NTNU

Norwegian University of
Science and Technology

Testing of safety mechanisms in software-intensive systems

Arne Bjørgan

Master of Science in Computer Science
Submission date: June 2011
Supervisor: Tor Stålhane, IDI

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

Problem Description

Testing, unlike debugging, should give us confidence that the system is working properly. Usually testing focuses on what the system should do, and forgets that there are events that should not occur due to safety requirements.

Safety analysis methods give us measures to prevent, limit and reduce consequences. This thesis should look at which test methods and test environments that fit testing the different safety mechanisms best. These mechanisms are defined from safety analysis output.

How can the barrier model – prevent, control, reduce consequences - be used to define the test cases for a safety system?

Assignment given:	17. January 2011
Supervisor:	Tor Stålhane

Abstract

As software systems increasingly are used to control critical infrastructure, transportation systems and factory equipment, the use of proper testing methods have become more important. Systems that can cause harm to people, equipment or the environment they operate in are called safety critical systems.

The suppliers of safety critical systems make use of safety analysis methods to investigate possible hazards. The output from the analysis is possible causes and effects of the hazards found. These results are a large part of the basis for writing safety requirements for the system.

The safety requirements should be tested thoroughly to avoid accidents. It is important that the right testing technique is applied to test these systems. The consequences of a system failure can be high, and it is thus crucial to make use of a testing technique that has an approach that fits safety testing.

Barrier models and safety analysis results are often used to help writing test cases for safety critical systems. This thesis will present an experiment that study several testing techniques, and how they can use the information given by safety analysis results and the barrier model to write test cases.

Preface

This is a report made by Arne Bjørgan in the course *TDT4900 Computer and Information Science, Master Thesis*, written during the spring of 2011 at the Norwegian University of Science and Technology (NTNU).

I would like to thank my supervisor, Tor Stålhane, for help throughout the whole semester. He has contributed with valuable insight into the field of safety critical systems, as well as report writing in general. This help has been highly appreciated.

Trondheim, June 16, 2011

Arne Bjørgan

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

This chapter provides report outline and project context.

1.1 Report outline

This report consist of four parts. First we present the the **Preliminary Study**. This part is meant as preparation for the experiment that shall be done. As well as presenting the needed theory, this part shall decide which safety analysis method and testing techniques to use in the experiment. The second part presents the **Experiment Planning and Operation**. This part starts out with presenting the two research questions. The main chapter presents the available documentation for the steam boiler system, that shall be used in the experiment. Afterwards the experiment participants and groups are presented, as well as the available information for each group. The experiment execution is also presented. The last chapter guides the reader through the assessment of the deliveries, that was done in several stages. The third part presents **Results and Discussion**. In addition, it states possible sources of fault. The fourth and last part contains **Conclusions and Further Work**.

1.2 Project context

The work is done in cooperation with the CESAR project at IDI, NTNU. The safety system presented in the experiment is documentation of a steam boiler safety system provided by ABB.

Part I

Preliminary Study

CHAPTER 2

Safety analysis methods

This chapter will present safety analysis methods and decide which one that suits the experiment best.

2.1 Introduction

Safety analysis methods are used to find and prioritize hazards for a given safety critical system. There are a wide variety of methods available. The most used methods are listed in the next section.

2.2 Safety analysis methods

The most used safety analysis methods are listed below. They are discussed further in [4]. These methods are compared to decide which one fits the experiment best. See table 2.1.

- Preliminary Hazard Analysis (PHA)
- Failure Mode Effect Analysis (FMEA)
- Fault Tree Analysis (FTA)
- Hazard and Operability studies (HazOp)

2.2.1 Comparison of safety analysis methods

The methods listed above use different approaches to find the hazards, causes and effects. They are also used in different development phases. This thesis looks at how use of safety analysis results is useful for writing test cases for a safety system. The system which is used in the upcoming experiment is a steam boiler which only exists in a concept phase. Because of this, the upcoming experiment regarding test case writing should make use of a safety analysis method that fits best for the concept phase of development. Since the experiment participants have limited time to solve the task, it is important that the analysis results are easy to understand.

Safety analysis method	Project phase	General/Specific	Top-down/Bottom-up	Understanding the output
PHA	Early	General	None	Easy
FMEA	Middle	Both	Top-down	Difficult
FTA	Middle	Specific	Bottom-up	Difficult
HazOp	Early	General	None	Easy

Table 2.1: Comparison of safety analysis methods

The comparison above is based on statements taken from [5]. Given the comparison above, PHA fits best for the experiment. PHA is suitable for early project phases. The article found in [6] discusses the use of PHA for prototype analysis, and states: "Sometimes even a simple picture or diagram can help show potential hazardous situations".
PHA's general approach to finding hazards gives us a result that is only limited by the knowledge of the group performing the PHA. PHA results are presented in a table that is easy to understand without any previous knowledge of the method. Since most of our experiment participants do not have any prior experience with safety analysis results, it is crucial for the experiment that the results are understandable in a short matter of time. PHA will be explained in detail in the next section.

2.3 Preliminary Hazard Analysis (PHA)

The preliminary hazard analysis (PHA) technique is a safety analysis tool for identifying hazards, their associated causal factors, effects, level of risk, and mitigation design measures when detailed design information is not available. [7] For every hazard there should be defined preventive actions to prevent accidents from occuring. PHA is typically performed in the early project phases. PHA is performed by a team, which should have different backgrounds. There should be domain experts, future users, safety experts and software/hardware experts. Diversity in expertise is needed to be sure that as many hazards as possible are found. Different people will have different viewpoints, and typically find different kinds of hazards. Every hazard found should be connected to a set of causes and effects. There can be one or more of both. Defining the causes and effects are important, both to decide the severeness of an accident, and to define preventive actions. PHA can also be combined with fault tree analysis. FTA, with its bottom-up manner, investigates a tree of linked causes that leads to accidents. By combining PHA and FTA, the team can look at the hazards from different angles. This yields more insight into the hazards, and will help in developing the safety system as good as can be.

2.3.1 PHA results

The result from a PHA is a table with one row or more per hazard. The table should, as a minimum, consist of the following columns:

- **Hazard**
  The name of the hazard.
- **Causes**
  The cause(s) for this hazard.
- **Effects**
  The effect(s) of this hazard.
- **Preventive actions**
  Preventive actions against the hazard.

It is important to detect every possible cause and effect for each hazard. To define preventive actions, every cause needs to be covered. If we forget to identify and handle one cause, it may result in accidents because of safety system failure. In the example provided in table 2.2, we have added a column that describes which barrier the respective preventive action is realized in.

2.3.2 PHA - An example

This example is from ABB. The table showed in 2.2, is the result of a PHA performed on a steam boiler that operates on a factory site.

Hazard	Cause	Main effect	Preventive action	Realization
Too high pressure in the tank	Not able to turn off the heating (sensor, control, actuator, connections)	Boiler explodes	Safety valve	Original requirement
			Turn off the heat	Detect and control
	Feeding pump failure (too strong)	Boiler rupture	Turn off power to the feeding pump	Detect and control
Too high water level	Water level regulation failure (sensor, control, actuator, connections)	Water to the process	Pump emergency stop	Detect and control
Too high pressure in the feed pipe	Non-return valve failure	Release boiling water to the water supply	Two non-return valves in series	-
			Emergency valve for releasing pressure	Original requirements
The tank is too hot	Too little water and too much heat (sensor, control, actuator, connections)	Tank gets hot/fire	Turn off the heat	Detect and stop
			Add water?	-
Unintentional leaks	Corrosion	People get scalded	Inspection, collector tray or quality assurance	-
	Bad welding/fittings	People get scalded	Inspection, collector tray or quality assurance	-
Electric shock	Short circuit	People get hurt/killed	Fuses	-
Flooding	Breakage in pipes	Damage to equipment and/or environment	Flow meter, collector tray	-

Table 2.2: Result from PHA performed on ABB's steam boiler [3]

As we can see from this table, each hazard can have multiple rows for causes, effects and preventive actions. The rightmost column names the realization of the preventive action, which is the barrier that implements the preventive action. The use of barriers are explained in the next chapter.

CHAPTER 3

Barrier model

This chapter presents the barrier model, and how it is applied to prevent accidents in safety critical systems.

3.1 Introduction

The barrier model refers to the safety mechanisms implemented to avoid dangerous behaviour in safety critical systems.

3.2 Anatomy of an accident

We can often see that an accident happens after a series of events occuring in software or hardware. Green's anatomy of an accident, showed in figure 3.1, explains how a system escalates from normal operation to an accident through a series of events.

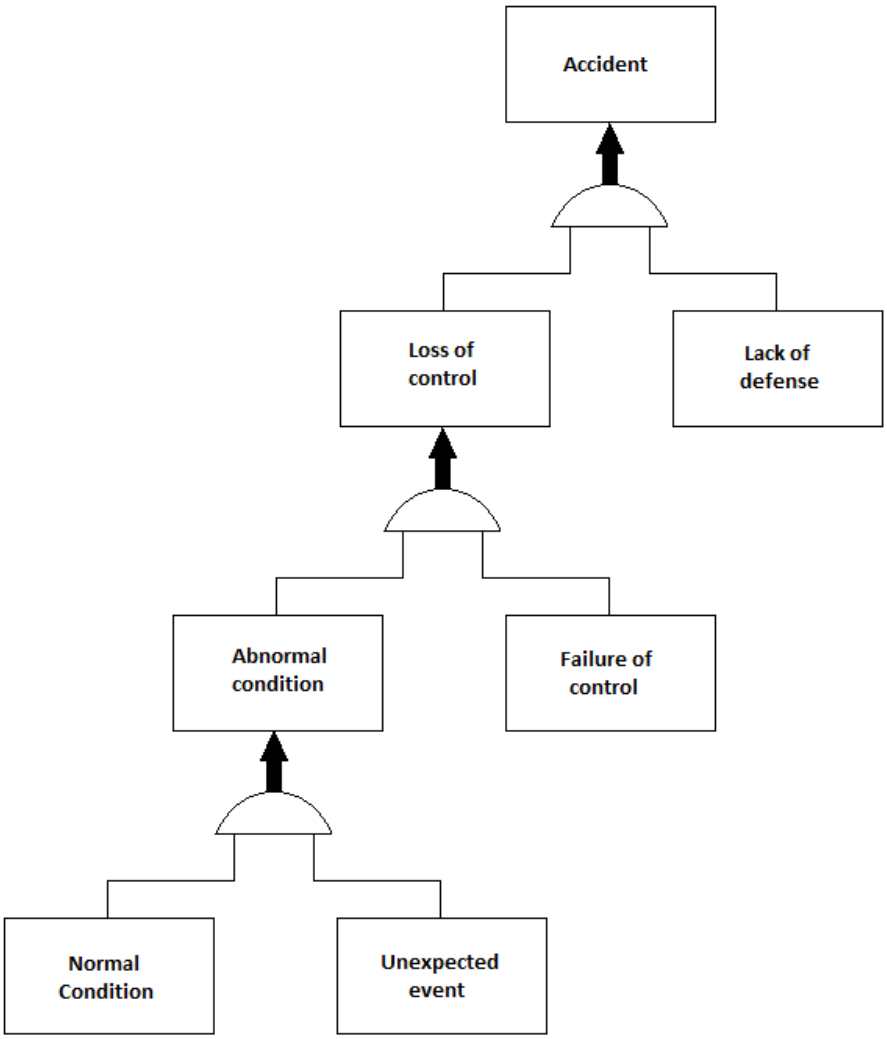


Figure 3.1: Green's anatomy of an accident [1]

It is important to understand that an accident often occurs after multiple barriers have failed to work as intended. These barriers are safety measures implemented in the system. The barriers should form a defence against accidents occuring, and are implemented in different levels.

3.3 The barrier model

The barrier model presents several levels of barriers to minimize or remove the consequences of faulty events in a safety critical system.

- **Failure prevention**
  Prevent a hazard from leading to a problem
- **Failure control**
  Prevent a problem from causing a dangerous event
- **Failure recovery/reduction**
  Reduce the effect of a dangerous event if it can not be prevented

These three types of safety measures can be seen in figure 3.1. A barrier is an obstacle, an obstruction, or a hindrance that may either prevent an action from being carried out or an event from taking place, or thwart or lessen the impact of the consequences. [8] Failure prevention should keep the system in normal condition. Unexpected events should be avoided, by monitoring the system adequately. If this fails to work, an unexpected event might occur. The system then reaches an abnormal condition. This is where failure control comes into the picture. Control measures should be implemented to avoid failure of the system. The system should be able to detect the abnormal condition, analyze it and perform action on actuators so that the system can be put back into normal operation. If the control measures fail to work, we have loss of control over the system. We now have a situation where we are close to an accident. This is where failure reduction should be put into action. The system should understand that the failure control system has failed, and perform actions that put the system to a stop. This can be done by cutting the power or other drastic measures. This is a last way out, and if no action is executed at this stage an accident is close. This is shown in figure 3.2.

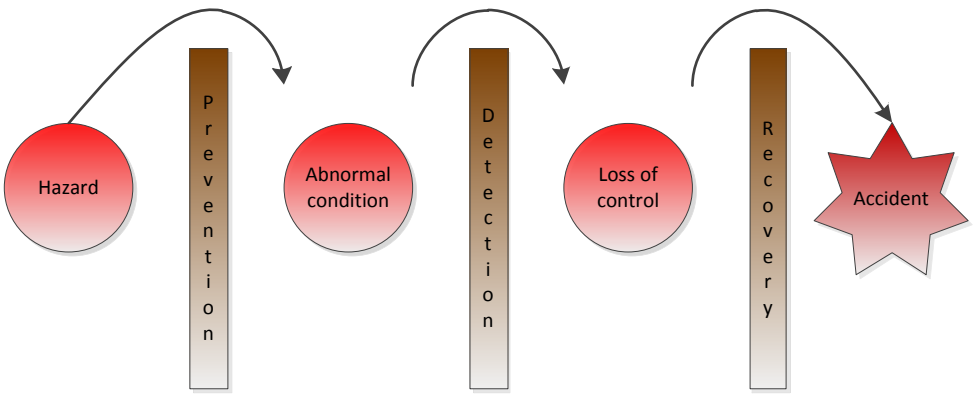


Figure 3.2: Barriers against accidents

It is crucial that these levels of safety measures cooperate to avoid accidents. The barriers should be implemented so that multiple barriers are working against every possible hazard found in the safety analysis. Operating procedures for handling crises should be defined with regards to the barriers implemented in the system. [9]

3.3.1 Failure prevention

Failure prevention is the first barrier to avoid accidents. Prevention mechanisms are implemented in the system, to prevent faulty events from occuring at all. An example of a failure prevention mechanism is a safety valve in a steam boiler, that should open if the pressure exceeds a set limit.

3.3.2 Failure control

Failure control is the next barrier. When a faulty state has occured, it is important to detect and control this as soon as possible to put recovery measures to effect. Detection measures for hardware can for example be a heartbeat given from one of the hardware units. This heartbeat is checked for arrival at a given time interval. When the heartbeat is not received within this interval, we can assume that the hardware unit has failed. It is now crucial to recover from this faulty event before it escalates towards an accident.

3.3.3 Failure recovery

Failure recovery is used to recover the system from a faulty state. For every faulty event found in safety analysis, there should be a possibility to recover control over the system before accidents occur. One simple example of a recovery measure is to have an emergency unit that can shut down the system.

3.4 Barriers - an example

This section will provide an example of how we can write safety requirements from a hazard analysis. The example is taken from [4], and deals with safety in Air Traffic Management (ATM). The ATM system uses different barriers to prevent accidents from happening. To simplify, we only consider the hazard of a possible mid-air collision (MAC) between two aircrafts. For this hazard we should have different safety measures for different levels, as we have described in the previous section. An accident can only occur if all these measures fail to work as intended. The consequence of this hazard occuring is obviously high.

3.4.1 MAC prevention measures

- **Airspace design:**
  Structuring the airspace so as to keep aircrafts apart spacially, in the lateral and/or vertical dimensions

- **Conflict avoidance:**
  Planning the routing and timing of individual fligths so that the aircraft, if they follow their planned trajectories, would not pass each other within the prescribed minimum separation

3.4.2 MAC detection/recovery measures

- **Conflict resolution:**
  Detecting conflicts when they do occur and resolving the situation by changing the heading, altitude or speed of the aircraft appropriately

3.4.3 Description of the mechanisms

As shown in figure 3.1, we have different types of mechanisms. Both prevention and detection/recovery mechanisms are implemented in the system to provide different levels of protection against accidents. The measures listed are separated into two different services. A **Separation Provision** service controls the flow of traffic within the declared capacity. **Collision Avoidance** is intended to recover the situation only for those potential accidents that **Separation Provision** has not removed from the system. The mechanisms differ in how they are controlled; either by human, machine or a combination of the two.

- **Air Traffic Control** recovery mechanisms - human and/or machine-based safety nets
- **Pilot** recovery mechanism - again, human and/or machine-based safety nets
- **Providence** - i.e pure chance

The barriers mentioned are not 100% effective even when working to specification. This means that we are always dealing with some minimized risk, but this level must be tolerable. Often these levels are set by laws regarding air traffic, and can therefore vary according to the country where the system is deployed. [10]

CHAPTER 4

Test environments

This chapter presents test environments used for testing safety critical systems.

4.1 Introduction

Several test environments are used for testing software systems. Safety critical systems often require simulated environments so that we can test the system in a safe manner. This chapter will present a discussion on how safety critical systems can be tested both in simulated and real environments.

4.2 Interface specification

The system under test should use the same interface towards the test environment, regardless of whether it is a simulated or real environment. Figure 4.1 shows how the system should be connected to the environments.

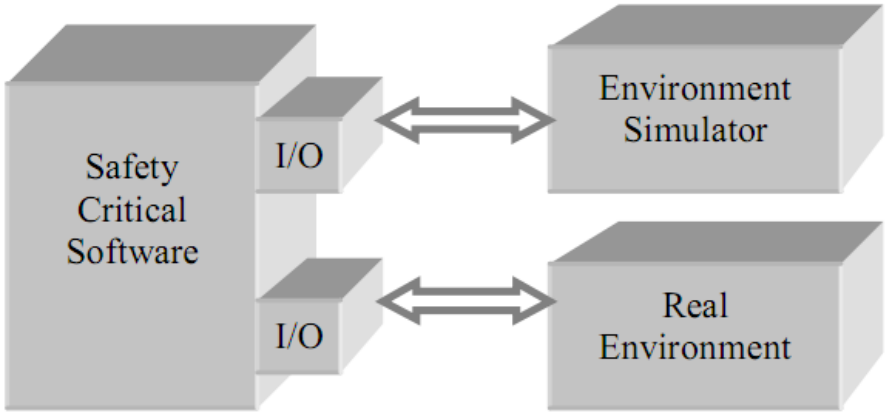


Figure 4.1: Interface against test environments [2]

As we see, the environments are treated as equal environments. This is important to achieve valid test results from the environment simulation testing.

4.3 Environment simulator testing

Many safety critical systems can not be tested in full operation before late development stages, because of the possibility of an accident. In these cases, the system needs to be run in an artifical environment, where the agents and sensors are simulated. It is important that the artifical environment represents the real life environment, so that the test results are valid. When performing testing in a simulated environment, there should be a clear and consise interface definiton available. All contact points with the environment surrounding the system when put into operation must be correct, so that the system perceives the test environment as the real life environment. The simulated environment is connected to the system in the same way as the real environment should be connected.
Other factors that decide if simulated test environments should be used are cost and development stage. Some safety critical systems rely on cooperation with expensive hardware or software that might not be available at an early stage of testing. If

this hardware or software can be simulated, money can be saved for later stages of testing. Also, if the project is in an early stage, there might be only high level design available. Testing in a simulated environment can confirm if the design suits the requirements. In this way early versions of the systems can be tested without large expenses for hardware. The design can be validated or rejected based on these tests. This is a cheap way of testing different system designs whitout the need of sensors, actuators and other hardware.

4.4 Real environment testing

As the project goes on and more system characteristics are well defined, full operation testing can be done. The real environment should be connected with the same interface that the simulated environment was, as shown in figure 4.1. In this way, we have test documentation about the systems characteristics before the operational testing begins. The operational testing can be done on site or in an similar environment if this is available.
Testing on site is often done for acceptance testing. It is important for the company responsible for development, testing, or both, that this testing gives the buyer or potential buyer confidence in the system. If environment simulator testing is lacking, or badly done, there is a big risk for the system to fail, at least partly, on site. This does not give a lot of confidence, neither in the system, or the responsible company.

4.5 Comparison of test environments

A comparison of the two different test environments is provided below.

Criteria	Environment simulation	Real environment
Project phase	Earlier stages	Later stages
Cost	Cheaper	More expensive
Validity	Less	More

Table 4.1: Comparison of test environments

As the earlier descriptions of the environments suggest, a combination of these would be the best in most situations. Environment simulation gives the possibility to try out early system designs without high costs. In addition it prepares the system to be tested for full operation on site. Operation testing is often done on delivery as a part of acceptance testing for the buying company.

CHAPTER 5

Test methods

This chapter presents test methods used for testing safety critical systems.

5.1 Introduction

A number of test methods can be used for testing safety critical systems. This chapter presents two test cathegories for such systems, and the most important test methods that belong in these categories.

5.2 Specific and general approaches to testing

When testing, one of the approaches is to test specifically on known hazards. When testing in this manner, we want to see how the system handles hazardous events. There should be barriers implemented against these faults, often in the three categories prevention, detection and recovery. The strength of this approach is that we test every known hazard systematically, and end up with a system that can handle all known faults. The weakness is in finding new faults. When testing a specific system state, it is difficult to reveal potential threats that are not yet found.
A general approach is strong where the specific approach is weak, and the other way around. If we use a very specific approach, we have little chance of finding new faults. This is where the general approach is strong. This approach has a much better chance of finding new, unknown faults. As [4] suggests, it may be better to use both approaches to both cover the known faults and find new ones. If we use the approaches in an iterative manner, we have a good chance at testing the system thoroughly. Testing can stop when we have tested all known faults, and no new faults are found during testing. We must also be sure that we have a high test coverage of the system, whether it is code or path coverage. The limit for sufficient coverage depends on the nature of the system, and should be set prior to testing. The limit may be increased while testing is performed, if we feel that the limit is set too low. It is not recommended to lower the limit, since this often may be a result of too little time to perform testing. It would be better to just delay the delivery date, to assure that the system is tested thoroughly.

5.3 Fault injection

In the last decade, fault injection has become a popular technique for experimentally determining the dependability parameters of a system, such as detection latency or fault coverage. [11] It provides the means for studying interactions between faults, errors, failures and fault-handling mechanisms. [12] Fault injection got a high score in the comparison done in [4], and due to the technique's approach towards testing faults it will be used in the experiment. Fault injection is best at testing the known faults. Fault injection is not good at finding new faults, and should therefore be used together with a more general test method to be able to find unknown faults. Fault injection tests the handling of known faults systematically. For each known fault, the testers should inject the fault into the system, either on an agent/sensor, or in the code, depending on the type of fault. They should then monitor the system, and keep track of what safety measures that are set into action to stop this hazardous state from developing into an accident. Testing is finished when the faults

known prior to testing are covered, and other additional safety measures needed is implemented and tested.

5.3.1 Hardware fault injection

Hardware fault injection is one way of performing fault injection testing. It is performed by manipulating the system hardware, either with or without direct hardware contact.

- **Hardware fault injection with contact**
  Hardware fault injection with contact can be done by damaging the hardware directly, for example by cutting an electrical circuit or damaging a hardware component.

- **Hardware fault injection without contact**
  Hardware fault injection without contact can for example be done by using electrical induction to create noise or faults in the system hardware. When we do not have direct contact, it may be difficult to test specific faults, but one bonus is that other faults can be found by coincidence. [13]

5.3.2 Software fault injection

Software fault injection manipulates software to achieve hazardous states in the system under test. The fault is inserted either before or after the system is put into operation. Prior to operation the fault can be inserted into both uncompiled and compiled code. When the system is put into operation, the testers should monitor if the fault actually is executed, and how the system uses safety measures to handle the fault and put the system back into a normal state. [13]

5.3.3 Fault injection test case

Table 5.1 shows an example test case using the fault injection technique.

Test case number	0
Name	Speed too high
Pre-condition	The speed sensor reads a too high value
Pass criteria ***(how should the system react to this?)***	1. Decrease speed so that 2. The current speed is below or equal to the set speed
Textual description	Testing is done by manipulating the speed sensor so that it reads a too high speed. The system should react by decreasing the speed so it is under or equal to the set speed.

Table 5.1: Example fault injection test case

5.4 Boundary value analysis/Equivalence partitioning

BVA/Equivalence partitioning is chosen as the second testing technique to be used in the experiment. This technique got the second highest score in the comparison done in [4], only beaten by fault injection. Boundary value testing is best fitted for testing where we have a limit value set in the requirements. When testing in this manner, we want to execute loops below, right at and right above their boundary conditions.
If a requirement contains a value, this is the value that represents the end of one partition and the start of another partition. The highest value below the boundary is a maximum value, while the lowest number above the boundary is a minimum value. They are both boundary values. Testing should be done so that the values fall on one side or the other, and see if the predicted system response is generated. Safety critical systems often operate with boundary values, so the nature of this technique makes it well suited for testing safety critical software. [4] Typical values used to test the extremities, according to [14]:

- Min (Minimal)
- Min+ (Just above minimal)
- Nom (Average)
- Max- (Just below maximum)
- Max (Maximum)

In this way every possibility is checked, without having to check every possible value within the partitions. When the participants in the experiment shall write their test cases, it is expected that only the actual boundary is checked and not all of the typical values listed above. This is due to the participant lack of experience with this testing technique.

5.4.1 Boundary value analysis test case

Table 5.2 shows an example test case using boundary value analysis.

Test case number	0
Name	Speed too high
Pre-condition	current_speed>set_speed
Pass criteria ***(how should the system react to this?)***	1. Decrease speed so that 2. current_speed<=set_speed
Textual description	If the current speed is over the set speed, the system should decrease speed until the current speed is under the set speed

Table 5.2: Example boundary value analysis test case

Part II

Experiment Planning and Operation

CHAPTER 6

Experiment introduction

This chapter gives an introduction to the steam boiler experiment. It will also state the research questions for this experiment.

6.1 Introduction

The experiment investigates the process of writing test cases for safety critical systems. The experiment also investigates how the use of barriers in the software and the use of results from safety analysis of the system helps in writing test cases. In addition it compares two methods for testing the system, namely fault injection and equivalence partitioning/boundary value analysis.

6.2 Experiment goal

This experiment has two research questions:

- **RQ 1:** Which of the two test methods fault injection and equivalence partitioning/boundary value analysis is best suited for writing test cases for safety critical systems?
- **RQ 2:** Is the use of safety analysis results and barrier description helpful when writing test cases for a safety critical system?

CHAPTER 7

Steam boiler system

This chapter presents the steam boiler safety system which will be used in the experiment.

7.1 Introduction

The system used for this experiment is a steam boiler used as a pilot for ABB in the CESAR project. The steam boiler design exists in two different versions, before and after safety barriers are implemented. Functional and safety requirements, textual use cases, PHA results and implemented barriers are also available.

7.2 Requirements

The system has two sets of requirements; functional requirements and safety requirements. [3]

7.2.1 Functional requirements

FR1: The steam boiler shall be able to deliver steam to an industrial process.
FR2: The steam boiler shall be able to produce steam using electrical heating element.
FR3: The steam boiler shall be able to control steam pressure using thermostat of electrical heating element.
FR4: The steam boiler shall be able to control water level using feeding pump.
FR5: The feeding pump shall be able to deliver water using non-return valve.
FR6: If steam pressure greater than critical pressure level then the steam boiler shall open safety valve.

7.2.2 Safety requirements

SR1: If the water level is greater than max water level, the safety system shall stop the feeding pump.
SR2: If the steam pressure is greater than the max pressure level, the safety system shall stop the feeding pump.
SR3: If the external temperature is greater than the max external temperature, the safety system shall cut power to the heating element.
SR4: The user shall be able to inspect the steam boiler at a minimum rate of TBD times per year.
SR5: If the water flow is greater than the max water flow, the safety system shall stop the feeding pump.

7.3 Textual use cases

There are four textual use cases for the system.

Use Case Name	Drum temperature control
Actor	Main control system
Pre-condition	Safety control system activated
Post-condition	Drum temperature is below limit or power to heater has been disconnected and alarm has been set
Trigger	Time triggered – run every 10 seconds
Read outside drum-temperature sensor T_outside	
Is T_outside>T_outside-limit?	
Yes => send power-off signal to actuator (Contactor) set “Too high temperature” alarm	
End use case	

Figure 7.1: TUC1: Drum temperature control

Use Case Name	Water level control
Actor	Main control system
Pre-condition	Safety control system activated
Post-condition	Water level too high
Trigger	Water level sensor triggered
Is water level sensor triggered?	
Yes => send power-off signal to actuator (Water pump) Set “Too high water level” alarm	
End use case	

Figure 7.2: TUC2: Water level control

Use Case Name	Pressure control
Actor	Main control system
Pre-condition	Safety control system activated
Post-condition	Pressure too high
Trigger	Time triggered – run every 10 seconds
Read inside pressure sensor P	
Is P>P-limit?	
Yes => send power-off signal to actuator (Water pump) Set “Too high pressure” alarm	
End use case	

Figure 7.3: TUC3: Pressure control

Use Case Name	Water flow control
Actor	Main control system
Pre-condition	Safety control system activated
Post-condition	Water flow failure
Trigger	Time triggered – run every 10 seconds
Read flow meter FM	
Is FM>FM-max?	
Yes => send power-off signal to actuator (Water pump) Set “Water flow failure” alarm	
End use case	

Figure 7.4: TUC4: Water flow control

7.4 The original system

The original system design is a straight forward steam boiler design with safety measures implemented.

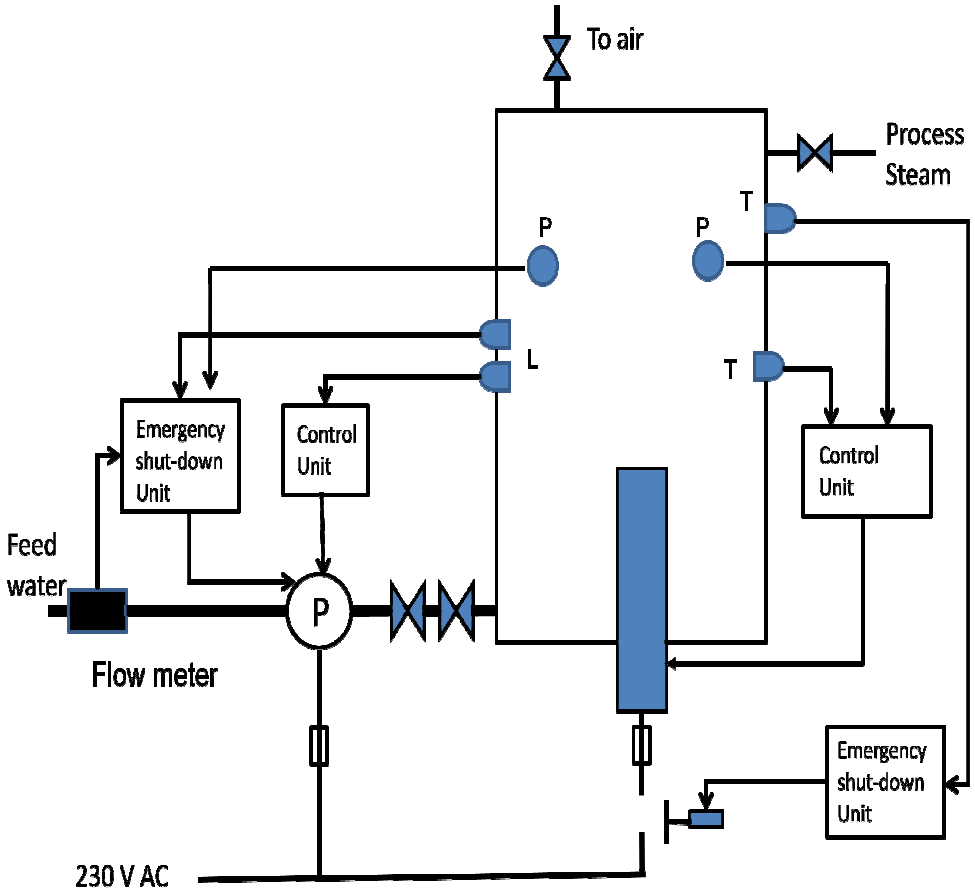


Figure 7.5: Steam boiler design [3]

The water pump, indicated by P, pumps water into the boiler. The heating element heats the water inside the tank, and steam is produced. There are several sensors which send signals to the control units.

- **Sensor L:** Is triggered when the water level exceeds a preset level
- **Sensor P:** Reads the pressure inside the tank
- **Sensor T:** Reads the temperature inside the tank

There are two control units which read this sensors, and execute actions according to the values they receive. The control units control two actuators; the water pump and the heating element. Also, there are two valves, one which delivers process steam, and one that sends excess steam to the air.

7.5 Results from safety analysis

The system has been the subject of a safety analysis. This safety analysis is conducted as a Preliminary Hazard Analysis(PHA). The result from this analysis is shown in table 7.1.

Hazard	Cause	Main effect	Preventive action	Realization
Too high pressure in the tank	Not able to turn off the heating (sensor, control, actuator, connections)	Boiler explodes	Safety valve	Original requirement
			Turn off the heat	Detect and control
	Feeding pump failure (too strong)	Boiler rupture	Turn off power to the feeding pump	Detect and control
Too high water level	Water level regulation failure (sensor, control, actuator, connections)	Water to the process	Pump emergency stop	Detect and control
Too high pressure in the feed pipe	Non-return valve failure	Release boiling water to the water supply	Two non-return valves in series	-
			Emergency valve for releasing pressure	Original requirements
The tank is too hot	Too little water and too much heat (sensor, control, actuator, connections)	Tank gets hot/fire	Turn off the heat	Detect and stop
			Add water?	-
Unintentional leaks	Corrosion	People get scalded	Inspection, collector tray or quality assurance	-
	Bad welding/fittings	People get scalded	Inspection, collector tray or quality assurance	-
Electric shock	Short circuit	People get hurt/killed	Fuses	-
Flooding	Breakage in pipes	Damage to equipment and/or environment	Flow meter, collector tray	-

Table 7.1: Results from PHA [3]

The different columns are described in the safety analysis chapter in the preliminary study.

7.6 Implemented barriers

The rightmost column in table 7.1 shows that there are two implemented safety barriers.

1. **Detect and control**
2. **Detect and stop**

Figure 7.6 shows how they should work together to prevent accidents.

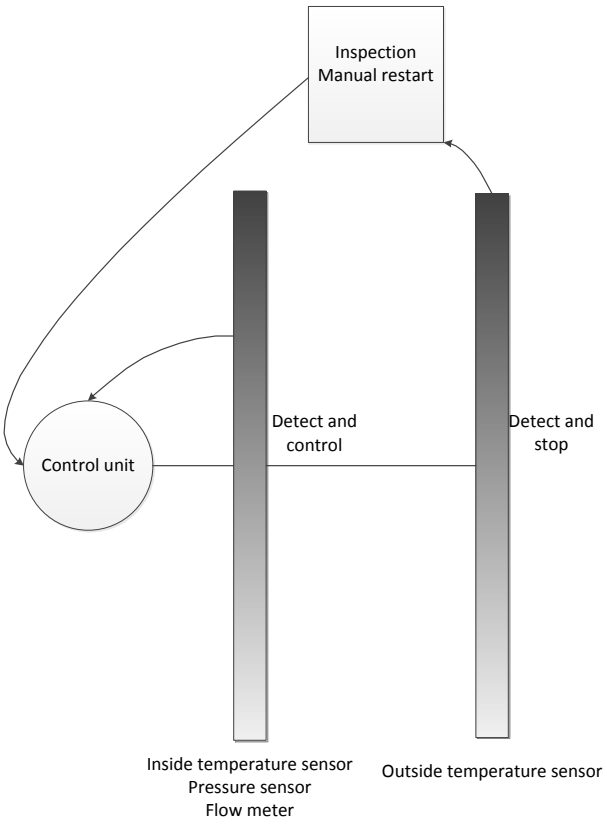


Figure 7.6: Two implemented barriers

The first barrier makes use of three sensors, and the second uses one. The first barrier is used for detecting and controlling faults in the system. It should be able to put the system back into a normal state after detecting a fault. If the first barrier fails, the second barrier will cut the power by using the emergency shut-down unit. This is a last measure to prevent accidents, and after this barrier is used, an inspection and possible mending is needed before the system can be restarted. The implementation of the two barriers are realized in the improved system.

7.7 The improved system

Figure 7.7 shows the system after the safety barriers are implemented.

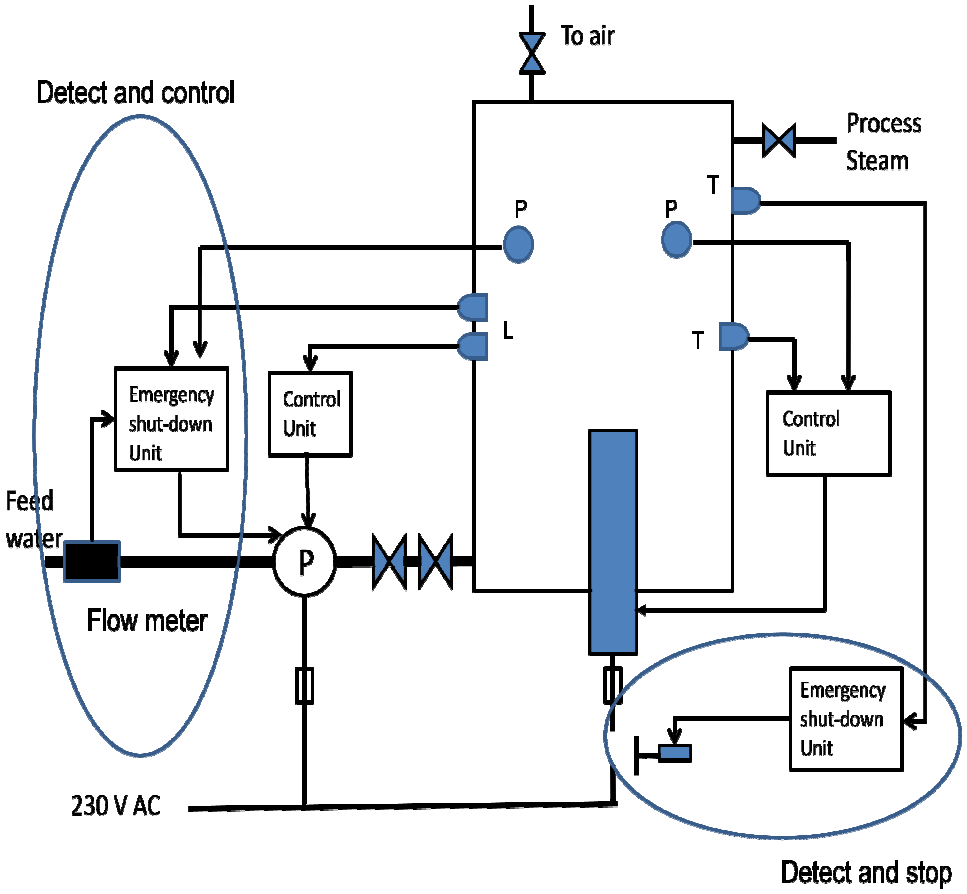


Figure 7.7: The improved system [3]

The improved system includes two safety units, which read the sensors and control the water pump and heating element accordingly. As explained in the previous section, two main safety barriers are implemented in this version of the system. These are named 'detect and control' and 'detect and stop'. The control systems are realized through control units, which should control the actuators by reading the different sensors and interpreting these values. If the control system fails to work, 'detect and stop' should be put into action. This barrier's main job is to shut down the system as a last safety measure. It is crucial that this barrier works as intended, as it is the last measure to avoid accidents.

CHAPTER 8

Experiment operation

This chapter presents the participants and groups, as well as the experiment execution.

8.1 Introduction

The participants are students at Industrial Economy studies at NTNU. They do not have courses that cover the fields of safety critical systems, safety analysis or testing. We assume that they do not have any experience with this kind of work at all. They will be split into four groups. Each group get different cases to solve. The experiment duration is one hour.

8.2 Experiment groups

The participants in this experiment will be split into four groups.

GROUPS		Test method	
		Fault injection	**Equivalence partitioning/boundary value analysis**
Additional information	**None**	G1: 7 participants	G3: 7 participants
	PHA results and barriers	G2: 8 participants	G4: 8 participants

Table 8.1: Experiment groups

Two of the groups will use fault injection, while the other two will use equivalence paritioning/boundary value analysis. The two groups using the same test method, are split into two subgroups. Only one of the subgroups using fault injection will use barriers and safety analysis results in their work. The same goes for the two groups using BVA/equivalence partitioning.

8.3 System documentation available for the groups

Each group will be given some information about the system. Table 8.2 shows which information is available to which group.

	G1	G2	G3	G4
System design	X		X	
System design with barriers		X		X
Safety requirements	X	X	X	X
Textual use cases	X	X	X	X
PHA/barriers		X		X
Fault injection task	X	X		
BVA task			X	X

Table 8.2: System documentation available for each group

We can see that the only difference is that two of the groups get to use PHA results and barriers, and in addition the system design diagram shows how the barriers are implemented. All groups are also provided with an example test case to help in formatting their test cases properly. The four different cases can be seen in the appendices.

8.4 Experiment execution

The experiment is performed over one hour. The participant's main task is to write test cases for the steam boiler system. The one hour experiment is split into these subtasks:

- The cases are handed out
- **5-10 minutes:** Steam boiler safety system presentation on PowerPoint
- **10-15 minutes:** Read and understand the steam boiler system
- Test case templates are handed out
- **35-45 minutes:** Write test cases

The output from this experiment is the test cases written by the participants. The test cases are assessed in the next chapter.

CHAPTER 9

Assessment of the deliveries

This chapter presents the different stages of assessment done to evaluate the deliveries from the experiment participants.

9.1 Introduction

The assessment of the deliveries was done over numerous stages by the author and supervisor of this thesis. A reference set of test cases was used as help to decide if a test case was valid or not.

9.2 Test cases covering the system

Given the system documentation, we have made a set of test cases that provide good coverage of the system. The test cases are written with regard to the set of safety requirements together with the system design documents. The listed test cases will form a basis for giving scores to the experiment participants answers.

#	NAME	PRECONDITION	PREVENTIVE ACTION	REFERENCE
01	*Water flow*	WF>MAX_WF	Stop feeding pump	SR4, TUC4
02	*Water level*	WL>MAX_WL	Stop feeding pump	SR1, TUC2
03	*Internal temperature*	INT_TEMP>MAX_INT_TEMP	Stop heating element	FIGURE
04	*External temperature*	EX_TEMP>MAX_EX_TEMP	Cut power heating el.	SR3, TUC1
05	*Steam pressure_max1*	P>MAX_P	Stop feeding pump	SR2, TUC3
06	*Steam pressure_max2*	P>MAX_P	Stop heating element	FIGURE
07	*Steam pressure_crit*	P>CRIT_P	Open safety valve	SR5
08	*Return valves*	P>MAX_P	Release pressure	PHA
09	*Leakage*	Unintentional leaks	Inspection, QA, c. tray	PHA
10	*Short circuit*	Damage on electrical comp.	Usage of fuses	PHA
11	*Pipe breakage*	Pipe damage	Flow meter, c. tray	PHA

MAX_WF=*max water flow*, **WL**=*water level*, **MAX_WL**=*max water level*,

INT_TEMP=*internal temperature*, **MAX_INT_TEMP**=*max internal temperature*, **EX_TEMP**=*external temperature*,

MAX_EX_TEMP=*max external temperature*, **P**=*pressure*, **MAX_P**=*max pressure*, **CRIT_P**=*critical pressure*

Table 9.1: Test cases covering the system

Test case number 1 through 7 are based on the safety requirements, system design sheets and use cases. Test case 8 through 11 are based on the safety analysis results. As a result of this, the latter will be more visible for the two groups which are provided with PHA results. Test cases 8 through 11 are therefore not critical to the same extent as the first 7. Note that test cases not provided in the reference set can also be given points.

9.3 Criterias for setting the score

In advance we agreed on the criterias for giving points.

- Every test case should get a score of either 0, 0.5 or 1.
- To achieve a score of 1, the pre-condition and pass criteria must be spot on. The textual description must describe how the testing should be done.

9.4 Stage 1: Individual assessment

When the experiment was done, we had a set of test cases as output. Each of the participants had made their own set of test cases, and every delivery was given an ID. The ID is on the format <group no.>-<participant no.> We made a copy of the deliveries so that both the author and the supervisor had identical sets. At first we assessed the test cases individually. This was done because it would give us a basis for discussion about the assessment afterwards. When both had given scores to every delivery, we sat down to find deviations and discuss the assessment.

9.5 Stage 2: Comparing assessments

When the first assessment was done, a meeting was held to compare the assessments. We were prepared to find deviations, since a factor of subjectiveness is present when giving a score to a test case. We found that we had a lot of deviations. Fortunately, almost all of the deviations stemmed from three main causes:

1. Multiple test cases from reference set represented in one test case.
2. Different assessments of the quality of the textual description.
3. Test case 8 from table 9.1 is not directly connected to the control system.

After a discussion of these three causes, we added three more criterias to the assessment.

- Some test cases include multiple test cases from the reference set of test cases. They can be given a score over the maximum of 1.
- Test cases that are correct, but lack a good textual description are given a score of 0.5.
- Test case 8 from the reference set in table 9.1 is given a score of maximum 0.5. The safety valve is not included in the control system, but is a mechanical valve that controls flow direction. Since the task given could be somewhat diffuse regarding this matter, half a point is given.

9.6 Stage 3: Individual assessment

After setting these three new criterias, we assessed every delivery individually a second time. We assumed that more compliance would be achieved when more ground rules were set for giving points.

9.7 Stage 4: Comparing assessments

After setting the scores individually, we met again to discuss the results. This time we found more compliance for the groups using fault injection. After some minor adjustments, we came to an agreement on every delivery from the two groups using fault injection as their test method. There were still large differences in the assessment of the groups using boundary testing. We quickly saw that the main issue was the textual description, and that we had a difference in opinion regarding how much description that was needed to get a full score. This problem boiled down to a core question for the experiment. How much should be demanded from the experiment participants? We understood that we needed to give this some thought until we had reached a decision that justified both the participants and our opinions.
After discussing this matter further, we decided that the difference between the barriers, namely 'detect and control' and 'detect and stop' needed to be kept intact. Mainly because this was part of one of the research questions, and therefore an important part of the experiment. If we were to give a full score on test cases that did not emphasize this fact, we would remove some of the differences between the groups that got information about barriers and those who did not get this information. Participants that understood this difference needed to be awarded for this to keep the experiment results valid.
Also, there were some differences in opinion regarding how detailed the textual description should be. We decided that test cases for the control system needed a valid pre-condition and system reaction, combined with a textual description to get a full score. The textual description had to describe how the system should be monitored during testing. For 'detect and stop' testing, it should be mentioned that the control system needs to be stopped before testing the second barrier. To sum up, these groundrules for setting the score was added:

- The participants need to understand the difference between the implemented barriers, where this is needed.
- The textual description needs to describe how the system should be monitored during testing.
- For 'detect and stop' testing, the control system should be stopped before testing can start. This needs to be mentioned to achieve a full score for the test case.

9.8 Final groundrules for assessment

After four assessment stages, we agreed on a set of common groundrules for giving out scores. Before the final comparison of assessments, these groundrules were summarized in a table. The groundrules are displayed in table 9.2.

Final groundrules for assessment
Every test case can get a maximum of 1 point
To achieve top score, the pre-condition and pass criteria must be spot on. The textual description must be good
The textual description must describe how the testing should be done
Test cases including multiple correct test cases from the reference set should get one individual score per reference test case
Test cases including multiple correct test cases from the reference set can get more than the maximum score of 1 point
Test cases that are correct but lack a good textual description should get 0.5 points
Test case 8 from the reference set is given a maximum of 0.5 points.
Test cases regarding the barrier 'detect and stop', needs to include that the control system must be stopped before testing to achieve the full score
The textual description needs to describe how the system should be monitored during testing
Test cases regarding the barrier 'detect and stop', needs to include that the control system must be stopped before testing to achieve the full score

Table 9.2: Final groundrules for test case assessment

With these groundrules put down, we assessed the test cases individually for final adjustments once more. After that, the final comparison is done to decide on differences that are still there.

9.9 Stage 5: Final comparison of assessments

A final meeting was held to finalize the assessment of the test cases. After setting all of the previous mentioned groundrules we assumed that a lot of the differences were removed. Our assumption was correct, and only smaller adjustments and discussions were needed to decide on the final scores for all participants. The larger part of these adjustments were misunderstandings or misreadings of the test cases. The final scores for the participants are presented in the next part.

Part III

Results and Discussion

CHAPTER 10

Results

This chapter provides experiment results.

10.1 Introduction

This chapter presents both individual and group results from the steam boiler experiment.

10.2 Individual results

The experiment participants were split into four groups, as shown in table 8.1. Each participant was given an ID that consisted of two numbers, the first number denotes the group ID and the second number is an individual ID to separate each delivery. The scores for all participants are provided in table 10.1.

ID	NUMBER OF CASES	SCORE
1-1	5	4.0
1-2	6	5.0
1-3	8	6.5
1-4	12	10.0
1-5	6	4.5
1-6	10	5.5
1-7	5	4.5
2-1	7	6.0
2-2	7	4.5
2-3	7	6.5
2-4	6	4.5
2-5	8	6.0
2-6	7	6.0
2-7	7	6.5
2-8	6	6.0
3-1	13	7.5
3-2	11	6.0
3-3	9	6.0
3-4	9	5.0
3-5	8	5.0
3-6	9	3.5
3-7	3	2.5
4-1	8	5.0
4-2	6	4.5
4-3	5	2.5
4-4	5	4.5
4-5	7	4.5
4-6	4	2.0
4-7	10	8.0
4-8	7	4.5

Table 10.1: Participant scores

This table shows the number of written test cases and the score given for each participant. Quality assurance of these scores are secured through the assessment stages described in the previous chapter. The next section provides analysis of these data.

10.3 Group results

This section provides data analysis of the experiment results. A brief discussion of the most important test data are done after each test result table. The analysis should provide a base for later discussion regarding the research questions provided earlier. The software used for data analysis is Minitab and Microsoft Excel. Testing is done by performing a two-sample t-test assuming unequal variances.

10.3.1 Fault injection

1->2

t-Test: Two-Sample Assuming Unequal Variances

	Fault injection	
	Without PHA/barriers	*With PHA/barriers*
Mean	**5,714**	**5,750**
Variance	**4,238**	**0,643**
Observations	**7**	**8**
HMD	0	
df	8	
t Stat	-0,043	
P(T<=t) one-tail	0,483	
t Critical one-tail	1,860	
P(T<=t) two-tail	**0,967**	
t Critical two-tail	2,306	

Table 10.2: Fault injection t-test

The t-test is performed on group 1 and 2, fault injection with and without the use of PHA results and barriers. The test shows that the mean score for these groups are basically the same. The group with information about barriers and PHA results has a slightly higher mean. The large p-value is a result of the high variance for group 1. Since the p-value is large, the results are not useful for any discussion, since the results most likely are random. A larger number of observations can be used to get useful results.

10.3.2 Boundary value analysis

3->4

t-Test: Two-Sample Assuming Unequal Variances

	Boundary value analysis	
	Without PHA/barriers	*With PHA/barriers*
Mean	**5,071**	**4,438**
Variance	**2,786**	**3,246**
Observations	**7**	**8**
HMD	0	
df	13	
t Stat	0,707	
P(T<=t) one-tail	0,246	
t Critical one-tail	1,771	
P(T<=t) two-tail	**0,492**	
t Critical two-tail	2,160	

Table 10.3: Boundary value analysis t-test

The t-test is performed on group 3 and 4, boundary value analysis with and without the use of PHA results and barriers. The test shows that the difference in mean score for these groups is 0.63. The group that did not use PHA results and barriers has the highest mean score. The variance for these groups are 2.79 and 3.25, which is high. The p-value is 0.49, which is a result of the high variance for these groups. Since the difference in mean is small and the p-value is high, the results can not be used to conclude anything. The risk of random results are very likely. Since none of the two first tests give clear differences, a combination of groups should be tested to increase the number of observations.

10.3.3 PHA/barriers

1+3->2+4

t-Test: Two-Sample Assuming Unequal Variances

	Total	
	Without PHA/barriers	*With PHA/barriers*
Mean	**5,393**	**5,094**
Variance	**3,353**	**2,274**
Observations	**14**	**16**
HMD	0	
df	25	
t Stat	0,484	
P(T<=t) one-tail	0,316	
t Critical one-tail	1,708	
P(T<=t) two-tail	**0,632**	
t Critical two-tail	2,060	

Table 10.4: Extra information t-test

This t-test is performed on group 1 and 3 combined, against group 2 and 4 combined. Since we rejected the hypothesis for group 1 versus 2, and 3 versus 4, these data sets are pooled together since they stem from the same population. The combination is done to get a higher number of observations. Group 1 and 3 did not get to use PHA results and barriers, while group 2 and 4 did. The groups that did not use PHA results and barriers got the highest mean score, 0.30 higher than the groups with this information. The number of observations are 14 and 16, but still the variance is high. The p-value is 0.63 and the risk of random results are present. This test does not give the expected results, but there are a number of variables that affect the results. These will be discussed later in this thesis, as sources of fault.

10.3.4 Without PHA/barriers

1->3

t-Test: Two-Sample Assuming Unequal Variances

	Without information	
	Fault injection	*Boundary value analysis*
Mean	**5,714**	**5,071**
Variance	**4,238**	**2,786**
Observations	**7**	**7**
HMD	0	
df	12	
t Stat	0,642	
P(T<=t) one-tail	0,267	
t Critical one-tail	1,782	
P(T<=t) two-tail	**0,533**	
t Critical two-tail	2,179	

Table 10.5: Without PHA/barriers t-test

This t-test is performed on group 1 and 3, the two groups without PHA results and barriers. They use different test methods, namely fault injection and boundary value analysis. The difference in mean score for these groups are 0.64. Both groups have a large variance. This variance gives us a high p-value, 0.53. This value tells us that the risk for a random result is high. Although the p-value is rather high, the difference in mean score gives us an indication that there might be valid differences between the two test methods used in the experiment. To get a verification of this, the two groups using PHA results and barriers should be tested, to see if there also exists a difference between the test methods when extra information is available.

10.3.5 With PHA/barriers

2->4

t-Test: Two-Sample Assuming Unequal Variances

	With information	
	Fault injection	*Boundary value analysis*
Mean	**5,750**	**4,438**
Variance	**0,643**	**3,246**
Observations	**8**	**8**
HMD	0	
df	10	
t Stat	1,883	
P(T<=t) one-tail	0,045	
t Critical one-tail	1,812	
P(T<=t) two-tail	**0,089**	
t Critical two-tail	2,228	

Table 10.6: With PHA/barriers t-test

This t-test is performed on group 2 and 4, the two groups using PHA results and barriers. Again, the groups are using different test methods, namely fault injection and boundary value analysis. The difference in mean score is 1.31. This is a higher difference than the previous test showed. This result verifies that there is a difference between the two test methods, as the previous test indicated. The variance for the groups are 0.64 and 3.25. The p-value is 0.089. The low p-value indicates that there is a low likelihood for random results. Since the two latter tests indicate a difference for the two test methods, a test with the groups combined should be done to get a higher number of observations.

10.3.6 Test method

1+2->3+4

t-Test: Two-Sample Assuming Unequal Variances

	Total	
	Fault injection	*Boundary Value Analysis*
Mean	**5,733**	**4,733**
Variance	**2,138**	**2,924**
Observations	**15**	**15**
HMD	0	
df	27	
t Stat	1,721	
P(T<=t) one-tail	0,048	
t Critical one-tail	1,703	
P(T<=t) two-tail	**0,097**	
t Critical two-tail	2,052	

Table 10.7: Test method t-test

This t-test is performed on group 1 and 2, against 3 and 4. Group 1 and 2 used fault injection, while group 3 and 4 used boundary value analysis. The data sets are pooled together since we accepted the hypothesis in one of the stand alone data sets. The difference in mean score is exactly 1 point. The variance for the two test methods are 2.14 and 2.92. The p-value is 0.097. This value tells us that there is a low chance of random results. Given the larger number of observations, this test is one of the more interesting. There is most likely a valid difference between using fault injection and boundary value analysis. The difference is present both for the groups using PHA results and barriers, and those who did not. Combined we get a stronger data set, which clearly indicates that a difference is present. A discussion of this result will be provided in the next chapter.

CHAPTER 11

Discussion

This chapter provides a discussion around the experiment results.

11.1 Introduction

This chapter will discuss the results presented in the previous chapter. The main focus will lie on answering the two research questions provided earlier.

11.2 Research question 1

- **RQ1: Which of the two test methods fault injection and equivalence partitioning/boundary value analysis is best suited for writing test cases for safety critical systems?**

  As we see from the results provided in table 10.7, fault injection scores better than boundary value analysis. This result is valid both for the groups that did not use PHA results and barriers, and those who did. The differences in mean are 0.64 and 1.31, respectively. The difference in mean when the groups are combined is exactly 1 point. One might think that 1 point is not much. One reason that it actually makes a big difference, is that the test cases are assessed in close combination with the reference set of test cases. The test cases in the reference set, especially test cases 1 through 6, are critical for the system safety and needs to be tested.
  The mean scores for fault injection and boundary value analysis respectively, are 5.73 and 4.73. Since each test case from the reference set is given 1 point maximum, it is clear that a score equal to 6 covers the most critical points of safety testing for this system. In other words, 6 points is what we should expect that the participants were able to achieve. The groups using fault injection are close to the expected score of 6 points, while the groups using boundary value analysis lack slightly more than one important test case on average. This could be the result of two things. Either they have overseen one important aspect of the system, or they have written one or more test cases lacking information. Given the nature of fault injection versus boundary value analysis, it is valid to assume that fault injection yields a better basis for writing safety tests. Fault injection has a natural coupling to safety system testing, which often is based on testing how the system reacts to a fault.
  When analyzing the test cases delivered from the boundary value analysis groups, one thing that repeats itself is the lack of a good textual description of the tests. They have difficulties in explaining how the testing should be done in practice. The groups using fault injection seem to have a better understanding of this. Using fault injection to test the steam boiler, the main procedure is to manipulate one or more sensors to provoke the safety system to perform one or more actions on the actuators. The couplings between sensors, control systems and actuators are fairly well described in the handouts, through design figures combined with textual descriptions. In the deliveries, participants using fault injection usually go straight to the point. Participants using boundary value analysis are more vague when describing the different steps of testing.

11.3 Research question 2

- **RQ2: Is the use of safety analysis results and barriers helpful in writing test cases for a safety critical system?**

  As we see from table 10.4, the participants that did not get the PHA results and barriers scored 0.30 points better on average than the participants who did. The mean score for the participants that did not use PHA results and barriers, is 5.39. The participants that used PHA results and barriers scored 5.09 points on average. As we can see from the t-test result, the p-value for this test is 0.632. This value tells us that there is a high risk that the results are a coincidence. If we only look at the groups using fault injection, the group that had PHA results and barriers available scored slightly better than those who did not, 0.04 on average. The p-value for the fault injection groups is 0.967. If we look at the same for boundary value analysis, the groups that did not use PHA results and barriers scored 0.63 points better on average. Again, the P-value is very high, 0.492.
  Even though no conclusion can be made from these results due to the high p-values, there can be made some assumptions. When the given time for the experiment was about to run out, many of the participants were still writing on their deliveries. The two groups that had PHA results and barrier information in their case description had two more pages of information than the groups that did not. Since the time span for this experiment was only one hour in total, it is possible that the groups that had more information did not have sufficient time to both understand the system and write all test cases.
  The total experiment time of one hour was divided into three parts. During the first 5-10 minutes a short presentation of the system was given on PowerPoint. The next 10-15 minutes the participants were supposed to get an understanding of the system based on the handouts. The templates for writing test cases were deliberately not handed out before the participants had read through the handouts. This was done to ensure that they actually tried to understand the system before starting to write their test cases. The last 35-45 minutes were used to write test cases. The two groups that had the two extra pages to read through in their handouts, might have used more than the given 10-15 minutes to read and understand the system.
  Given the assumptions above, it is clear that PHA results and barrier information does not automatically yield better test case writing. Extra time needs to be given to the test team, so they have sufficient time to read and understand the information. The information might even be useless if no time is earmarked for this specific task. It is important to note that this statement is based on assumptions, and that no conclusion can be made.

CHAPTER 12

Sources of fault

This chapter provides sources of fault for the experiment.

12.1 Introduction

This chapter presents sources of faults for the experiment. The sources of fault are divided into different types.

12.2 Time

As discussed in the previous chapter, time could be one of the main sources of fault. Most of the participants used all the time available, so if more time had been given one might see differences in the results. If this assumption is correct, the results that suffered most would be the testing done in connection to research question 2. Results regarding research question 1 are not affected, since there are an equal number of observations for participants with the extra two pages available.

12.3 Quality of case descriptions

The quality of the case descriptions for the four groups might be aberrant. Even though the similarity of the case descriptions were emphasized when making them, there might exist deviations that affect the test results.

12.4 Participant knowledge/experience

All participants were students in the same class. We had no knowledge of their experience or interest in the field of knowledge relevant to this experiment. We have made the assumption that these deviations are spread randomly over the groups, so that the final results still were valid. The case descriptions were handed out randomly to avoid that participants with the same interests were put in the same group. These measures help in randomizing the participants, but with a relatively small number of observations one can never be sure.

Other possible sources of fault are mentioned below. Only the sources of fault relevant for this experiment are chosen from a checklist. Some of the descriptions are reported directly from [15], where the checklist is found.

12.5 Conclusion validity

12.5.1 Low statistical power

Some of the t-tests performed on this data give low statistical power. This is discussed for each t-test, so it should be clear which tests reveal a true pattern in the data and which do not.

12.5.2 Fishing and the error rate

Fishing for a specific result is a threat, and it can be done deliberately or not. Since the experiment data is found by assessment of test cases, there is a risk of undeliberately looking for the wanted answers. Several stages of assessment and two people performing the assessment independently is done to increase the quality of the assessments. The error rate is denoted mainly by the p-value for each test. This value is emphasized when discussing the research questions.

12.5.3 Reliability of measures

Objective measures, that can be repeated with the same outcome, are more reliable than subjective measures. For example there may be poor question wording in the case descriptions.

12.5.4 Random heterogeneity of subjects

There is always heterogeneity in a study group. Since the participants are not selected from a general enough population, this reduces the external validity of the experiment.

12.6 Internal validity

12.6.1 Maturation

This is the effect of the fact that the subjects react differently as time passes. Examples are when the subjects are affected negatively (tired or bored) during the experiment, or positively (learning) during the course of the experiment. This might be a problem in this experiment, due to the different participants that were involved.

12.6.2 Selection

This is the effect of natural variation in human performance. Volunteers are generally more motivated and suited for a new task than the whole population. Since every participants were volunteers for this experiment, we have avoided this problem.

12.7 Construct validity

12.7.1 Hypothesis guessing

When people take part in an experiment they might try to figure out what the purpose and intended result of the experiment is. Then they are likely to base their behaviour on their guesses about the hypothesis, either positively of negatively, depending on their attitude to the anticipated hypothesis. This source of fault could have had an impact on this experiment.

12.7.2 Evaluation apprehension

Some people are afraid of being evaluated. It is a tendency among people to try to look better when being evaluated, which is confounded to the outcome of the experiment. Since our participants are used to being evaluated on a regular basis, this problem should be avoided in this experiment.

12.8 External validity

12.8.1 Interaction of selection and treatment

This is an effect of having a subject population, not representative of the population we want to generalize to. Since we used participants with no known experience, the results may be different than if we used participants that do this as their job on a daily basis. Nevertheless, the group differences should be valid because we used participants with the same academic background.

Part IV

Conclusions

CHAPTER 13

Conclusions and further work

This chapter provides conclusions made from this master thesis. It also suggests further work.

13.1 Conclusions

This section provides conclusions from the experiments, separated for test method and extra information.

13.1.1 Testing technique

As the experiment results suggest, we see a difference in the use of fault injection and boundary value analysis when writing test cases for safety critical systems. Fault injection scored better, both for groups using the extra information and those who did not. The average difference for these two testing techniques is close to 1 point. This point represents one important test case from the reference set of test cases for the steam boiler. It is valid to assume that fault injection yields a better basis for writing safety tests. Fault injection has a natural coupling to safety system testing, which often is based on testing how the system reacts to an injected fault. Test cases written for fault injection testing were less vaguely written than the boundary value analysis tests.

13.1.2 Barrier model and safety analysis results

The data material from the experiment did not give any statistically valid results for the use of extra information. The extra information did not help the participants in writing test cases for the steam boiler. As sources of fault suggest, one problem might be that participants using the extra information did not have enough time to use this information. Given the results from the experiment, extra information does not automatically give any better system coverage or test case quality. The only conclusion to draw from this is that extra information requires extra time, both to understand and make use of it. If not, the information might be useless.

13.2 Further work

To investigate the use of barrier models and safety analysis results more closely, we should run a new experiment. In this experiment, the participants with extra information need to get extra time to understand and make use of this information. In this way, we may get valid results that show that this information makes a difference. If possible, there should also be a larger number of observations.

References

[1] A. Bjørgan, "Test methods for safety critical systems," 2010.

[2] A. C. Tribble and S. P. Miller, "Software intensive systems safety analysis," *IEEE A&E SYSTEMS MAGAZINE*, 2004.

[3] L. D. Gowen, J. S. Collofello, and F. W. Caliss, "Preliminary hazards analysis for safety-critical software systems," *Computers and Communications, 1992. Conference Proceedings, Eleventh Annual International Phoenix Conference*, 1992.

[4] C. A. Ericson, *Hazard analysis techniques for system safety.* John Wiley & Sons, INC., 2005.

[5] ABB, "Steam boiler documentation,"

[6] E. Hollnagel and C. Cacciabue, "Reliability assessment of interactive systems with the system response generator. safety and reliability '92," *ESREL, London, Elsevier Applied Science.*, 1992.

[7] E. Hollnagel, "Accident analysis and barrier functions," *TRAIN - Traffic Safety and Information Environment for Train Drivers*, 1999.

[8] T. Stålhane, "Development of safety critical software," *Lecture notes, TDT68, Analyse av trygghet i IT-systemer*, 2010.

[9] C. Sandom and D. Fowler, "People and systems: Striking a safe balance between human and machine," *Developments in Risk-based Approaches to Safety, Proceedings of the 14th Safety-Critical Systems Symposium, Springer-Verlag.*, 2006.

[10] H. Tu and F. Wu, "How to design an environment simulator for safety critical software testing," *Eigth Asian Test Symposium (ATS'99)*, 1999.

[11] J. Güthoff and V. Sieh, "Combining software-implemented and simulation-based fault injection into a single fault injection method," *Fault-Tolerant Computing, 1995. FTCS-25. Digest of Papers., Twenty-Fifth International Symposium*, 1995.

[12] J. Christmansson, M. Hiller, and M. Rimèn, "An experimental comparison of fault and error injection," *Software Reliability Engineering, 1998. Proceedings. The Ninth International Symposium*, 1998.

[13] M.-C. Hsueh, T. K. Tsai, and R. K. Iyer, "Fault injection techniques and tools," *US Department of Defense Advanced Research Projects Agency*, 1997.

[14] B. Neate, "Boundary value analysis," *University of Wales Swansea*.

[15] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslen, *Experimentation in software engineering*. Kluwer Academic Publishers, 2000.

Appendices

APPENDIX A

Case 1

CASE 1

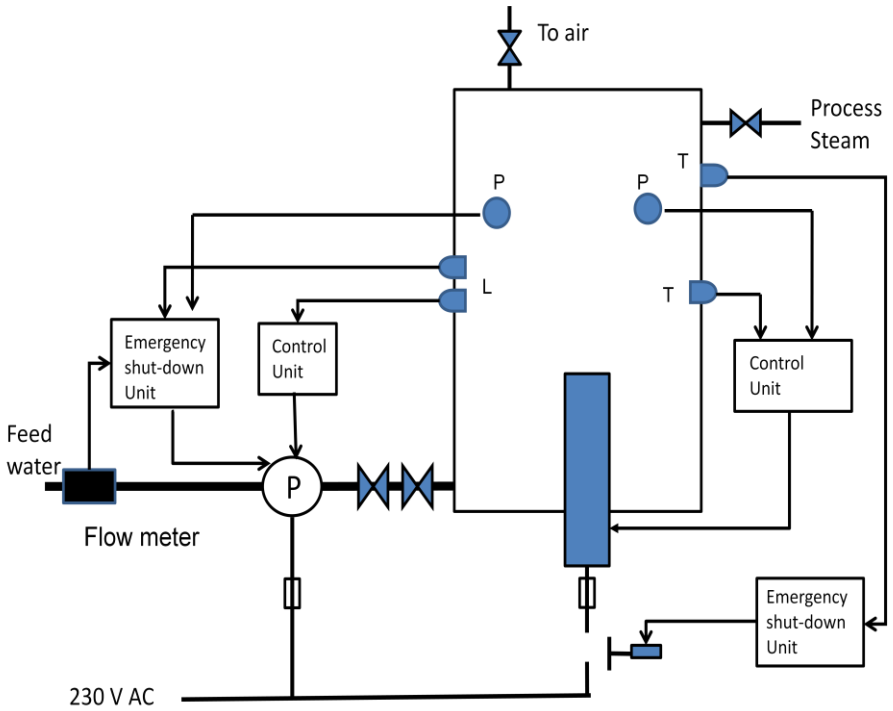
Introduction

The system under test is a control system for a steam boiler. The following documentation is handed out:

- Steam boiler with safety control units
- Safety requirements
- Textual use cases

Task 1: Get an understanding of the system (15-20 min)

Steam boiler with safety control units



The water pump, denoted by P, pumps water into the boiler. The heating element heats the water inside the tank, and steam is produced. There are several sensors which send signals to the control units.

Sensor L: Gives a signal if the water is above this level
Sensor P: Reads the pressure inside the tank
Sensor T: Reads the temperature inside/outside the tank (you can name them T_inside and T_outside when writing test cases)

There are two control units which use these sensors, and execute actions according to the values they receive. The control units control two actuators; the water pump and the heat element. Also, there are two valves, one which delivers process steam, and one that sends excess steam into the air. It is possible to shut down power to the heating element.

Safety requirements

1. **If** water level **greater than** max water level **then the** safety system **shall** stop feeding pump
2. **If** steam pressure **greater than** max pressure level **then the** safety system **shall** stop feeding pump
3. **If** external temperature **greater than** max external temperature **then the** safety system **shall** cut power heating element
4. **If** water flow **greater than** max water flow **then the** safety system **shall** stop feeding pump
5. **If** steam pressure **greater than** critical pressure level **then the** steam boiler **shall** open safety valve

Textual use cases

Use Case Name	Drum temperature control
Actor	Main control system
Pre-condition	Safety control system activated
Post-condition	Drum temperature is below limit or power to heater has been disconnected and alarm has been set
Trigger	Time triggered – run every 10 seconds
Read outside drum-temperature sensor T_outside	
Is T_outside > T_outside-limit?	
Yes => send power-off signal to actuator (Contactor) set “Too high temperature” alarm	
End use case	

Use Case Name	Water level control
Actor	Main control system
Pre-condition	Safety control system activated
Post-condition	Water level too high
Trigger	Water level sensor triggered
Is water level sensor triggered?	
Yes => send power-off signal to actuator (Water pump) Set “Too high water level” alarm	
End use case	

Use Case Name	Pressure control
Actor	Main control system
Pre-condition	Safety control system activated
Post-condition	Pressure too high
Trigger	Time triggered – run every 10 seconds
Read inside pressure sensor P	
Is P>P-limit?	
Yes => send power-off signal to actuator (Water pump) Set “Too high pressure” alarm	
End use case	

Use Case Name	Water flow control
Actor	Main control system
Pre-condition	Safety control system activated
Post-condition	Water flow failure
Trigger	Time triggered – run every 10 seconds
Read flow meter FM	
Is FM>FM-max?	
Yes => send power-off signal to actuator (Water pump) Set “Water flow failure” alarm	
End use case	

Task 2: Write test cases for the system (40-45 min)

Your task is to write test cases for the steam boiler. You should use the test method named "Fault Injection".

Fault injection is a test method that is widely used in the area of testing safety critical software. As the name says, fault injection is testing based on injecting faults into the system to see how it reacts. There are several approaches using this method, the two most common is to manipulate the software or the hardware to simulate that a hazard is occurring. One way of doing this is to manipulate sensors with faulty values, to see if the system can recover by executing commands to the actuators (water pump, heating element etc.).

You should only write test cases that uses fault injection techniques, and your goal is to write test cases that test all the faults in the steam boiler that you can think of. The test cases should achieve as high coverage of the system as possible, so be sure to read the documentation carefully to achieve a good understanding of all the aspects of the system.

Fill your test cases into the tables that are handed out. More tables are available if needed. An example test case is provided below. The example is taken from a cruise control system for a car, where the speed sensor is manipulated so that it reads a too high value.

Test case number	0
Name	Speed too high
Pre-condition	The speed sensor reads a too high value
Pass criteria ***(how should the system react to this?)***	1. Decrease speed so that 2. The current speed is below or equal to the set speed
Textual description	Testing is done by manipulating the speed sensor so that it reads a too high speed. The system should react by decreasing the speed so it is under or equal to the set speed.

APPENDIX B

Case 2

CASE 2

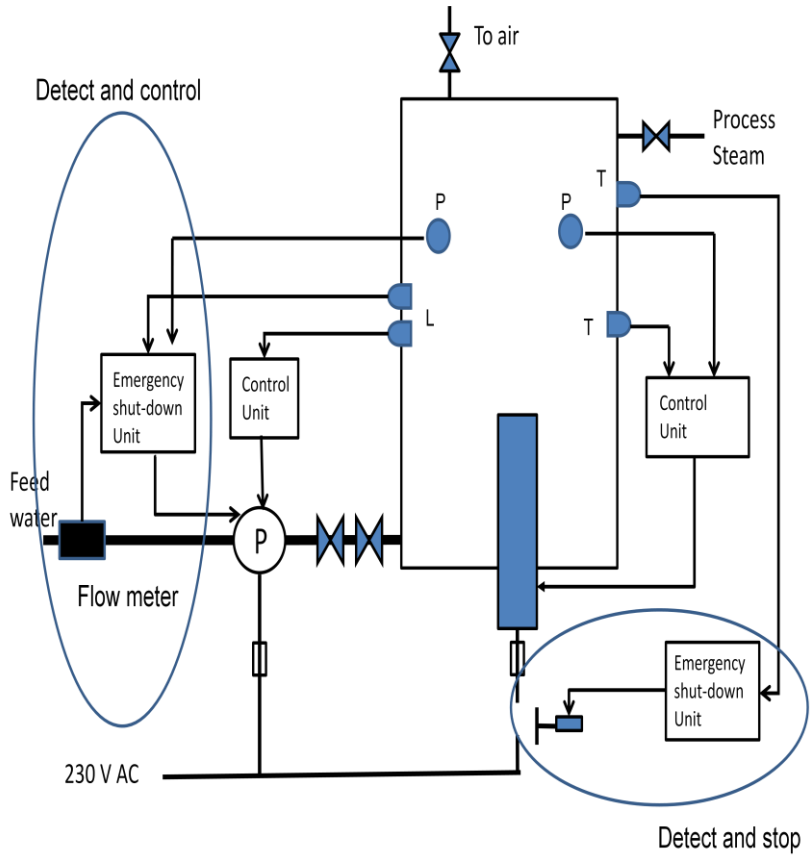
Introduction

The system under test is a control system for a steam boiler. The following documentation is available:

- The steam boiler with safety control units
- Safety requirements
- Textual use cases
- Results from safety analysis (PHA)
- The implemented safety barriers

Task 1: Get an understanding of the system (15-20 min)

Steam boiler with safety control units



Steam boiler with control system and designated safety measures

The water pump, denoted by P, pumps water into the boiler. The heating element heats the water inside the tank, and steam is produced. There are several sensors which send signals to the control units.

Sensor L: Gives a signal if the water is above this level
Sensor P: Reads the pressure inside the tank
Sensor T: Reads the temperature inside/outside the tank (you can name them T_inside and T_outside when writing test cases)

There are two control units which read these sensors, and execute actions according to the values they receive. The control units control two actuators; the water pump and the heat element. Also, there are two valves, one which delivers process steam, and one that sends excess steam into the air. It is also possible to shut down power to the heating element.

Safety requirements

1. **If** water level **greater than** max water level **then the** safety system **shall** stop feeding pump
2. **If** steam pressure **greater than** max pressure level **then the** safety system **shall** stop feeding pump
3. **If** external temperature **greater than** max external temperature **then the** safety system **shall** cut power heating element
4. **If** water flow **greater than** max water flow **then the** safety system **shall** stop feeding pump
5. **If** steam pressure **greater than** critical pressure level **then the** steam boiler **shall** open safety valve

Textual use cases

Use Case Name	Drum temperature control
Actor	Main control system
Pre-condition	Safety control system activated
Post-condition	Drum temperature is below limit or power to heater has been disconnected and alarm has been set
Trigger	Time triggered – run every 10 seconds
Read outside drum-temperature sensor T_outside	
Is T_outside>T_outside-limit?	
Yes => send power-off signal to actuator (Contactor) set “Too high temperature” alarm	
End use case	

Use Case Name	Water level control
Actor	Main control system
Pre-condition	Safety control system activated
Post-condition	Water level too high
Trigger	Water level sensor triggered
Is water level sensor triggered?	
Yes => send power-off signal to actuator (Water pump) Set “Too high water level” alarm	
End use case	

Use Case Name	Pressure control
Actor	Main control system
Pre-condition	Safety control system activated
Post-condition	Pressure too high
Trigger	Time triggered – run every 10 seconds
Read inside pressure sensor P	
Is P>P-limit?	
Yes => send power-off signal to actuator (Water pump) Set “Too high pressure” alarm	
End use case	

Use Case Name	Water flow control
Actor	Main control system
Pre-condition	Safety control system activated
Post-condition	Water flow failure
Trigger	Time triggered – run every 10 seconds
Read flow meter FM	
Is FM>FM-max?	
Yes => send power-off signal to actuator (Water pump) Set “Water flow failure” alarm	
End use case	

Results from PHA

Hazard	Cause	Main effect	Preventive action	Realization
Too high pressure in the tank	Not able to turn off the heating (sensor, control, actuator, connections)	Boiler explodes	Safety valve	Original requirement
			Turn off the heat	Detect and control
	Feeding pump failure (too strong)	Boiler rupture	Turn off power to the feeding pump	Detect and control
Too high water level	Water level regulation failure (sensor, control, actuator, connections)	Water to the process	Pump emergency stop	Detect and control
Too high pressure in the feed pipe	Non-return valve failure	Release boiling water to the water supply	Two non-return valves in series	-
			Emergency valve for releasing pressure	Original requirements
The tank is too hot	Too little water and too much heat (sensor, control, actuator, connections)	Tank gets hot/fire	Turn off the heat	Detect and stop
			Add water?	-
Unintentional leaks	Corrosion	People get scalded	Inspection, collector tray or quality assurance	-
	Bad welding/fittings	People get scalded	Inspection, collector tray or quality assurance	-
Electric shock	Short circuit	People get hurt/killed	Fuses	-
Flooding	Breakage in pipes	Damage to equipment and/or environment	Flow meter, collector tray	-

Results from preliminary safety analysis – PHA

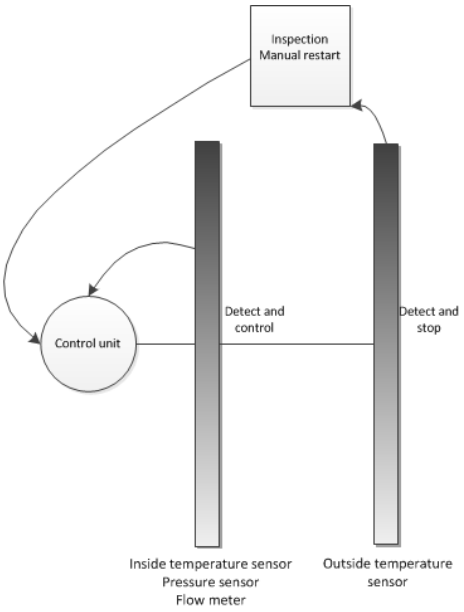
PHA(Preliminary hazard analysis) is a safety analysis method used to find and define possible hazards, together with their causes and effects. Preventive actions against the found hazards are also provided.

The implemented barriers

The preventive actions are realized through two different barriers of safety measures:

1. Detect and control.
2. Detect and stop.

The right column in the PHA results table shows which barrier is used to implement preventive actions. The barriers are explained below.



This figure shows how the two barriers are implemented into the system.

The first barrier makes use of three sensors, and the second uses one. The first barrier is used for detecting and controlling faults in the system. It should be able to put the system back into a normal state after detecting a problems/hazards.

If the first barrier fails to work, the second barrier should cut the power by using the emergency shut-down unit. This is a last measure to prevent accidents, and after this barrier is used, an inspection and possible mending is needed before the system can be restarted again.

The implementation of the two barriers is shown in the figure on the first page by circles around the components involved.

Task 2: Write test cases for the system (40-45 min)

Your task is to write test cases for the steam boiler. Use the test method "Fault Injection".

Fault injection is a test method that is widely used in the area of testing safety critical software. As the name says, fault injection is based on injecting faults into the system to see how it reacts. There are several approaches using this method, the two most common is to manipulate the software or the hardware to simulate that a hazard is occurring. One way of doing this is to manipulate sensors with faulty values, to see if the system can recover by executing commands to the actuators (water pump, heating element etc.).

You should only write test cases that uses fault injection techniques, and your goal is to write test cases that test all the faults in the steam boiler that you can think of. The test cases should achieve as high coverage of the system as possible, so be sure that you have read the documentation carefully, and achieved a good understanding of all the aspects of the system.

Fill your test cases into the tables that are handed out. More tables are available if needed. An example test case is provided below. The example is taken from a cruise control system for a car, where the speed sensor is manipulated so that it reads a too high value.

Test case number	0
Name	Speed to high
Pre-condition	The speed sensor reads a too high value
Pass criteria ***(how should the system react to this?)***	1. Decrease speed so that 2. The current speed is below or equal to the set speed
Textual description	Testing is done by manipulating the speed sensor so that it reads a too high speed. The system should react by decreasing the speed so it is under or equal to the set speed.

APPENDIX C

Case 3

CASE 3

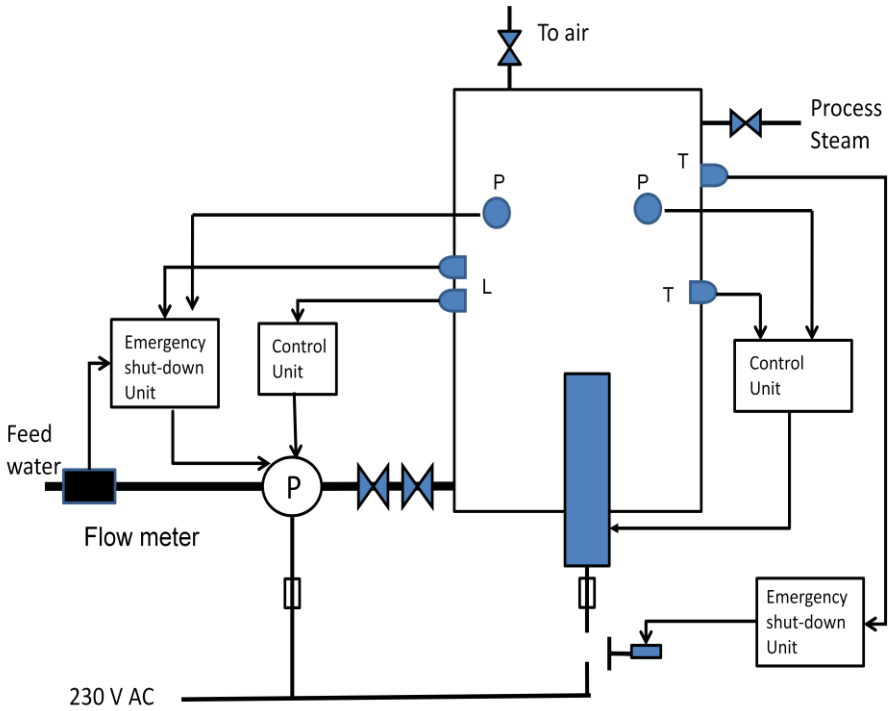
Introduction

The system under test is a control system for a steam boiler. The following documentation is available:

- Steam boiler with safety control units
- Safety requirements
- Textual use cases

Task 1: Get an understanding of the system (15-20 min)

The steam boiler with safety control units



The water pump, denoted by P, pumps water into the boiler. The heating element heats the water inside the tank, and steam is produced. There are several sensors which send signals to the control units.

Sensor L: Gives a signal if the water is above this level
Sensor P: Reads the pressure inside the tank
Sensor T: Reads the temperature inside/outside the tank(you can name them T_outside and T_inside when writing test cases)

There are two control units which uses these sensors, and execute actions according to the values they receive. The control units control two actuators; the water pump and the heat element. Also, there are two valves, one which delivers process steam, and one that sends excess steam into the air. It is also possible to shut down power to the heating element.

Safety requirements

1. **If** water level **greater than** max water level **then the** safety system **shall** stop feeding pump
2. **If** steam pressure **greater than** max pressure level **then the** safety system **shall** stop feeding pump
3. **If** external temperature **greater than** max external temperature **then the** safety system **shall** cut power heating element
4. **If** water flow **greater than** max water flow **then the** safety system **shall** stop feeding pump
5. **If** steam pressure **greater than** critical pressure level **then the** steam boiler **shall** open safety valve

Textual use cases

Use Case Name	Drum temperature control
Actor	Main control system
Pre-condition	Safety control system activated
Post-condition	Drum temperature is below limit or power to heater has been disconnected and alarm has been set
Trigger	Time triggered – run every 10 seconds
Read outside drum-temperature sensor T_outside	
Is T_outside>T_outside-limit?	
Yes => send power-off signal to actuator (Contactor) set “Too high temperature” alarm	
End use case	

Use Case Name	Water level control
Actor	Main control system
Pre-condition	Safety control system activated
Post-condition	Water level too high
Trigger	Water level sensor triggered
Is water level sensor triggered?	
Yes => send power-off signal to actuator (Water pump) Set “Too high water level” alarm	
End use case	

Use Case Name	Pressure control
Actor	Main control system
Pre-condition	Safety control system activated
Post-condition	Pressure too high
Trigger	Time triggered – run every 10 seconds
Read inside pressure sensor P	
Is P>P-limit?	
Yes => send power-off signal to actuator (Water pump) Set “Too high pressure” alarm	
End use case	

Use Case Name	Water flow control
Actor	Main control system
Pre-condition	Safety control system activated
Post-condition	Water flow failure
Trigger	Time triggered – run every 10 seconds
Read flow meter FM	
Is FM>FM-max?	
Yes => send power-off signal to actuator (Water pump) Set “Water flow failure” alarm	
End use case	

Task 2: Write test cases for the system (40-45 min)

Your task is to write test cases for the steam boiler. You should write the test cases with the use of the test method "Boundary Value Analysis/Equivalence partitioning".

The test method is based on the limits or values for a system. The most common approaches to this method is to check if a value is over/under a given value/limit, and define what the system should do if this occurs. The test case will check if the system acts correctly when a limit is crossed. One example for cruise control for a car: if current_speed>set_speed, the car should decrease speed, either by braking or letting go of the accelerator.

You should only write test cases that use the technique described above. More specifically, check if values for sensors are higher/lower/equal than a given limit, and define how the system should react if behaving correctly. Your goal is to write test cases that test all the faults in the steam boiler that you can think of. The test cases should achieve as high coverage of the system as possible, so be sure to read the documentation carefully to achieve a good understanding of all the aspects of the system.

Fill your test cases into the tables that are handed out. More tables are available if needed. An example test case is provided below.

Test case number	0
Name	Speed to high
Pre-condition	current_speed>set_speed
Pass criteria ***(how should the system react to this?)***	1. Decrease speed so that 2. current_speed<=set_speed
Textual description	If the current speed is over the set speed, the system should decrease speed until the current speed is under the set speed

APPENDIX D

Case 4

CASE 4

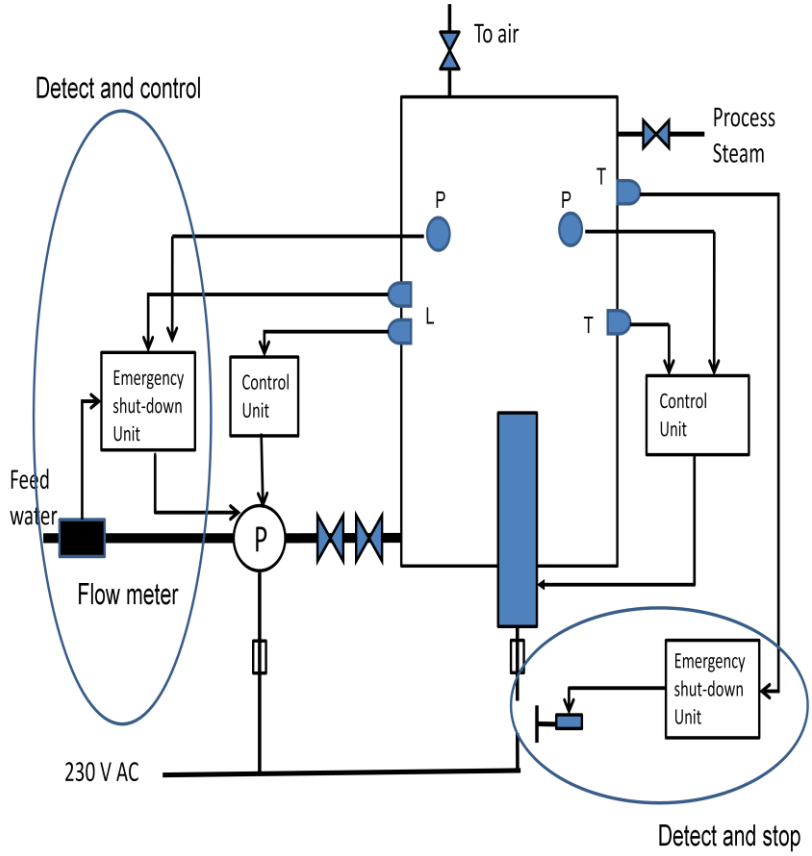
Introduction

The system under test is a control system for a steam boiler. The following documentation is available:

- The steam boiler with safety control units
- Safety requirements
- Textual use cases
- Results from safety analysis (PHA)
- The implemented safety barriers

Task 1: Get an understanding of the system (15-20 min)

Steam boiler with safety control units



Steam boiler with control system and designated safety measures

The water pump, denoted by P, pumps water into the boiler. The heating element heats the water inside the tank, and steam is produced. There are several sensors which send signals to the control units.

Sensor L: Gives a signal if the water is above this level
Sensor P: Reads the pressure inside the tank
Sensor T: Reads the temperature inside/outside the tank(you can name them T_outside and T_inside when writing test cases)

There are two control units which read these sensors, and execute actions according to the values they receive. The control units control two actuators; the water pump and the heat element. Also, there are two valves, one which delivers process steam, and one that sends excess steam into the air. It is also possible to shut down power to the heating element.

Safety requirements

1. **If** water level **greater than** max water level **then the** safety system **shall** stop feeding pump
2. **If** steam pressure **greater than** max pressure level **then the** safety system **shall** stop feeding pump
3. **If** external temperature **greater than** max external temperature **then the** safety system **shall** cut power heating element
4. **If** water flow **greater than** max water flow **then the** safety system **shall** stop feeding pump
5. **If** steam pressure **greater than** critical pressure level **then the** steam boiler **shall** open safety valve

Textual use cases

Use Case Name	Drum temperature control
Actor	Main control system
Pre-condition	Safety control system activated
Post-condition	Drum temperature is below limit or power to heater has been disconnected and alarm has been set
Trigger	Time triggered – run every 10 seconds
Read outside drum-temperature sensor T_outside	
Is T_outside>T_outside-limit?	
Yes => send power-off signal to actuator (Contactor) set “Too high temperature” alarm	
End use case	

Use Case Name	Water level control
Actor	Main control system
Pre-condition	Safety control system activated
Post-condition	Water level too high
Trigger	Water level sensor triggered
Is water level sensor triggered?	
Yes => send power-off signal to actuator (Water pump) Set “Too high water level” alarm	
End use case	

Use Case Name	Pressure control
Actor	Main control system
Pre-condition	Safety control system activated
Post-condition	Pressure too high
Trigger	Time triggered – run every 10 seconds
Read inside pressure sensor P	
Is P>P-limit?	
Yes => send power-off signal to actuator (Water pump) Set “Too high pressure” alarm	
End use case	

Use Case Name	Water flow control
Actor	Main control system
Pre-condition	Safety control system activated
Post-condition	Water flow failure
Trigger	Time triggered – run every 10 seconds
Read flow meter FM	
Is FM>FM-max?	
Yes => send power-off signal to actuator (Water pump) Set “Water flow failure” alarm	
End use case	

Results from PHA

Hazard	Cause	Main effect	Preventive action	Realization
Too high pressure in the tank	Not able to turn off the heating (sensor, control, actuator, connections)	Boiler explodes	Safety valve	Original requirement
			Turn off the heat	Detect and control
	Feeding pump failure (too strong)	Boiler rupture	Turn off power to the feeding pump	Detect and control
Too high water level	Water level regulation failure (sensor, control, actuator, connections)	Water to the process	Pump emergency stop	Detect and control
Too high pressure in the feed pipe	Non-return valve failure	Release boiling water to the water supply	Two non-return valves in series	-
			Emergency valve for releasing pressure	Original requirements
The tank is too hot	Too little water and too much heat (sensor, control, actuator, connections)	Tank gets hot/fire	Turn off the heat	Detect and stop
			Add water?	-
Unintentional leaks	Corrosion	People get scalded	Inspection, collector tray or quality assurance	-
	Bad welding/fittings	People get scalded	Inspection, collector tray or quality assurance	-
Electric shock	Short circuit	People get hurt/killed	Fuses	-
Flooding	Breakage in pipes	Damage to equipment and/or environment	Flow meter, collector tray	-

Results from preliminary safety analysis – PHA

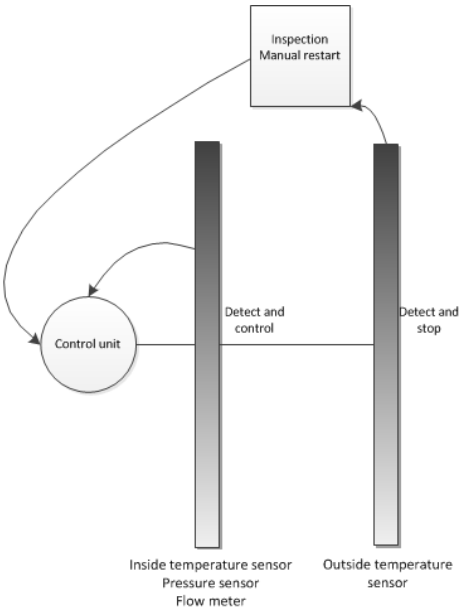
PHA(Preliminary hazard analysis) is a safety analysis method used to find and define possible hazards, together with their causes and effects. Preventive actions against the found hazards are also provided.

The implemented barriers

The preventive actions are realized through two different barriers of safety measures:

1. Detect and control.
2. Detect and stop.

The right column in the PHA results table shows which barrier is used to implement preventive actions. The barriers will be explained below.



This figure shows how the two barriers are implemented into the system.

The first barrier makes use of three sensors, and the second uses one. The first barrier is used for detecting and controlling faults in the system. It should be able to put the system back into a normal state after detecting a problems/hazards.

If the first barrier fails to work, the second barrier should cut the power by using the emergency shut-down unit. This is a last measure to prevent accidents, and after this barrier is used, an inspection and possible mending is needed before the system can be restarted again.

The implementation of the two barriers is shown in the figure on the first page by circles around the components involved.

Task 2: Write test cases for the system (40-45 min)

Your task is to write test cases for the steam boiler. You should write the test cases with the use of the test method "Boundary Value Analysis/Equivalence partitioning".

The test method is based on the limits or values for a system. The most common approaches to this method is to check if a value is over/under a given value/limit, and define what the system should do if this occurs. The test case will check if the system acts correctly when a limit is crossed. One example for cruise control for a car: if current_speed>set_speed, the car should decrease speed, either by braking or letting go of the accelerator.

You should only write test cases that use the technique described above. More specifically, check if values for sensors are higher/lower/equal than a given limit, and define how the system should react if behaving correctly. Your goal is to write test cases that test all the faults in the steam boiler that you can think of. The test cases should achieve as high coverage of the system as possible, so be sure to read the documentation carefully, and achieved a good understanding of all the aspects of the system.

Fill your test cases into the tables that are handed out. More tables are available if needed. An example test case is provided below.

Test case number	0
Name	Speed to high
Pre-condition	current_speed>set_speed
Pass criteria ***(how should the system react to this?)***	1. Decrease speed so that 2. current_speed<=set_speed
Textual description	If the current speed is over the set speed, the system should decrease speed until the current speed is under the set speed