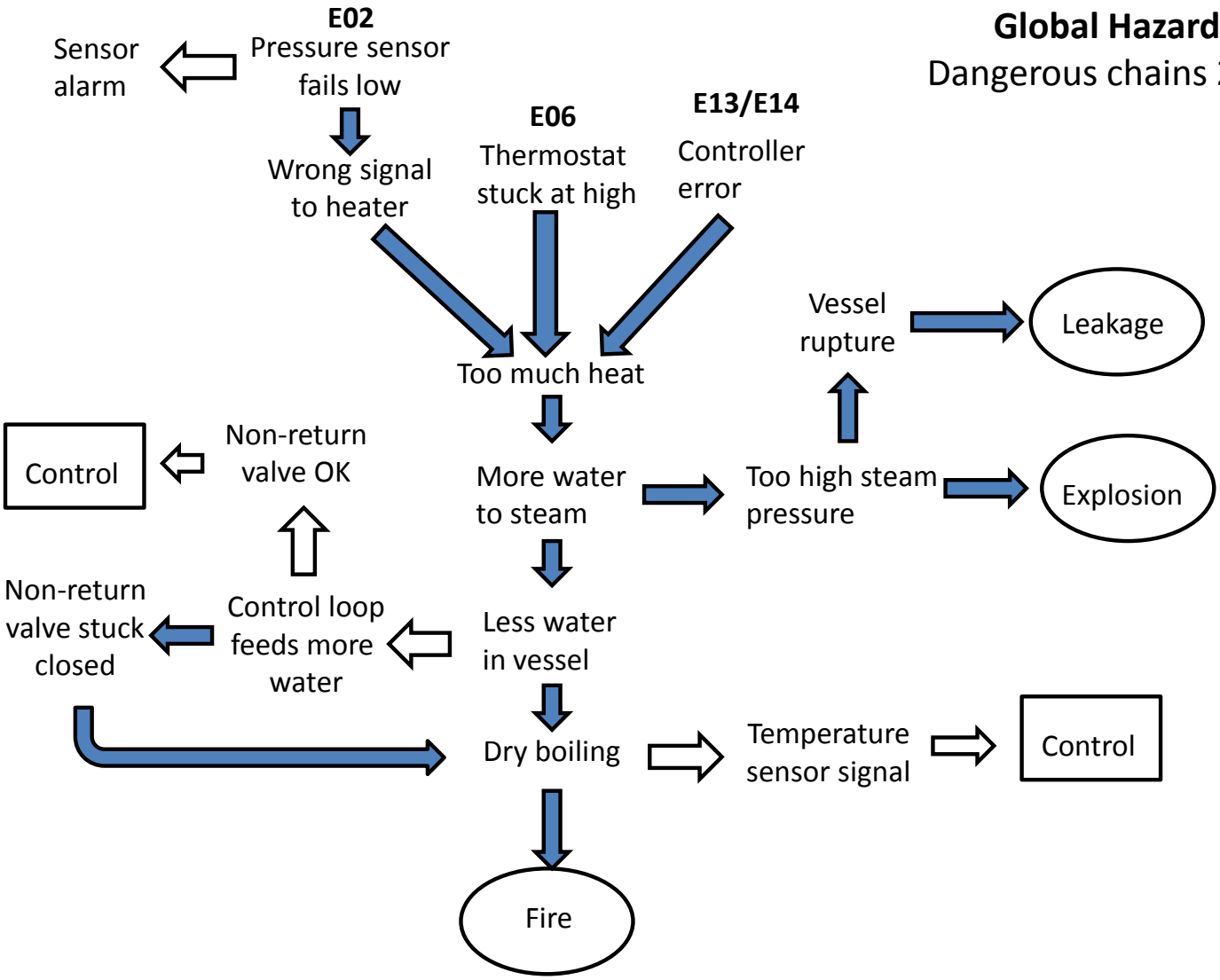
	Disagree strongly	Disagree	Neutral	Agree	Agree strongly
Q1. I found T2 easy to learn	☐	☐	☐	☐	☐
Q2. I found T2 easy to understand	☐	☐	☐	☐	☐
Q3. T2 helped me identify threats I otherwise would have missed	☐	☐	☐	☐	☐
Q4. T2 made me find threats more quickly than if just using common sense	☐	☐	☐	☐	☐
Q5. T2 made the discovery of threats more systematic	☐	☐	☐	☐	☐
Q6. If I need to identify safety threats in a future project, I would use T2 if possible	☐	☐	☐	☐	☐
Q7. I was confused about how to apply T2 to the problem	☐	☐	☐	☐	☐
Q8. T2 worked well as a basis for discovering threats	☐	☐	☐	☐	☐
Q9. The use of T2 made me less creative	☐	☐	☐	☐	☐
Q10. T2 caused me to waste time on threats of minor importance	☐	☐	☐	☐	☐

A.2 Can-cause chains

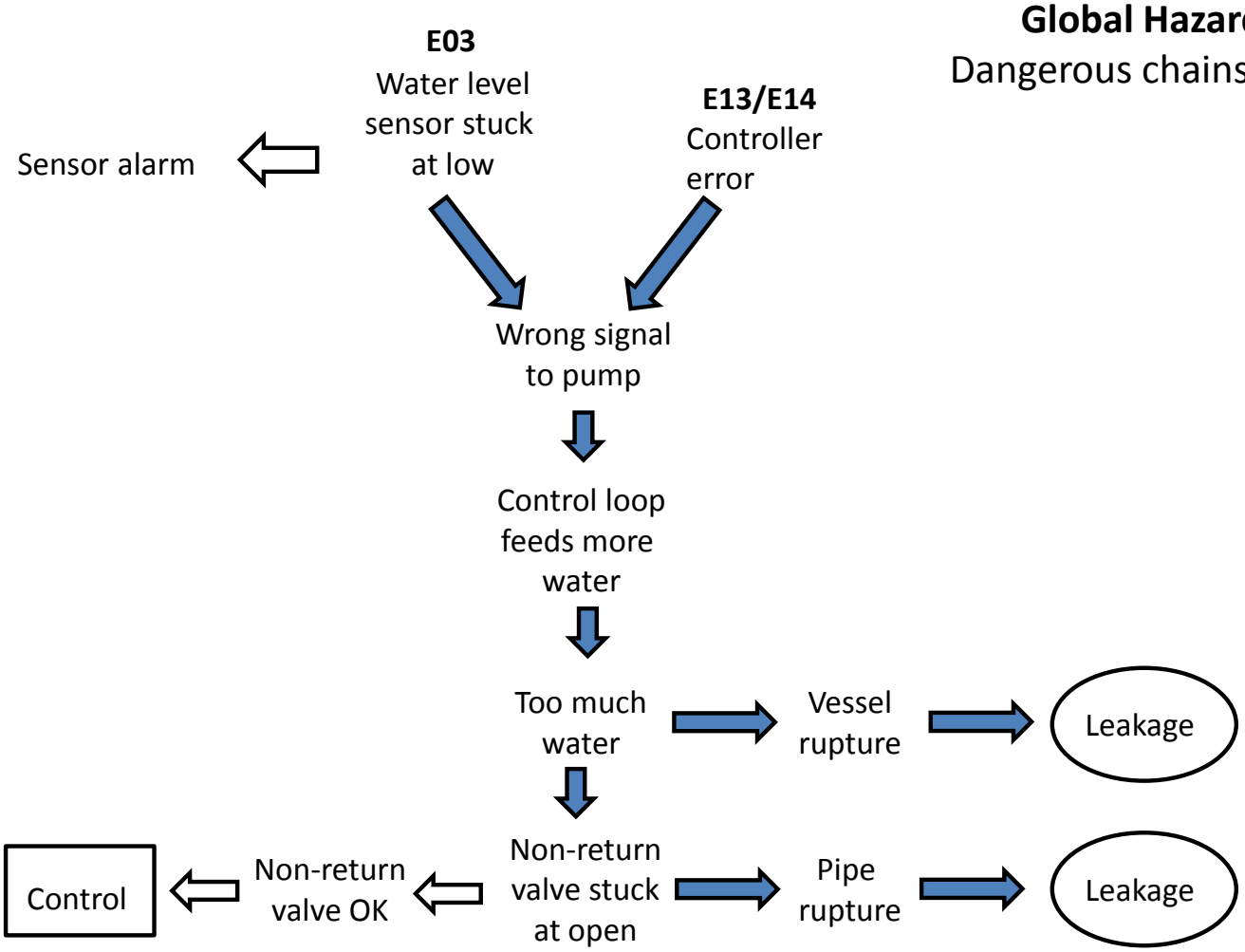
Global Hazards
Dangerous chains 1



Global Hazards
Dangerous chains 2

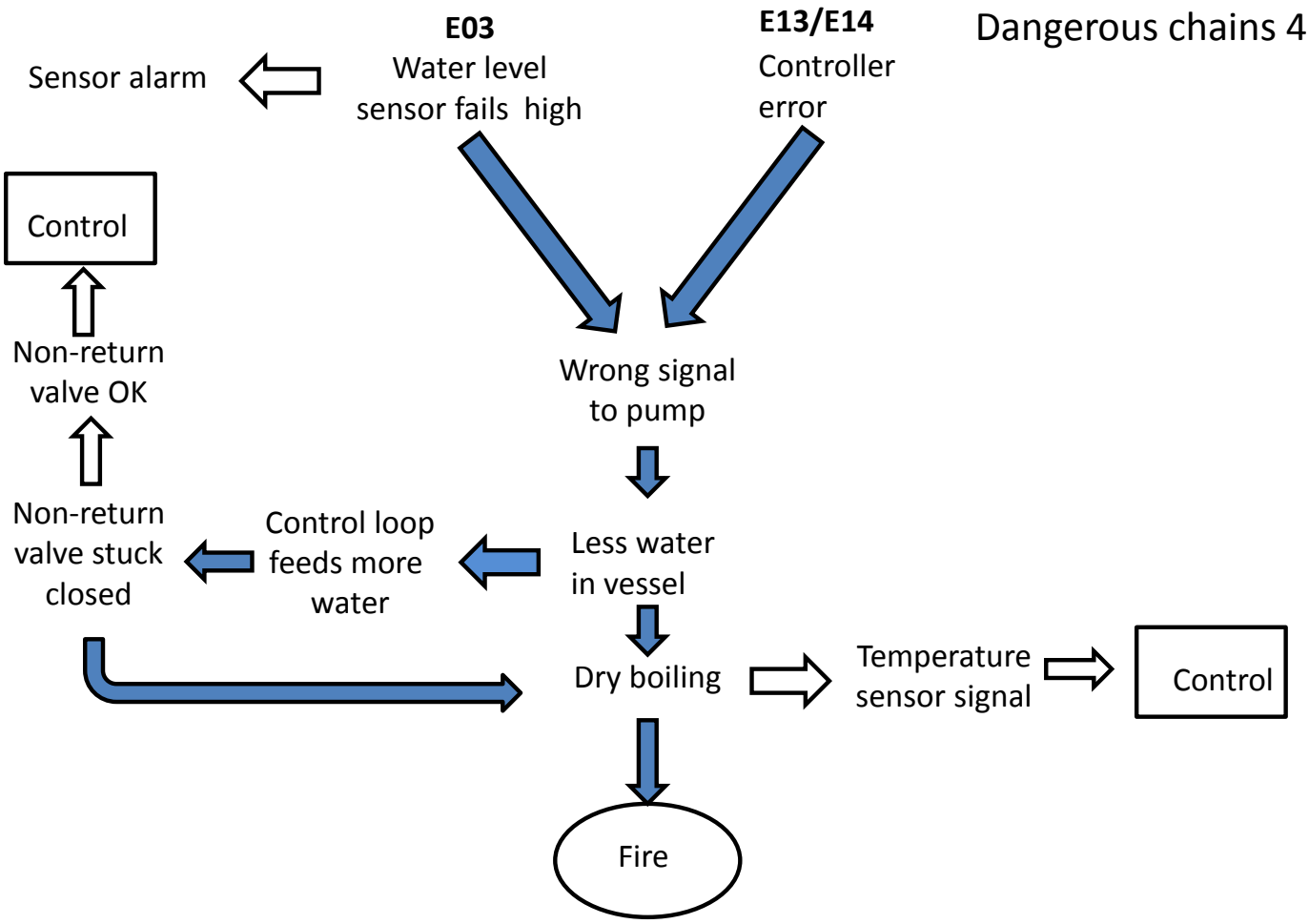


Global Hazards
Dangerous chains 3

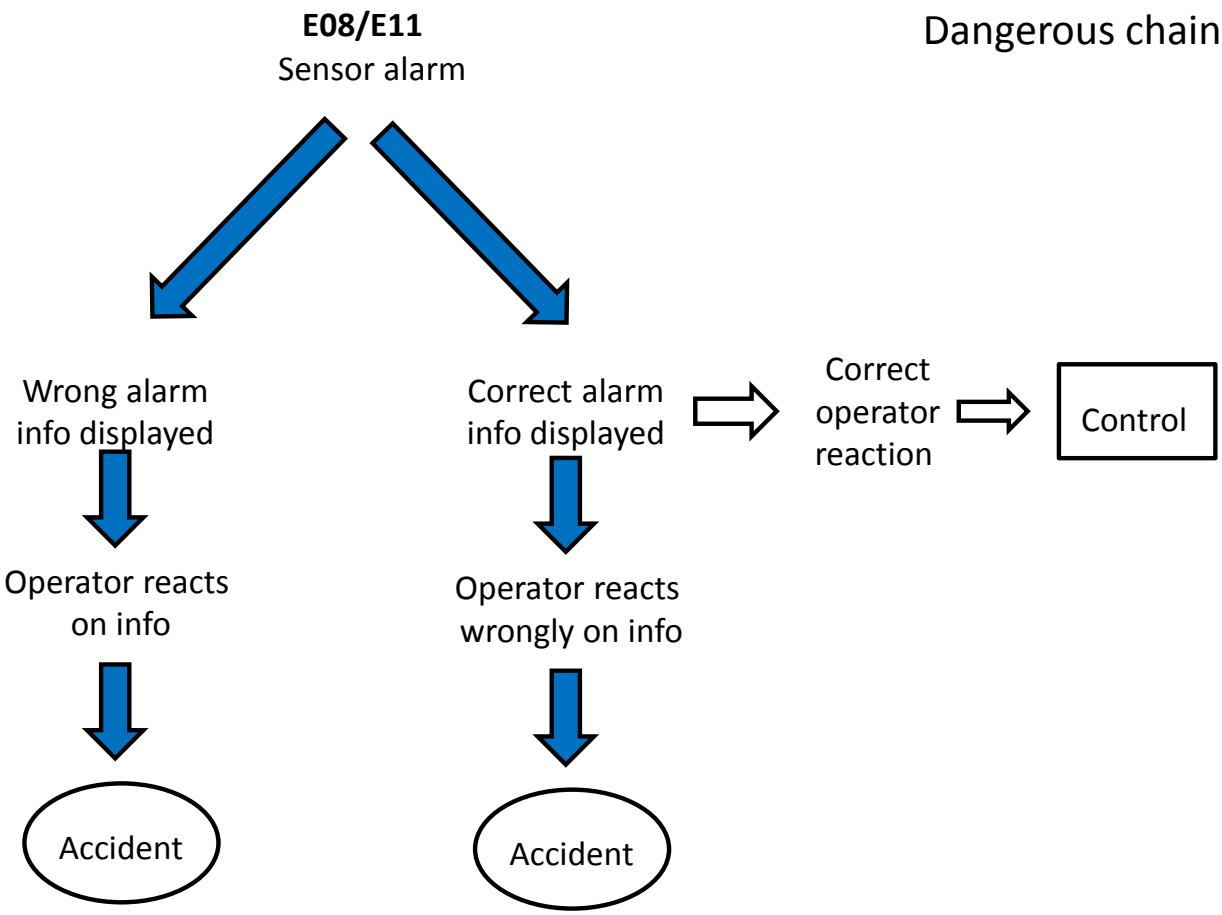


Global Hazards

Dangerous chains 4



Global Hazards
Dangerous chains 5



A.3 Accidents for a simple steam boiler

Accidents for a simplified steam boiler

We need to control two sources:

- Pressure. Too high pressure can lead to explosions or vessel rupture. The latter will lead to water leakage
- Heat. Too much heat, which again can lead to fire or to too high a pressure

There are two ways to control the system and to prevent accidents:

- Two control loops – one for water level and one for temperature and pressure. Each loop has one or more sensors and a mechanism for control.
    - Temperature / pressure sensor and a thermostat
    - Water level sensor and a feed-water pump
- An operator who receives info from the sensors and can override the control system by e.g. turning off the heat or stopping the pump.

If the operator cannot increase the heat or pressure, an operator error alone cannot cause an accident.

We will assume that all sensors can fail in one out of three ways:

- Stuck at high – indicates a higher temperature than the real one and will not be influenced by changes in its environment
- Stuck at low – indicates a lower temperature than the real one and will not be influenced by changes in its environment
- Failing – no signal is received. We will further assume that the control system always discovers this and informs the operator. It is, however, up to the operator to take appropriate action.

Initiating events:

- Sensor errors
    - Temperature – system adds more heat than necessary
    - Pressure sensor – system does not react on too high pressure
    - Water level – system does not react on too much or too little water
- Other equipment errors
    - Non-return valve error – prevents water from entering the vessel or do not prevent water from going back into the feeding pipe.
    - Heating element and thermostat – give too much heat or cannot be turned off
    - Control system error – gives wrong signal to feeding pump or heating element or display incorrect info to the operator.

A.4 Problems identified for train control system

- OP – Operator problems
  - 100 Incoming messages
    - 101 Receives message too late
    - 102 Misunderstand message, info or request. May also be due to e.g. language problems or cultural differences.
    - 103 Do not receive message
    - 104 Incomplete message received from system
    - 105 Incomplete message received from other person
    - 106 Ignores message or alarm
  - 200 Operator action
    - 201 Wrong action or non-existing or wrong command
    - 202 Unable to perform action
    - 203 Fails to ack
    - 204 Wrong ack
    - 205 Responds too late
    - 206 Fails to respond – completely or partly
    - 207 Fails to perform
    - 208 Wrong response
    - 209 Enters wrong info
    - 210 Forget to perform action – whole of part of
    - 211 Do not save info – e.g. log info
    - 212 Do not save changes to schedule etc.
  - 300 Operator scheduling
    - 301 Wrong maintenance scheduling
    - 302 Wrong train scheduling
    - 303 Fails to schedule
    - 304 Fails to report
  - 400 Operator equipment problems
    - 401 No alarm signal given
    - 402 Control panel error
  - 500 Operator knowledge
    - 501 Wrong situation analysis
    - 502 Fail to discover dangerous situation
    - 503 Wrong interpretation of system's functionality
    - 504 Lack of training
    - 505 Lacking info – all or in part
  - 600 Operator overload, e.g. due to panic
- CS – Computer system problems
  - 100 Does not save or deletes info
  - 200 Demands a wrong action
  - 300 Reacts wrongly to command or do not react at all
  - 400 Shows wrong info, including false alarms
  - 500 Unavailable – e.g. due to network problems
  - 510 Down – e.g. due to crash
  - 600 Shows no info or no alarms
  - 700 Other software errors
- EP – Engineer problems
  - 100 Incoming messages
    - 101 Receives message too late

        - 102 Misunderstand message, info or request. May also be due to e.g. language problems or cultural differences.
        - 103 Do not receive message
        - 104 Incomplete message received from system
        - 105 Incomplete message received from other person
        - 106 Ignores message or alarm
    - 200 Engineer action
        - 201 Wrong action or non-existing or wrong command
        - 202 Unable to perform action
        - 203 Fails to ack
        - 204 Wrong ack
        - 205 Responds too late
        - 206 Fails to respond – completely or partly
        - 207 Fails to perform
        - 208 Wrong response
        - 209 Enters wrong info to operator
        - 210 Forget to perform action, do not notice the need to perform action
        - 211 Do not save info – e.g. log info
        - 212 Do not save changes to schedule etc.
    - 300 Engineer scheduling
        - 301 Wrong maintenance scheduling
        - 302 Wrong train scheduling
    - 400 Engineer equipment problems
        - 401 No alarm signal given
        - 402 Control panel error
    - 500 Engineer knowledge
        - 501 Wrong situation analysis
        - 502 Fail to discover dangerous situation
        - 503 Wrong interpretation of system's functionality
        - 504 Lack of training
    - 600 Engineer overload, e.g. due to panic or stress
- TC – Technical communication problems
    - 100 Technical problems with telecom equipment
    - 200 Bad signal coverage or poor radio signal
    - 300 Busy line
    - 400 Other technical communication problems
- TE – Track-side equipment problems
    - 100 Wrong signal set
    - 200 Signal equipment fails