Developing a Data Visualization Tool for Minimal Cut Sets

A Graduation Project by Jorien Kip

Supervised by Dr. M.I.A. Stoelinga
University of Twente
7-7-2017

Abstract

Big industries, such as the nuclear [1], aerospace [2] and chemical [3] engineering industries, use risk analysis to identify the system's causes of failures and the systems weaknesses to ensure that their operations are conducted safe and reliable. In order to identify, analyse and prioritize such risks, fault tree analysis can be used. There is already a wide number of techniques available to analyse fault trees but there is a lack of sophisticated presentations of the results of such fault tree analysis techniques. Consequently, results are hard to interpret and the presentations of those results are hardly intuitive.

The result of this project is a stand-alone tool that visualizes minimal cut sets in an intuitive and appealing manner. The visualization is realized with the use of common graph types, intuitive design principles and interaction techniques.

Acknowledgement

This project would not have been possible without the help of others. Therefore, I would like to thank some persons in particular. Firstly, I would like to thank Mariëlle Stoelinga, who has been my supervisor for the entire project. She has been a great support by guiding me throughout the project. Also, I received a lot of feedback for which I am grateful. Secondly, I would like to thank David Huistra, he helped me tremendously with programming the entire tool and came up with great improvements for the tool. Furthermore, I would like to thank him for the guidance he gave me throughout this project and all the feedback. Lastly, I would like to thank Job Zwiers who has been my critical observer and provided me with useful feedback.

Table of Contents

1 INTRODUCTION 7

Part I

2 STATE OF THE ART 9
2.1 Introduction Fault Trees Analysis 9
2.1.1 Fault Tree Construction 9
2.1.2 Gates and Events 9
2.1.3 Additional Gates and Events 10
2.1.4 Fault Tree Analysis 11
2.2 FTA Software 12
2.2.1 Fault Tree Analyser – ALD Software [15] 13
2.2.2 RAM Commander -ALD [16] 14
2.2.3 Isograph [17] 15
2.2.4 ConceptDraw [18] 17
2.2.5 ITEM Toolkit [19] 18
2.2.6 EPRI CAFTA [21] 20
2.2.7 BlockSim – ReliaSoft [23] 21
2.2.8 Comparison 23
2.2.9 Summary 24

Part II

3 THE ASSIGNMENT 26
3.1 Client & Assignment 26
3.2 Focus 26
4 IDEATION 27
4.1 Requirements 27
4.2 User Needs 29
4.3 Sketches 30
4.3.1 Method 30
4.3.2 Design Choices 31
4.4 Result 34
5 REALIZATION 36
5.1 Result 36
5.2 Design Choices 37
5.2.1 Adaptable to Large Amounts 38
5.2.2 Intuitive Design 39
5.2.3 Visual Appeal 40
5.2.4 Unachieved Requirements 41
5.3 Technical Implementation 41
5.3.1 Development Tool 41
5.3.2 Coding 41
5.3.3 External Code 45
5.3.4 Data format 46
6 EVALUATION 49

6.1 Online Survey ... 49
6.2 Results ... 49
6.2.1 Responses ... 49
6.2.2 Intuitive ... 50
6.2.3 Ease of Use ... 52
6.2.4 Appeal ... 54
6.2.5 Avoid Information Overload ... 54
6.2.6 Compared to Current FTA Tools ... 55
6.2.7 Feedback ... 55
7 CONCLUSION & FUTURE WORK ... 56
7.1 Conclusion ... 56
7.2 Future Work ... 56
8 APPENDICES ... 57
8.1 Appendix A: Criteria & Scores FTA Software ... 57
8.2 Appendix B: Sketches ... 60
8.3 Appendix C: Survey for Comparing Numbers ... 66
8.4 Appendix D: Coding ... 68
8.4.1 Index2.html ... 68
8.4.2 Chart 1.js ... 75
8.4.3 Chart 2.js ... 79
8.4.4 Chart 3.js ... 83
8.4.5 Circles_order.js ... 86
8.4.6 Bluehighlights.js ... 86
8.5 Appendix E: Evaluation Survey ... 88
REFERENCES ... 95

List of Figures

Figure 2.1: Fault tree example with as undesired event: Road trip stranded [6]. ... 9
Figure 2.2: Print screen Fault Tree Analyser after probability calculation. ... 13
Figure 2.3: Print screen Fault Tree Analyser displaying all the minimal cut sets ranked by their probability. ... 13
Figure 2.4: Print screen RAM Commander with available example of a fault tree. ... 14
Figure 2.5: Print Screen RAM Commander 3D bar graph. ... 15
Figure 2.6: Fault Tree Diagram Isograph. ... 16
Figure 2.7: Results Summary Isograph. ... 16
Figure 2.8: Screenshot Isograph Bar Graph with Consequence Importance ... 17
Figure 2.9: ConceptDraw Fault Tree Diagrams [18]. ... 18
Figure 2.10: Screenshot ITEM Toolkit – Fault tree example after performing analysis. ... 18
Figure 2.11: Screenshot ITEM Toolkit – Fault tree example with most critical path in red. ... 19
Figure 2.12: Bar graph different importance measures. ... 19
Figure 2.13: Print screen CAFTA. ... 20
Figure 2.14: Print Screen BlockSim. ... 21
Figure 2.15: Screen Shot of ReliaSoft BlockSim showing the Minimal Cut Sets ordered by size. ... 22
Figure 2.16: Screen Shot of ReliaSoft BlockSim Quick Calculation Pad with tooltip. ... 22
Figure 4.1: Title and description of basic events in winNUPRA manual [25]. ... 27
Figure 4.2: Bar graph showing the contribution of each minimal cut set to the top level event's probability, the events in the minimal cut set and in which MCSs a certain event appears in. ... 31
Figure 4.3: Left: all the events in a minimal cut set. Centre: the contribution of a minimal cut set to the top level event's probability. Right: an event appearing in multiple minimal cut sets. ... 32
Figure 4.4: Interface Design Explanation. ... 33
Figure 4.5: Interface design and explanation of interactivity between blocks. ... 33
Figure 4.6:Total Product Concept. ... 34
Figure 4.7: IPhone 6 Interface, Spotify Interface and Samsung Interface. ... 35
Figure 5.1: Start Screen Data Visualization Tool. ... 36
Figure 5.2: Impression of Data Visualization Tool with Selections. ... 37
Figure 5.3: Tooltips of chart 3 and chart 1. ... 37
Figure 5.4: Upper Right Menu when clicked on. ... 37
Figure 5.5: Chart 2 with different legend conditions. ... 38
Figure 5.6: Chart 3 visualizing 4, 7 and 20 minimal cut sets. ... 39
Figure 5.7: Visual Elements in Data Visualization Tool. ... 39
Figure 5.8: Data Visualization Tool without Template. ... 40
Figure 5.9: Start and End Angle calculation for Chart 3. ... 44
Figure 5.10: On the left the SB Admin template highlighting the similarities with the data visualization tool on the right. ... 45
Figure 5.11: Donut Chart without and with Drop Shadow. ... 46
Figure 5.12: Icons used from Material Icons and Font Awesome. ... 46
Figure 5.13: Excel Format of each Data File. ... 47
Figure 5.14: Format with Repetition of the Unavailability. ... 48
Figure 6.1: Visual Clues with indication of how many participants found these useful in blue. ... 50
Figure 6.2: Chart 2 with Highlighted and Less Visible Parts. ... 51
Figure 6.3: Results of the Question: How appealing is this tool, compared to current MCS visualizations? ... 55

List of Graphs

Graph 2.1: Radar Chart showing each Score for each FTA Software 23
Graph 6.1: Results of the Statement: I had the feeling that I knew what I was doing while using the tool.. 50
Graph 6.2: Results of the Statement: The tool did what I expected it to do.. 50
Graph 6.3: Results of the Statement: I had to put in a lot of effort to use this tool. 52
Graph 6.4: Completion Rate per Group. 52
Graph 6.5: Results of the Question: How easy was it to do execute the task?..................................... 53
Graph 6.6: Perceived Appeal for each Chart and the Total Tool.. 54
Graph 6.7: Results of the Statement: The chart showed too much information.................................. 54
Graph 6.8: Results of the Statement: I had the feeling that there was too much information on the page. 54

List of Tables

Table 2.1: Visual elements for each of the analysis Fault Tree Analyser provides................................ 14
Table 2.2: Visual elements for each of the analysis RAM Commander provides. 15
Table 2.3: Visual elements for each of the analysis Isograph provides.. 17
Table 2.4: Visual elements for each of the analysis ITEM Toolkit provides.. 19
Table 2.5: Visual elements for each of the analysis EPRI CAFTA provides. 20
Table 2.6: Visual elements for each of the analysis BlockSim provides. 22

1 Introduction

High-hazard industries, such as the NASA [2] and the Boeing Company [4], use risk analysis to identify the system's causes of failures and the systems weaknesses to ensure that their operations are conducted safe and reliable. Fault tree analysis is one of the most recognizable and well-known risk assessment tool [2]. FTA is a top-down approach to analyse failures and undesired events of a system with the use of a logic block diagram. In order to identify, analyse and prioritize such risks these FTA models are made by specialists. Fault trees are visualized by simple block diagrams but can get rather extensive and complicated because of the overwhelming number of blocks in the visualization which makes it inaccessible for the non-professional user [20]. After a fault tree has been constructed, analysis can be performed. There are many analyses that can be performed and these give insight into the weaknesses of the system that is being analysed. These analyses often result in tables with an excess of numbers which can be incomprehensive. Hence, this research project aims at creating more intuitive and appealing visualizations of fault tree analysis.

This report has been separated into two parts. Part I will discuss the state of the art by first introducing fault tree analysis. Secondly, fault tree analysis software will be described and how these tools visualize fault tree analysis. In Part II, the assignment of this research project will be discussed and the research question that functions as the focus point of Part II. Next, the ideation chapter explains how the first product concept of a data visualization tool has been created. The next chapter, discusses the data visualization as a result of the realization phase and how this result is accomplished. Subsequently, this data visualization tool has been evaluated which will be explained in Chapter 6. This report ends with a conclusion and a discussion on future work.

Part I

State of the Art

2 State of the Art

In this chapter an introduction of fault tree analysis will be given, how a fault tree diagram is constructed and what kind of symbols are used. Moreover, the use of fault tree analysis will be described and examples of qualitative and quantitative analyses will be described. Furthermore, in Chapter 2.2 fault tree software will be analysed and compared to give an impression of existing visualizations of fault tree analysis.

2.1 Introduction Fault Trees Analysis

In order to perform a fault tree analysis several steps need to be taken [5]. First, the system that needs to be analysed should be defined. Secondly, the fault tree needs to be constructed. Thirdly, quantitative and qualitative analyses should be performed and finally, the analyses should be interpreted. The next section describes how these steps are done.

2.1.1 Fault Tree Construction

The fault tree analysis begins with an undesired event, a failure of the system, this is the top event. Other words used for this top event are top level event and top (level) undesired event. Such an undesired event could be 'road trip stranded', see Figure 2.1. To further analyse when and how this undesired event occurs the undesired event is expanded via gates, these gates define when the higher-level event will occur. The gates refer to events that can lead to a failure of the top-level event. If there are no further specific causes for an event then these are the basic events (BEs), the leaves of the tree, meaning that the basic event cannot be specified in more underlying events.

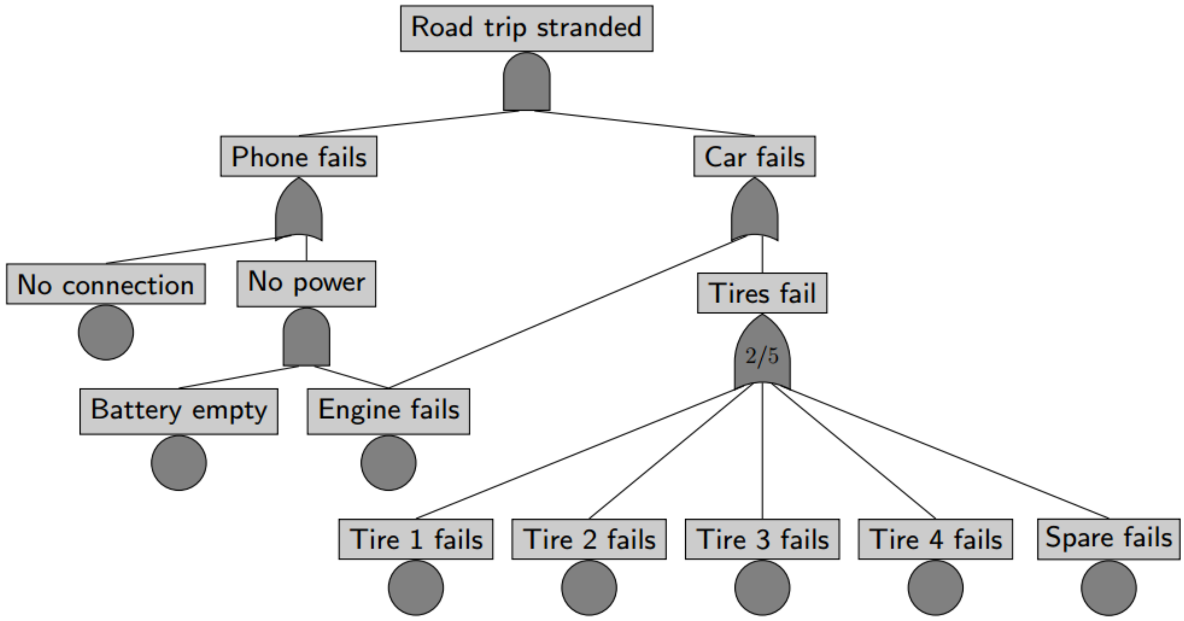
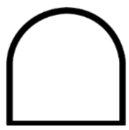


Figure 2.1: Fault tree example with as undesired event: Road trip stranded [6].

2.1.2 Gates and Events

Symbols are used for displaying gates and events in the fault tree diagram. In Figure 2.1 the fault tree uses five of the most basic symbols. These gate symbols connect events to other events.

AND-Gate



The AND-gate fails when all of its underlying events fail.
In Figure 2.1, this means that the road strip strands only if both the car and the phone fails because it is connected by an AND-gate.

OR-Gate

The OR-gate fails if at least one of the underlying events fail. In the example of the road trip, the event that the Phone fails is connected by an OR-gate. Meaning that the Phone fails occurs if at least one of the events: No connection and No power occurs.

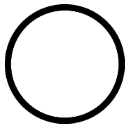
Voting OR-gate

The Voting OR-gate fails if at least *n* of its underlying events fail, also called a combination gate [2]. In the example of Figure 2.1, a Voting OR-gate describes that at least two of the five underlying events should occur so that the Tires fail occurs, meaning that at least two of the tires should fail to make this happen.

Intermediate Event

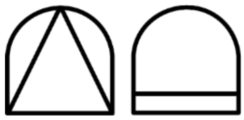
The intermediate event represents a failure of a subsystem. The intermediates events of the Road trip stranded are: Phone fails, Car fails and No power.

Basic Event



Basic events are depicted with a circle.

Exclusive OR-gate



(a) (b)

There are variations on the OR- and AND-gate. Namely, the exclusive OR-gate fails if exactly one of its underlying events fail, which is identical to a Voting OR-gate where one out of *N* underlying events must fail.

Priority AND-gate

(c) (d)

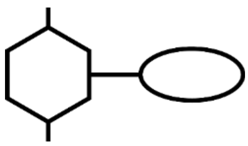
Furthermore, the priority AND-gate is a special case of the AND-gate where it fails only if the underlying events occur in a specified sequence. Where the sequence is often specified to the right of the gate in an ellipse but multiple presentations are possible.

There are different symbols used for the exclusive OR-gate and the priority AND-gate. For example, (b) And (c) are used by Concept Draw [7] while the NASA [2] uses (a) and (c) and Edraw [8] also uses (d).

2.1.3 Additional Gates and Events

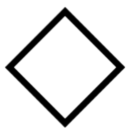
More syntactic elements are developed for more functionality of the static fault tree, they do not change the behaviour of the fault tree but are useful for modelling fault trees.

Inhibit-Gate



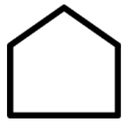
The Inhibit-gate fails only if all input events occur and an additional conditional event occurs. It is actually an AND-gate with an additional event. It is depicted as a hexagon. The inhibit-gate does not provide additional modelling capabilities but emphasizes that an additional event must also occur. The conditional event is drawn to the right of the inhibit gate and is depicted as an oval.

Undeveloped Event



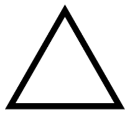
An undeveloped event is used when there is insufficient information or a lack of significance of the event.

House Event



The house event is used for an event that is normally expected to occur.

Transfer Event



Transfer events are used when a fault tree is too large to fit on a page or screen. The transfer in gate indicates that the tree is further developed on another page or screen with the corresponding transfer out gate. The tree that is connected to the transfer out gate can be attached at the corresponding transfer in gate.

2.1.4 Fault Tree Analysis

After such a fault tree has been constructed, it can be used for several purposes. One of the main uses of fault tree analysis is analysing the reliability of system and the system's components [5] [9]. With these reliabilities, it is possible to decide if reliability requirements have been met. Secondly, the fault tree can help in finding out which component of the system is more likely to be the source of trouble when the system has failed [10]. Moreover, the fault tree analysis can identify all the unique causes and commons causes for system failure [9] [11]. In the field of design, the fault tree can be used for the justification of system design changes [9] and it helps to design more fail-safe systems by for instance reducing the probability of top event failing [10]. Furthermore, the fault tree helps to understand the failure spread of the system. There are many analysing techniques available and these be divided into two techniques: qualitative and quantitative.

Qualitative Analysis Techniques

The qualitative analysis looks at the structure of the tree to get more insight of critical factors and thus weaknesses of the system. Using cut sets is one technique for doing qualitative analysis, other techniques are using minimal path sets and analysing common cause failures.

Minimal Cut Set

A cut set is a set of basic events that cause the undesired event to occur if all of the basic events of the cut set occur. If no events can be eliminated from the cut set than this is called a minimal cut set. A fault tree has a finite number of unique minimal cut sets which can be identified using Boolean algebra. The less events in the cut set, the more vulnerable the system is to that cut set. By decreasing the failure rates or probabilities of these cut sets you can improve the overall system's reliability.

Minimal Path Set

A Path set, also called success paths is a set of events that in case they do not fail it causes the system to stay operational. With improving the reliability of those minimal path sets, the overall reliability can be improved. If such a path set cannot be further reduced, this is called a minimal path set. These path sets indicate how to prevent the top event from failing. All the minimal path sets non-occurrence's are unique ways to assure that the top event will not fail.

Common Cause Failures

Common cause failures are different failures of the system but may be caused by the same (common) cause. These different system failures can occur at the same time or occur shortly after each other because of a common cause. Common cause failures are often more important because they cause more intermediate events to occur and thus giving the system a higher risk of failure.

Quantitative Analysis Techniques

Quantitative analysing techniques are focussed on the probabilities of the basic events. A probability can be assigned to each of the basic event. This probability represents the chance of the basic event to fail and is based on research and available data. The qualitative analysing techniques can be further divided into stochastic measures and importance measures.

Stochastic Measures

Stochastic measures can be calculated for each cut set and event. One stochastic measure is the reliability. This is the probability that the event does not fail. If the fault tree is continuous in time, the reliability of an event means the probability that the event will not fail for a certain period. This can also be presented as an unreliability, that being the probability of the event failing. Another stochastic measure is the availability. The availability of an event is the probability that the event (system) is working at a specified time. It can also be calculated over a time interval. Sometimes this is presented as unavailability. Another measure is the expected number of failures. These are the expected number of failures of the top event given a specific time limit. This measure is mostly used where failures of a system are very costly or dangerous. Moreover, the mean time to failure is the expected time between the moment that a system begins to operate to the first failure of the system. Then there is also mean time between failures which is the expected time between two subsequent failures of a system. For systems, which can have repairable elements, this means that the mean time between failures is the same as the mean time to failure and the mean time to repair together. This calculation is relevant for systems where failures are costly or dangerous. Another measure similar to mean time to failure is the mean time to repair.

Importance Measures

Importance measures determine the significance of all events in the fault tree as part of their contribution to the top event's probability. These importance measures can be useful during system development as they can indicate which components should get more attention through which a higher system reliability can be achieved [12]. Changing the elements with higher contributions to the top event's probability will cause a bigger change in the reliability of the system. Most of the time these measures result in a number between 0 and 1 for each system component or minimal cut set; the 1 signifies the highest level of importance [13]. Some importance measuring techniques are:

- **Minimal cut set size:** cut sets can be ordered by the number of events in the cut set. Cut sets with fewer events are generally more likely to fail since only a few events must fail simultaneously. Therefore these minimal cut sets get a higher importance.
- **Stochastic measures** can be used to order the cut sets. For each cut sets the stochastic measures can be calculated and subsequently it can be ordered.
- **Birnbaum's importance measure** measures the rate of change in the total probability of the top event as a result of change in the probability of a basic event.
- **Fussel-Vesely importance** measures the overall percentage of the contribution of events to the top event's probability.
- **Risk Reduction Worth or Top Decrease Sensitivity** measures the amount that the total probability of the top event would decrease if a basic event's failure probability would be zero, thus never fails. This can be calculated as a ratio or a difference.
- **Risk Achievement Worth or Top Increase Sensitivity** measures the amount that the total probability of the top event would increase if a basic event's probability would be 1, thus always fails. This can be calculated as a ratio or a difference.
- **Initiating and enabling importance:** separately measuring the importance of initiating events that actively cause the undesired event and enabling events that can only fail to prevent the undesired event [14]. An example of initiating importance measure is the Barlow-Proschan measure [12].

2.2 FTA Software

There are several programs available for fault tree analysis. The next section describes six programs that provide fault tree analysis and one program that enables users to draw fault tree diagrams. These programs are being described because of their availability and because these are well-known and commonly used. All of the described programs which are commercial tools have been downloaded as a trial version. First these programs will be described and per software the visual elements for each of these qualitative and quantitative analysis have been identified and categorized in tables. Next, the software will be compared based on several criteria. Lastly, a summary is given with the most significant results from the analysis of the fault tree analysis software.

2.2.1 Fault Tree Analyser – ALD Software [15]

The Fault Tree Analyser is an online tool to make fault trees and analyse them, made by a firm ADL which is specialised in, among other things, reliability engineering and safety management. After opening the Fault Tree Analyser, it opens an example of a fault tree. This online tool uses the standard symbols for gates and events. The gates get a green colour and the title AND or OR which makes it easier to recognize the gates. Repetition of colour is also used for the event symbols; the same type of event get the same colour. In this example, the events and gates are nicely arranged by the horizontal alignment but the program allows you to drag everything to any other place. But there is a tool that puts everything back to its original layout. The parent event is located above its children events and sometimes centred this emphasizes the hierarchy. The titles of the basic events are displayed inside the rectangles above the symbols of these basic events (circles) instead of putting the titles inside the circles. The failure rate per hour of a basic event is shown beneath the symbol of this event without any distinction between the failure rates.

There is an ability to zoom in and out which makes it easy to navigate and oversee the whole fault tree. By double clicking on an event or gate you can edit it. More functions of the tool are calculating the cut sets probability and overall probability, showing repeating events, listing all events and listing all gates. After analysing the probabilities, the probabilities of each intermediate event and the top event are shown in the fault tree see Figure 2.2. In Figure 2.3 the minimal cut sets are shown in a table. They are ranked by the minimal cut set's probability (highest to lowest). In the table, you can see what events are involved in the cut set and the probability of those basic events.

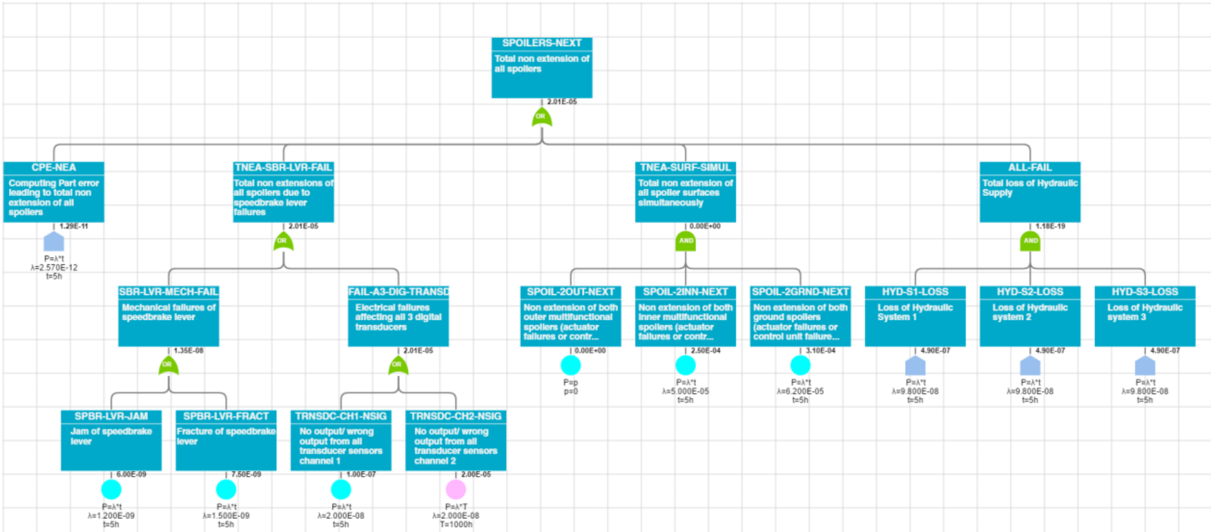


Figure 2.2: Print screen Fault Tree Analyser after probability calculation.

Minimal cut set's

Calculation : Worst case probability 'at risk time' per mission. Start time=0h, exposure=5h. Flight time=5h.
Result for Top Level : 2.01E-05. Number of MCS 7/7.Order of MCS: Min 1/ Max 3

#	CutSet prob.	Event prob.	Calc.parameters	Event Type	Event code	Event Description
1	2.00E-05					
		2.00E-05	λ=2.000E-08 T=1000h	Latent	TRNSDC-CH2-NSIG	No output/ wrong output from all transducer sensors channel 2
2	1.00E-07					
		1.00E-07	λ=2.000E-08	Evident	TRNSDC-CH1-NSIG	No output/ wrong output from all transducer sensors channel 1
3	7.50E-09					
		7.50E-09	λ=1.500E-09	Evident	SPBR-LVR-FRACT	Fracture of speedbrake lever
4	6.00E-09					

Figure 2.3: Print screen Fault Tree Analyser displaying all the minimal cut sets ranked by their probability.

		Numerical	Colour accent	Listed down in words	Placed near event	Sorted	Table
Qualitative Methods	MCS			+			+
	CCF	+	+		+		
Stochastic Measures	Unreliability	+			+		+
Importance Measures	MCSS			+			+

Table 2.1: Visual elements for each of the analysis Fault Tree Analyser provides.

2.2.2 RAM Commander -ALD [16]

ALD also developed a commercial tool: RAM Commander. An example of a fault tree can be seen in Figure 2.4. The fault tree itself is horizontally aligned and gate names are mentioned in the gate itself. Probabilities of the events are presented next to the events after analysis. Moreover, there is text underneath the basic events indicating what kind of event it is (e.g. repairable, probability) and important characteristics of that event for example the mean time to repair. In one of the available fault tree examples, they coloured the events which can help the user in finding the most interesting event. For example, they coloured the events with the highest probability in red.

There is little interaction, when hovering over an event it will shows the description of the event which you can edit yourself and the minimal cut sets can be ordered by number, probability, percentage (min. cut set prop. / total prop) and the order i.e. cut set size. Within the fault tree construction, the designer is able to insert text frames to show extra information as well as images.

The results from the analysis are given in separate windows, it is rather annoying to find the right window because to go to the fault tree again you have to minimize the current window and maximize the right window again. The results are given in tables, a 3D bar graph or a line graph, the graphs provide little interaction namely when hovering over a point it provides a tooltip with the variables on that specific point. An example of a 3D bar graph is shown in Figure 2.5.

Another interesting thing is that the program gives you hints; for example, when the minimal cut sets are calculated and these are given to the user in a table, a pop-up says: A minimal cut set is such that, if any basic event is removed from the set, the remaining events collectively are no longer a cut set.

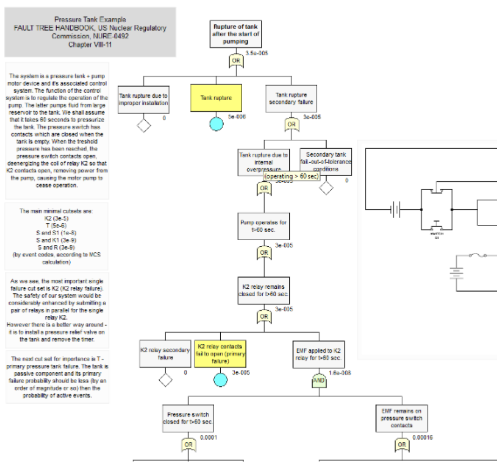


Figure 2.4: Print screen RAM Commander with available example of a fault tree.

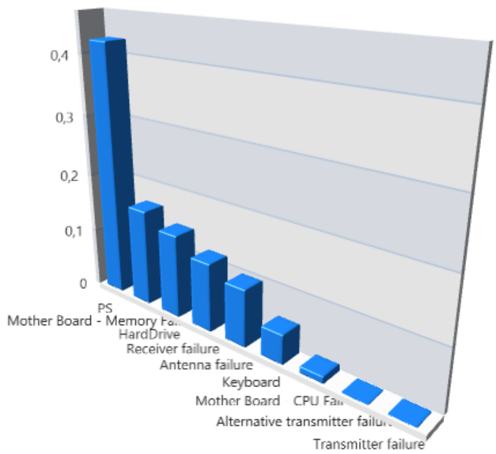


Figure 2.5: Print Screen RAM Commander 3D bar graph.

		Numerical	*Colour has meaning*	*Listed down in words*	*Placed near event*	*Sorted*	*Table*	*3D Bar Graph**	*Line Graph*
Qualitative methods	MCS			+		+	+		
	CCF	+	+		+				
Stochastic Measures	(un)Reliability	+					+		
	(un)Availability	+			+		+		+
	ENF								+
	MTTR	+			+				
Importance Measures	MCCS	+		+			+		
	Fussel-Vesely	+					+	+	
	RRW	+					+	+	
	RAW	+					+	+	
Sensitivity		+					+	+	

Table 2.2: Visual elements for each of the analysis RAM Commander provides.

** 3D Bar graph is sorted.*

2.2.3 Isograph [17]

Isograph develops software in the branch of reliability, safety and availability problems. The software is called the Reliability Workbench and allows you to draw fault trees and to analyse them. Standard symbols are used for the gates and events. The tree, see Figure 2.6, is well-arranged because of the horizontal alignment and again parent events are placed above its children but not always in the middle. The standard colour is the same for every gate and events which makes it harder to recognize different types of gates or events. However, this colour can be adjusted per icon. For more info, the user can include notes, these are displayed as tooltips when the cursor is on the gate/event. Also, anywhere on the page text labels and images can be included which can improve the usability. The titles of gates and basic events are shown in squares on top of those symbols.

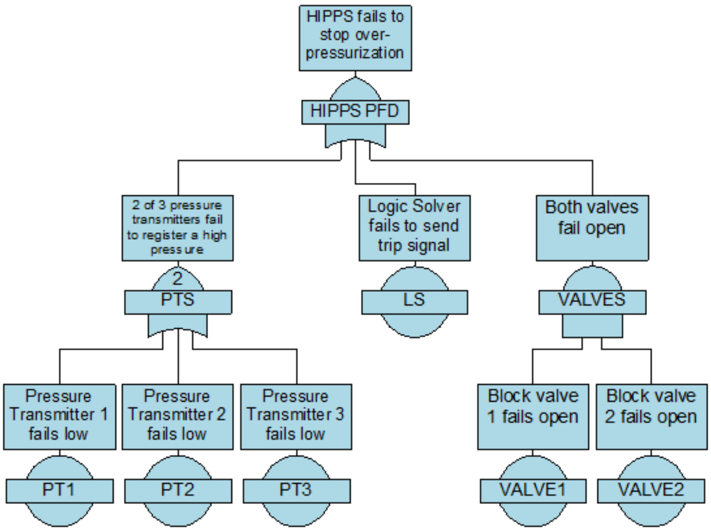
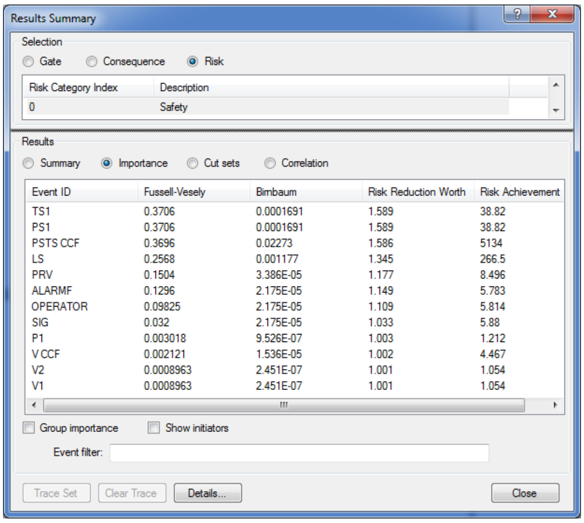


Figure 2.6: Fault Tree Diagram Isograph.

An example of the results of fault tree analysis with the use Isograph Fault Tree+ can be seen in Figure 2.7. It is a simple table with options that give the user the opportunity to switch between results that are shown in the table. At the top, the user can choose from what subject (Gate, Consequence, or Risk) the results will be shown. Underneath, the user can toggle between Summary, Importance, Cut sets or Correlation. A summary of all the available analysis is given in Table A.2 in Appendix A. This list is not complete; Isograph Fault Tree+ gives more options for analysis than the options described in these tables. Furthermore, graphs can be made such as a line graph that displays unavailability over time, or pie graphs displaying the importance of certain gates and graphs that display Probability against Unavailability. An example of a bar graph is given in Figure 2.8.



Event ID	Fussell-Vesely	Birnbaum	Risk Reduction Worth	Risk Achievement
TS1	0.3706	0.0001691	1.589	38.82
PS1	0.3706	0.0001691	1.589	38.82
PSTS CCF	0.3696	0.02273	1.586	5134
LS	0.2568	0.001177	1.345	266.5
PRV	0.1504	3.386E-05	1.177	8.496
ALARMF	0.1296	2.175E-05	1.149	5.783
OPERATOR	0.09825	2.175E-05	1.109	5.814
SIG	0.032	2.175E-05	1.033	5.88
P1	0.003018	9.526E-07	1.003	1.212
V CCF	0.002121	1.536E-05	1.002	4.467
V2	0.0008963	2.451E-07	1.001	1.054
V1	0.0008963	2.451E-07	1.001	1.054

Figure 2.7: Results Summary Isograph.

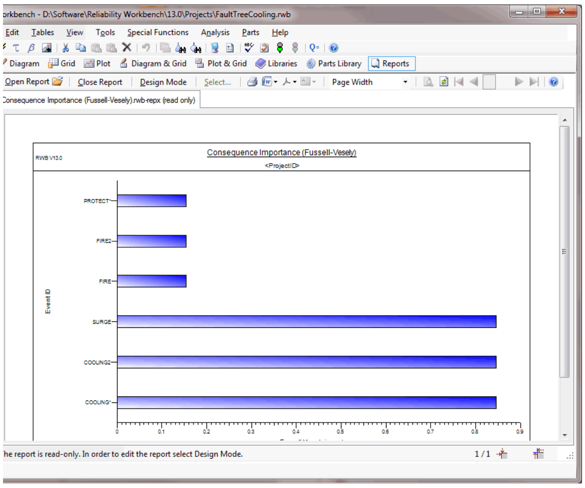


Figure 2.8: Screenshot Isograph Bar Graph with Consequence Importance

		Numerical	*Listed down in words*	*Placed on top of gate*	*Table*	*Bar*/Pie Chart*	*Other Graphs***
Qualitative methods	MCS		+		+		
Stochastic Measures	(un)Reliability	+			+		+
	(un)Availability	+		+	+		+
	ENF	+			+		
	MTTF	+			+		
	MTTR	+			+		
	MTBF	+			+		
Importance Measures	MCCS		+		+		
	Birnbaum	+			+	+	
	Fussel-Vesely	+			+	+	
	Barlow-Proschan	+			+	+	
	RRW	+			+		
	RAW	+			+		
	Sequential	+			+	+	

Table 2.3: Visual elements for each of the analysis Isograph provides.

** Bar chart is sorted, colours in graphs are for the different gates/events.*
*** Line/Curve/Bar/Area/Scatter line/Scatter curve/Scatter Graph.*

2.2.4 ConceptDraw [18]

ConceptDraw developed an extension for ConceptDraw PRO for drawing fault trees. ConceptDraw makes use of standard symbols for fault trees and the tree is nicely organised by using horizontal alignment. The parent event is placed above their children and mostly centred. The titles of events are shown in the event symbols itself; no additional rectangles are used. Moreover, different colours for gates and events helps to distinguish them. Titles for gates like AND and OR can be used on top of the gate which can help the user in understanding the gate type. The difference between the use of circles and ovals is not clear, if there is any. ConceptDraw does not support any sort of analysis on fault trees.

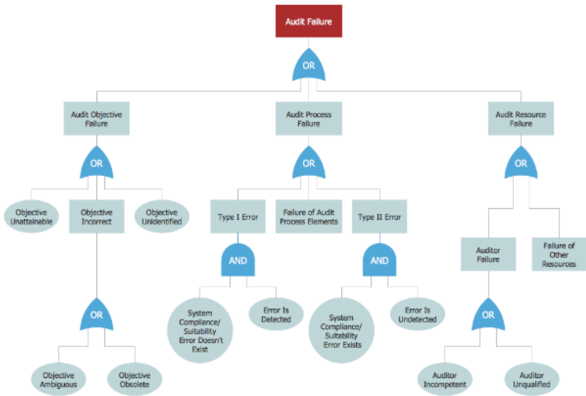


Figure 2.9: ConceptDraw Fault Tree Diagrams [18].

2.2.5 ITEM Toolkit [19]

ITEM Software call themselves the world leading developers of reliability, safety analysis and risk assessment software [20]. ITEM develops software that can be used for reliability analysis in a wide range of industries. Fault tree analysis can be done with the software ITEM Toolkit. Figure 2.10 gives an impression of a fault tree, made using this software. The fault tree visualization will not be further discussed since it is rather similar as Isograph's fault tree which is already described. What is interesting is that this software does use interaction, when hovering over the gates or events a tooltip pops up giving information on gate/event type, name and description. In case of a event it also provides information on the failure model name, type and description. After performing the analysis on this fault tree example, the tooltip expands with unavailability, failure frequency and unreliability. Moreover, there is an option that shows critical paths which is shown in Figure 2.11. The user can decide how many paths needs to be shown, their colour and line thickness and what importance measure determines the critical paths. Furthermore, this trial version allows the user to make graphs, e.g. plotting importance measures per event or unreliability over time. The user can choose what kind of graph is displayed which can cause misinterpretation of data, for instance see Figure 2.12, the bar graph shows that bad weather three importance measures which are high, however when the graph is depicted as an area under a line in a graph there seems to be only one importance measure that is that high. So, it might not be a good idea to let the user pick sort of graph.

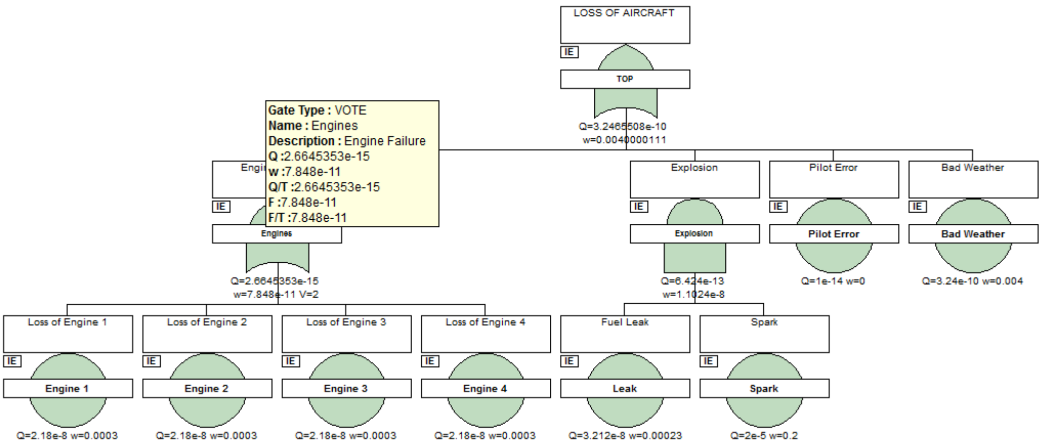


Figure 2.10: Screenshot ITEM Toolkit – Fault tree example after performing analysis.

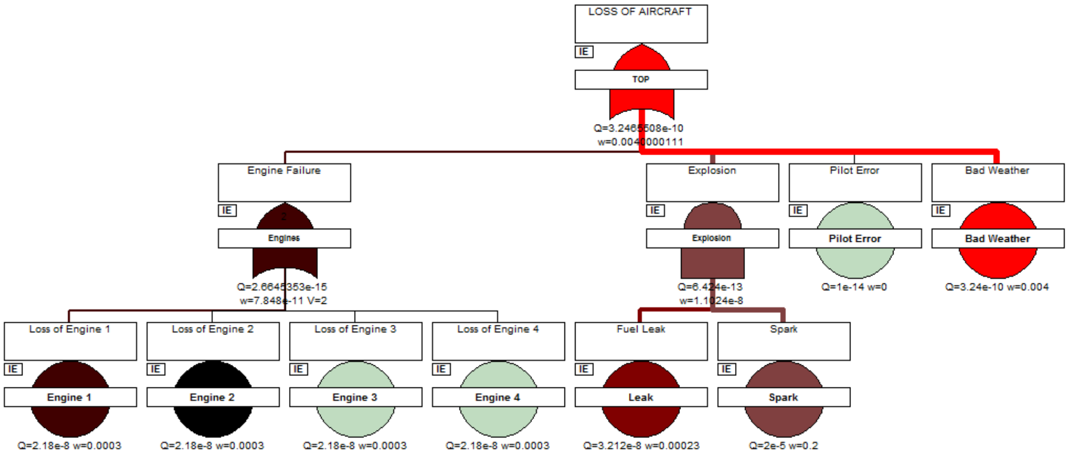


Figure 2.11: Screenshot ITEM Toolkit – Fault tree example with most critical path in red.

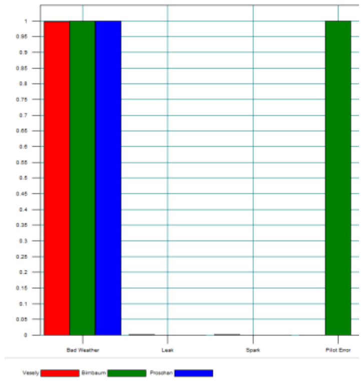


Figure 2.12: Bar graph different importance measures.

		Numerical	Listed down in words	Placed underneath	Table	Graphs*
Qualitative methods	MCS		+		+	
Stochastic Measures	(un)Reliability	+			+	+
	(un)Availability	+		+	+	+
	ENF	+			+	+
	MTTF	+			+	
	MTTR	+			+	
	MTBF	+			+	
Importance Measures	MCCS		+		+	
	Birnbaum	+			+	+
	Fussel-Vesely	+			+	+
	Barlow-Proschan	+			+	+
	RRW	+			+	

Table 2.4: Visual elements for each of the analysis ITEM Toolkit provides.

** Line plot, Horizontal bar, Vertical Bar, Pie chart, Area under a curve, 3D Manhattan graph, 3D Rooftop Graph, 2D stacked vertical Graph, 2D stacked horizontal graph, 3D isographic pie chart, Strata graph, Extended VBar, Extended HBar, Ribbon, Line only*

2.2.6 EPRI CAFTA [21]

EPRI stands for electric power research institute, this institute focuses, among other things, on safer and reliable electric power. They developed Computer Aided Fault Tree Analyser which is now used for over 30 years.

The fault tree looks rather the same as in the previous described software, see Figure 2.13. The main differences are that there is small colour use, only the intermediate events and top events have different colours so it easy to distinguish intermediate events from basic events and the appearance is very clean; every element of the fault tree has the same format. Furthermore, the probabilities are given next to the fault tree.

There are multiple interactions possible within this trial version. The user can choose to hide or expand parts of the fault tree by clicking on a little plus or minus sign at gates, which makes exploring fault trees more accessible if the fault tree is rather large. When one event is selected, the user can switch between the events by using the arrow keys, more information on these events is shown at the right of the window (name, probability, gate/event type etc.). Switching between results and the fault tree is rather easy, to switch only clicking on the different tab pages is necessary. Moreover, zooming in and out is possible.

In the trial version, it is not possible to run analysis on the fault tree example. But loading an existing file with the minimal cut sets of the fault tree example is possible. These minimal cut sets are presented in a table where there is little interaction possible. Namely, clicking on one of the variables in the table activates a slide in window on the right of the screen which shows the properties of that cut set. The user manual [22] has been used to know what analysis this software provides.

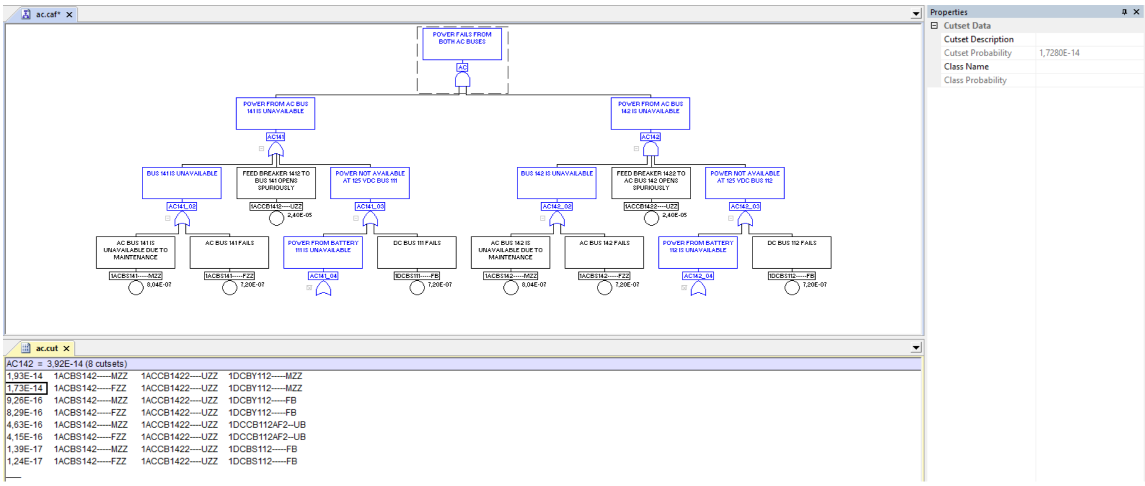


Figure 2.13: Print screen CAFTA.

		Numerical	*Listed down in words*	*Placed near event*	*Table*
Qualitative methods	MCS		+		+
	CCF	+		+	
Stochastic Measures	(un)Reliability	+		+	+

Table 2.5: Visual elements for each of the analysis EPRI CAFTA provides.

* Other results could not be evaluated since the trial version does not provide those functions.*

2.2.7 BlockSim – ReliaSoft [23]

ReliaSoft is a company that develops reliability software among which BlockSim [24]. BlockSim can both analyse repairable and non-repairable systems with the use of either reliability block diagrams. fault trees or Markov diagrams.

After entering the licence key to unlock the BlockSim trial, it is immediately evident that this program is the most appealing. The menu bar is looking very modern and comparable to the menu bar of Microsoft Word and is maybe therefore easier to use. An example of a fault tree is opened. The fault tree has a clean look; titles are represented on the gate or below the event. A big difference with the software described so far is that this tree connects gates either to basic events or gates but there are no intermediate events in this fault tree example. However, there is an option to include resultant events which function as intermediate events. The user can adjust many properties of the gates and events, among which: font, colour, shadow, border and title alignment. Adjusting properties can help the user to distinguish differences in the fault tree by giving them different colours for instance. After running an analysis, the probabilities are displayed next to the events and gates which is a bit problematic since events are colliding with the numbers. However, there is a function auto arrange which can solve this problem, but also poses another problem: the tree takes in way too much space so an overview in one look is not possible. Figure 2.14 shows the interface of BlockSim with an example of a fault tree.

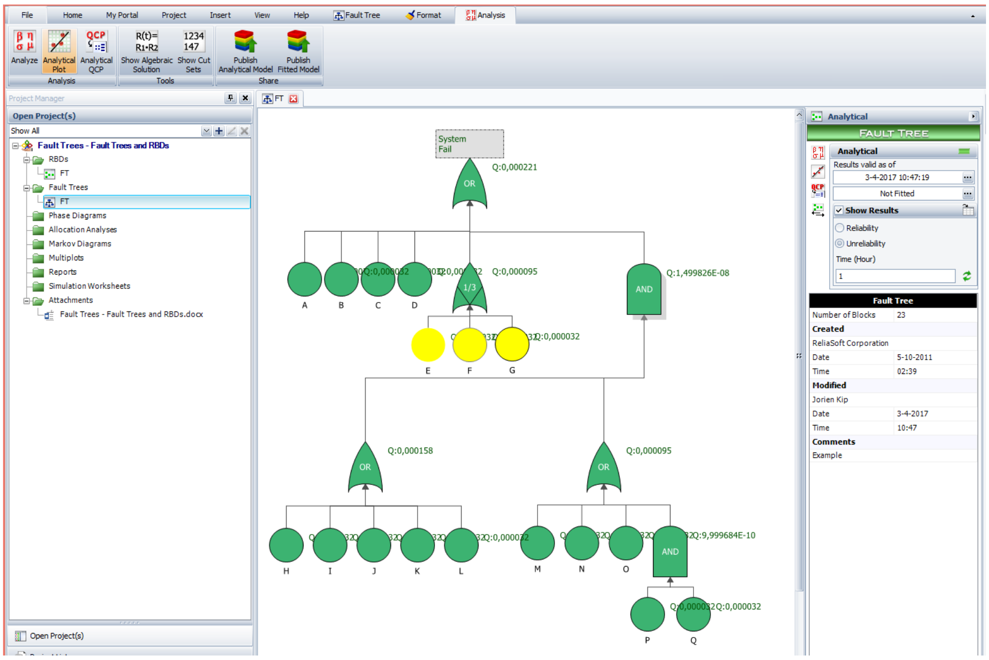


Figure 2.14: Print Screen BlockSim.

There are many interactions possible with the results of the fault tree. For example, there is interaction possible in the graph itself, hovering over the line shows you the coordinates of that specific point. Moreover, tooltips are shown for every function of the program so it gives you more insight in what that particular function does. This was especially helpful when running analyses because the program uses different names for some quantitative analyses. Another interaction is zooming in/out on the fault tree as well as on the graph. Moreover, sorting is possible when showing the minimal cut sets.

Results were given as numbers and in several graphs: bar graphs, tree maps and line graphs. Switching between results and the fault tree is easy, you don't have to run the analysis itself but you can click on tabs to switch between results and the fault tree. The minimal cut sets are shown in a separate window in a simplistic list, see Figure 2.15, and the user can choose to order these by size, reliability or unreliability. One thing that stood out is the Quick Calculation Pad shown in Figure 2.16. This window looks like a calculator and it shows all the previous calculated numbers, the user can insert variables important for the calculation for instance choosing the mission end time for calculating reliability.

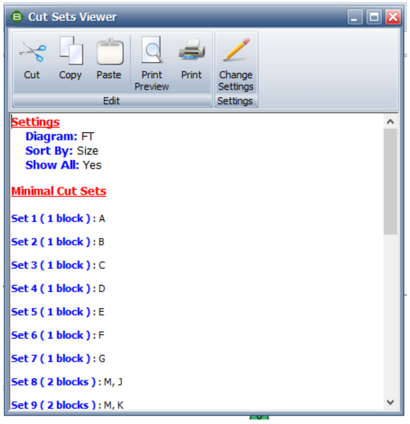


Figure 2.15: Screen Shot of ReliaSoft BlockSim showing the Minimal Cut Sets ordered by size.

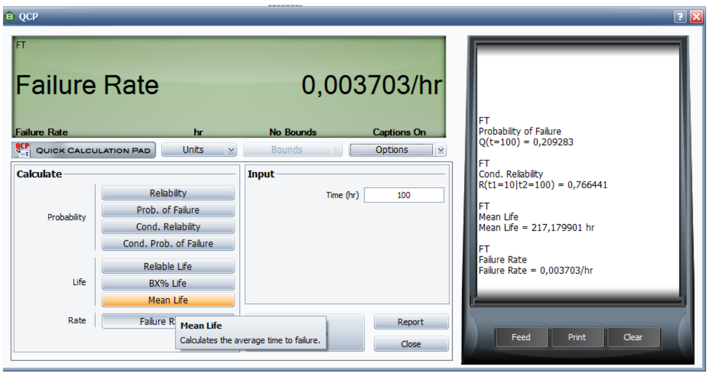


Figure 2.16: Screen Shot of ReliaSoft BlockSim Quick Calculation Pad with tooltip.

		Numerical	Colour has meaning	Listed down in words	Placed near event	Sorted	Calculator Interface	Line Graph	Bar Graph	Tree map (Tableau)
Qualitative methods	MCS			+		+				
Stochastic Measures	(un)Reliability	+			+		+	+		
	(un)Availability	+					+			
	ENF	+					+			
	MTTF	+					+			
Importance Measures	MCCS	+		+		+				
	Static Reliability		+						+	+

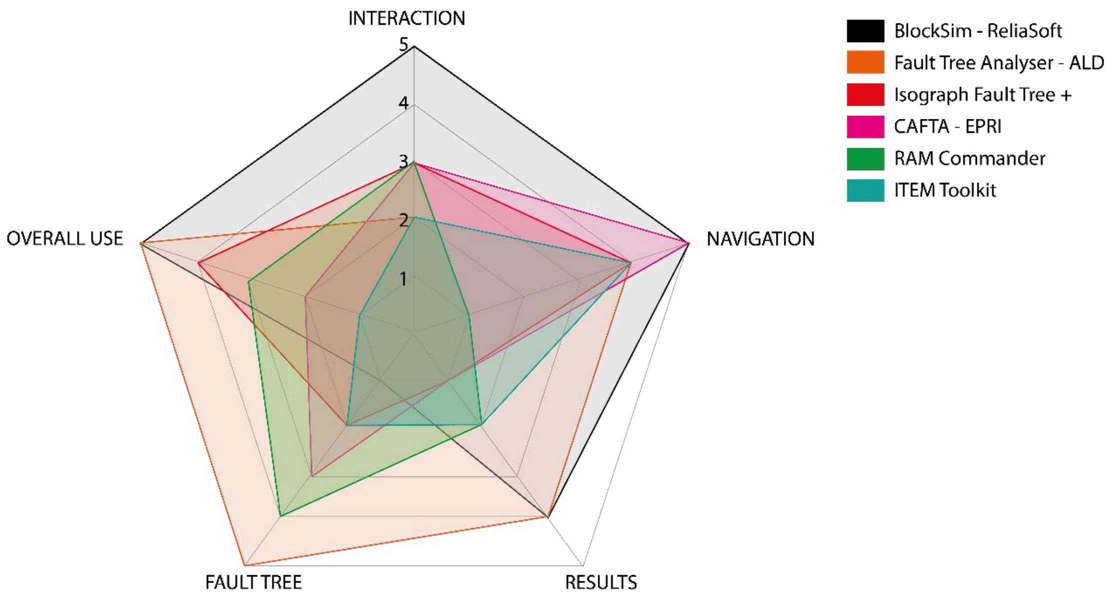
Table 2.6: Visual elements for each of the analysis BlockSim provides.

2.2.8 Comparison

The previous described software, except ConceptDraw, are compared in this section based on their interaction possibilities, navigation, presentation of results and fault tree and the overall use. Graph 2.1 shows the score for each of the software for the different criteria. In Graph A.1 in Appendix A the score for each of the software is shown separately. The criteria are:

1. Interaction: The quality and quantity of the interaction possible with the fault tree and the results.
2. Navigation: The ease of getting results and switching between result and the fault tree.
3. Results: The effectiveness and aesthetics of the representations of the results.
4. Fault tree: The effectiveness and aesthetics of the representation of the fault tree.
5. Overall use: The effectiveness and efficiency of the program.

Each program has been given a score on a scale from 1 to 5 for each of the criteria. For further explanation of the criteria see Appendix A.



Graph 2.1: Radar Chart showing each Score for each FTA Software

The best scoring software is BlockSim from ReliaSoft. This software has the most modern looking interface and therefore it was very easy to use and easy to navigate between results and fault tree. Moreover, the results are shown in either the calculator interface, a list or graphs which was better to understand than most of the other software because of tool tips and the aesthetics.

Fault Tree Analyser also has a high score. The overall use of this program was very nice because it is modern looking but also because it does not have many options in analysing the fault tree which is also a disadvantage since you can only analyse the minimal cut sets, reliability and common causes.

ITEM Toolkit has the lowest overall score. This software is difficult to use mainly because it is not clear how to get a graph visible on the screen and it is difficult to find where to click for the right analysis. Furthermore, the program looks outdated when looking at the interface as well as the results and the fault tree, so actually the whole program.

2.2.9 Summary

Most of the software were not appealing apart from ReliaSoft and Fault Tree Analyser. Also, the more modern a tool looked the more intuitive and appealing it is. Whenever graphs were an option to visualize the results, these were not appealing nor intuitive. Sometimes, the user was given the option to pick a graph type. This takes more time and the user is not always competent to choose the most fitting graph type which can cause misinterpretation of data. Often, different names for the same analysis technique were used. Consequently, it was not always clear what measures were taken. Minimal cut sets were only shown in tables or in a list. The events in the minimal cut set were only listed down in words. To calculate minimal cut sets, sometime the option was given to select a gate. But when the minimal cut sets were presented this gate was not visible to the user or it was unclear which gate it is supposed to be in the fault tree. Lastly, hints are useful. For example, when the minimal cut sets are calculated and these are given to the user in a table, a pop-up says: A minimal cut set is such that, if any basic event is removed from the set, the remaining events collectively are no longer a cut set.

Part II

The Assignment

3 The Assignment

3.1 Client & Assignment

M.I.A. Stoelinga works at the University of Twente as an associate professor in IT Risk Management and specializes in quantitative risk assessment, model-based testing and model checking. Moreover, she teaches a course at Creative Technology and supervises BSc and MSc projects. For this project, she is interested in a new way of visualizing fault tree analysis. There is already a wide number of techniques available to analyse fault trees, but there is a lack of sophisticated presentations of the results of fault tree analysis.

M.I.A. Stoelinga is interested in a framework for the visualization of fault trees analysis. She has given the assignment to make a stand-alone tool that only needs the fault tree analysis data as input which subsequently gets visualized by the tool. This data will be the result of the fault tree analysis which already took place. This stand-alone tool should be more sophisticated and interactive than existing software or tools. Other than that, it is should be appealing to the user.

3.2 Focus

The decision has been made to focus on visualizing a subset of all available fault tree analysis techniques. This decision has been made because of the time frame of this graduation project.

The focus has been set on minimal cut sets. From examining the fault tree analysis software, it can be concluded that minimal cut sets are the one fault tree analysis technique that has not been visualized other than presenting them in tables while minimal cut sets are important. Namely, they are all possible ways a system fails. Therefore, a stand-alone tool will be developed that aims to visualize minimal cut sets in a more interactive and appealing manner.

The research question of part II is as follows: How can minimal cut sets be visualized in an intuitive and more appealing manner?

4 Ideation

During the ideation phase, requirements and user needs have been developed which have been used to develop the first product sketches. All sketches have been evaluated which resulted in several design decisions and a concept idea for the data visualization tool.

4.1 Requirements

After doing an analysis of the state of the art, requirements have been made to serve as a basis for the first product ideas. The requirements are based upon the analysis of the state of the art and the interviews with the experts of NRG. This chapter describes each requirement.

Show the event's description

Minimal cut sets consist of one or more events. The analysis of the FTA programs shows that events are represented by an event code, name or ID which usually consist out of a few characters which are often a combination of capitals and numbers. Also, some FTA software have a limit on the amount of characters that can be used for this event code/name/ID [25]. This makes it easier to display this event code/name/ID in the fault tree and in the fault tree analysis because of its fixed length. This ID is hard to understand for people other than the analysts who specified these names when they analysed the system and produced a fault tree representing this system. On the other hand, the description of an event consists of words and is usually much longer. For example see Figure 4.1, it is clear that without the description, most people would not understand the event name. Therefor it is important that this event's description is shown in the data visualization, but should not always be visible due to its size.

Event Name	ST	Description	Prob.
AC-AD-KCB-PWR-CC	--	DEPENDENT FAILURE OF LINES BETWEEN AD AND KCB	3.840e-006
AC-AD-LOSP-24HRS	--	LOSS OF OFFSITE WITHIN 24 HRS OF INITATING EVENT	9.600e-005
AC-BA-BA03-BK-FO	--	BREAKER BA03 FAILS TO OPEN	5.099e-004
AC-BA-BA03-BK-TO	--	6KV CIRCUIT BREAKER BA03 FAILS TO REMAIN CLOSED	9.360e-006
AC-BA-BA07-BK-FC	--	BREAKER BA07 FAILS TO CLOSE	5.099e-004
AC-BA-BA0C-ZS-TO	--	LIMIT SWITCH BA10 CLOSED TRANSFER OPEN	4.019e-003
AC-BA-BA10-BK-FO	--	BREAKER BA10 FAILS TO OPEN	5.099e-004
AC-BA-BA7O-ZS-TO	--	LIMIT SWITCH BA07 OPEN TRANSFER OPEN	2.208e-005
AC-BA-BB07-BK-CC	--	COMMON CAUSE FAILURE TO CLOSE OF BA07 AND BB07	1.030e-005
AC-BA-BUS--BS-LF	--	AC 6KV BUS BA LOSS OF FUNCTION	3.120e-006
AC-BA-SUG0-S0-LF	--	SUG PART NEEDED FOR ALL SUG FUNCTIONS LOSS OF FUNCTION	3.420e-002
AC-BA-VOBA-VM-LF	--	VOLTAGE MEASUREMENT ON BUS BA FOR SUG LOSS OF FUNCTION	6.482e-005
AC-BAB-SUG-XS-WP	--	TEST SWITCH SCHNELL UMSCHALT GERAT LEFT IN WRONG POSITION	9.999e-003

Figure 4.1: Title and description of basic events in winNUPRA manual [25].

Show the contribution of the minimal cut set to the top event

Only a few FTA programs discussed in the state of the art show the contribution of a minimal cut set to the top event. First of all, winNUPRA shows this contribution and does this by dividing the minimal cut set's probability with the top events probability and then times 100 which results in a percentage see Equation 4.1. RAM Commander also shows this contribution while displaying the minimal cut sets. RAM Commander uses a percentage sign '%' and calculates this percentage the same way as WinNUPRA. Equation 4.1 calculates the contribution of a minimal cut set to the top event where P(MCS) is the probability of the minimal cut set and P(TLE) is the probability of the top level event. This contribution is also called the cut set importance [26].

Equation 4.1

$$\frac{P(MCS\ fails)}{P(TLE\ fails)} * 100\%$$

This contribution shows the percentage of the probability that MCS fails given that TLE fails:

Equation 4.2

$$P(MCS\ fails\ |\ TLE\ fails) = \frac{P(MCS\ fails\ \cap\ TLE\ fails)}{P(TLE\ fails)}$$

The probability that a minimal cut set fails and the top level event fails is the same as the probability that the minimal cut set fails because if a minimal cut set fails automatically the top level event fails. Thus:

Equation 4.3

$$P(MCS\ fails\ \cap\ TLE\ fails) = P(MCS\ fails)$$

Combining Equation 4.2 and 4.3 gives Equation 4.4.

Equation 4.4

$$\frac{P(MCS\ fails\ \cap\ TLE\ fails)}{P(TLE\ fails)} = \frac{P(MCS\ fails)}{P(TLE\ fails)}$$

Show all events in the minimal cut set and their probability

A minimal cut set consists out of one or more events. Each event has a probability of failure and the events of a minimal cut set together determine the minimal cut set's probability.

Show the unavailability of the minimal cut set

The unavailability of a minimal cut set is the probability that the minimal cut set will fail at a given time. This number is important to show since the higher the unavailability the higher the risk this minimal cut set creates for the entire system that is being analysed.

Show the order of the minimal cut set

The order of the minimal cut set is the amount of events that are in the minimal cut set. The lower the order the more important this minimal cut set is. Since minimal cut sets with fewer events are generally more likely to fail since only a few events must fail simultaneously.

Show for an event in how many cut sets it appears in

An event can occur in multiple cut sets. Events occurring in multiple minimal cut sets are more important since they can cause a failure of the top level event in multiple ways. Meaning that there are different combinations of events with one specific event occurring in all of those combinations.

Sort the minimal cut sets on unavailability

The default sorting of the minimal cut sets should be based on the minimal cut sets' unavailability because the higher the unavailability the higher the chance of failure of the system and thus a more important cut set. The most important minimal cut sets should be immediately visible meaning that they should be placed at the top of the table.

Small probabilities should not get invisible while visualizing these

The probability that an event or minimal cut set fails can be a number that is one thousandth (1/1000) or even one millionth (1/1000000). These numbers are normally being considered as very small but as a failure probability this can be a very high risk for the system and therefor it is important that these numbers are clearly visible in the data visualization.

The visualization should be applicable for a large amount of minimal cut sets

Systems that are being analysed by fault tree analysis can get a fault tree that is very extensive, meaning that also the amount of minimal cut sets can get very large. For instance, WinNUPRA has a limit on how many minimal cut set it solves namely 60,000 [25]. During the visit to NRG it became clear that the amount of minimal cut sets can easily run up to a list of 30,000 minimal cut sets or more. Although, it is not always desirable to look at all of the minimal cut sets because analysts are not always interested in minimal cut sets with a very low unavailability or in minimal cut sets with a high order. This limit is often determined before the FTA software is going to calculate the minimal cut sets. But the visualization should not be restricted to the amount of cut sets that can be visualized unless the user wants to.

Option to limit the amount of minimal cut sets loaded from the data

The amount of minimal cut sets can easily run up to 30,000 cut sets and it is not always desirable to analyse all of these minimal cut sets. A solution is to give the user the option to apply filters on the amount of minimal cut sets that will be loaded. The filters can limit the amount of cut sets by using a cut-off unavailability, limiting the amount itself and by specifying to which order minimal cut sets will be shown.

Give the option to select a gate for which the minimal cut set will be shown and show for which gate the minimal cut sets are calculated.

Minimal cut sets can be calculated for any gate in the fault tree, therefore it is important to show the user for which of the gates this has been done. Also, the amount of minimal cut sets can be limited by selecting a gate.

Interactivity: sorting on order and unavailability

Giving the user the option to choose and switch between the way the minimal cut sets are ordered helps the user to analyse the minimal cut sets of their interest. By giving the user this option they can analyse the minimal cut set with the highest or lowest unavailability and the minimal cut sets with the smallest or biggest order easier.

Adaptability

The data visualization should be adaptable to new minimal cut sets so that any user can visualize their minimal cut sets.

Important & interesting but omitted

Other functionalities which are important but not used for the stand-alone tool are: show the ‘failure’ path that the minimal cut set causes and show the events of the minimal cut set in the fault tree. But the intention is to make a standalone tool which uses only the data resulted from the fault tree analysis meaning that the data of the fault tree itself will not be used and therefore it would not be possible to include these functionalities.

4.2 User Needs

The target group of this graduation project consist of people that are experts in the field of fault tree analysis and people that know the basics of fault tree analysis. In order to make the data visualization tool usable for this target group, four user needs have been developed which will be discussed in the following paragraphs.

Intuitive
The data visualization tool should be intuitive; the user should be able to understand how the tool works without any special training. In other words, the tool should be self-explanatory. But how can intuitiveness be implemented into design? Everett McKay [27] has an answer to this question. He wrote a book about intuitive design and defines that a user interface is intuitive if it has an appropriate combination discoverability, affordance, predictability, responsiveness, efficiency and explorability. Only four of these concepts will be described because these are the most applicable for this data visualization tool. First of all, affordance, the user interface of the tool should provide the user with visual clues that indicate what the user needs to do. Secondly, predictability, the tool should perform as the user expected it to do. Thirdly, responsiveness, the user interface should give feedback when an action has been performed by the user. Lastly, efficiency, the user can use the tool with a minimum amount of effort.

Ease of Use
Besides that, the tool should be intuitive, it should be easy to use. These two concepts are clearly related; ease of use is about efficiency but also about effectiveness and satisfaction. The tool should be effective, meaning that the user should be able to achieve goals with accuracy and completeness. Moreover, the user should be satisfied with using the tool.

Appeal
The data visualization should have a certain amount of visual appeal. Visual appeal is what meets the eye; colours, positioning, shapes, animations. It can affect how we experience things. We associate aesthetically pleasing things often with greater value or a more pleasing experience. Also, it can attract attention from the user to the right position on the screen and thus aesthetics can act as a tool that guides the user through the data visualization tool. However, the visual elements should not distract the user from the actual goal of the data visualization.

Avoid Information Overload
The user should not get the feeling that he or she is overloaded by the amount of information shown. An information overload can confuse the user and distract from what is really important. Especially with a high number of minimal cut sets it is important to keep in mind that giving all the information of each minimal cut set at once can easily overwhelm the user.

4.3 Sketches
With the requirements in mind, many sketches have been made and are used as a starting point for the visualization tool of minimal cut sets which will be further explained in this chapter. Every single sketch has been evaluated together with M.I.A. Stoelinga. These sketches can be found in Appendix B where also can be seen which of the sketches are the most understandable and appropriate charts, these charts are presented with a green check mark. After creating and evaluating all sketches, two major design decisions have been made, as will be discussed in this chapter.

4.3.1 Method
After analysing the state of the art and setting up the requirements and user needs, sketches have been made. Generating sketches helps in generating ideas. It is a way of brainstorming by just doing and creating the idea itself. Inspiration has been found with the use of internet by looking at already existing charts, infographics and graphic designs. Sketches have been made with Adobe Illustrator because more advanced sketches can be made without a lot of effort compared to drawing. Another advantage is that if some elements are not looking good it can immediately be adapted or removed.

4.3.2 Design Choices

The first design decision is that the tool will include three charts each clearly conveying a different aspect of minimal cut sets. The second design decision is that the user interface will consist out of five blocks in order to meet the requirements while being accessible to the user. These design choices are more comprehensively described in this section.

Three Separate Charts

From the sketches, we can see that there need to be three separate visualizations in order to show all of the events in one minimal cut set and their individual contributions, the contribution of a MCS to the top level event's probability and events that occur in multiple minimal cut sets without an overflow of information.

As a counter-example, in Figure 4.2 a sketch can be seen of a chart that shows the contribution of a minimal cut set, what events are in the minimal cut set and how high their unavailability is. Moreover, when clicking on one piece of the bar, an event, it places all of these events to the bottom so the user can see in which minimal cut sets this event appears in. The problem with this sketch is that it shows too many different things that it needs more information. For instance, the percentage on top needs explanation because the percentage itself is not enough to understand that it is the contribution of the minimal cut set to the top level event's probability. Another problem is, when a bar is placed to the bottom, the percentage on top of the bar makes even less sense. Is it the percentage of the little bar that stays above the horizontal division? Or is it still the contribution of the minimal cut set to the top level event's probability?

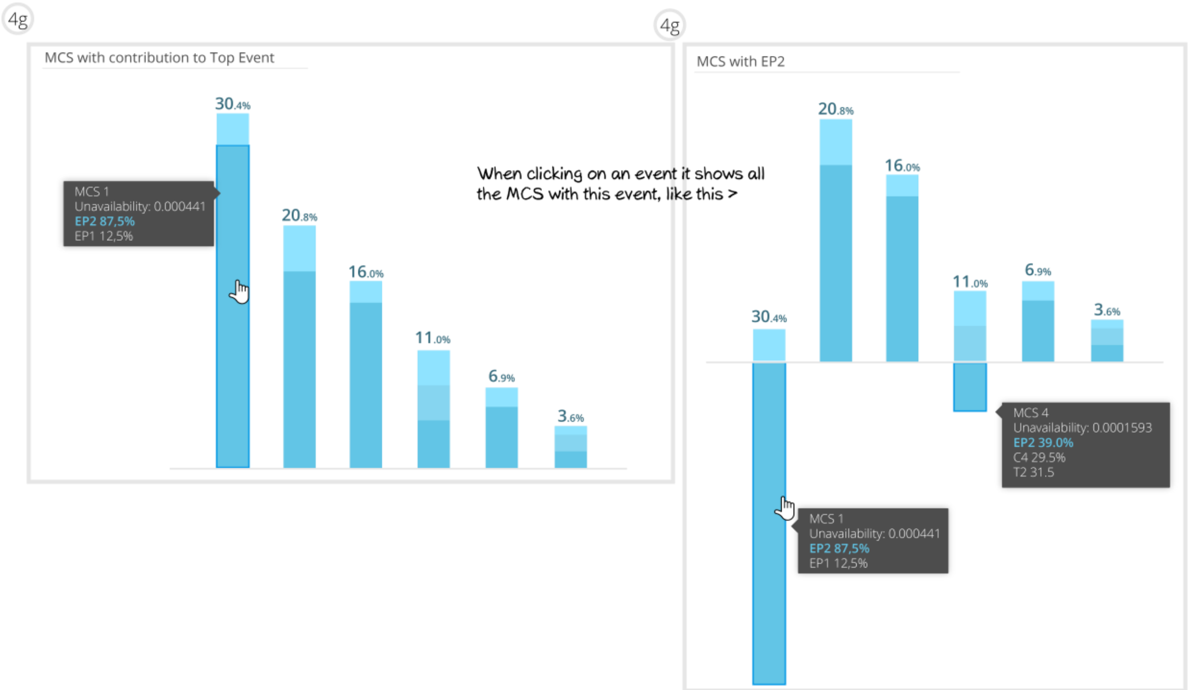


Figure 4.2: Bar graph showing the contribution of each minimal cut set to the top level event's probability, the events in the minimal cut set and in which MCSs a certain event appears in.

Therefore, the choice has been made to use three different charts. By showing these three parts separately it is easier for the user to concentrate on one visualization and thus to thoroughly examine only one part at the time.

From the validated charts, three charts have been chosen to visualize the parts mentioned before, these charts can be seen in Figure 4.3. This selection has been made because they are consistent in their shape which forms a coherent image and is less distracting than three different types of shapes. Also, consistency gives a feeling of stability. Moreover, Eels found that pie charts were just as easily, quickly and accurately to read as bar charts [28]. Also, pie charts are more efficient than bar charts when the data points increase [28].

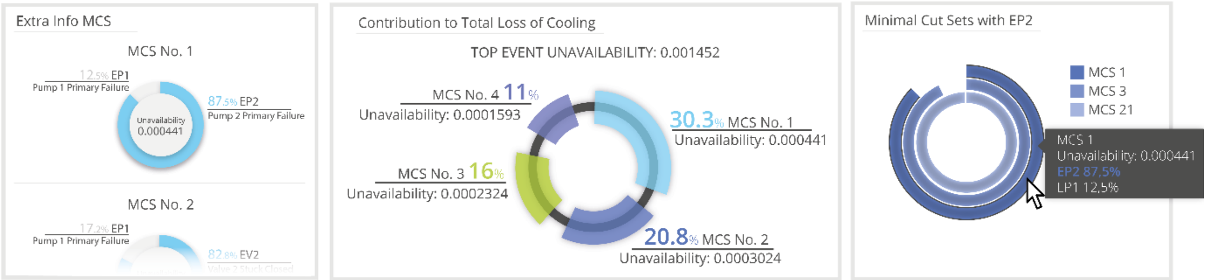


Figure 4.3: Left: all the events in a minimal cut set. Centre: the contribution of a minimal cut set to the top level event's probability. Right: an event appearing in multiple minimal cut sets.

Interface Design – *5 Blocks*

While making the sketches, it became clear that the user interface needs five different blocks to show the three charts in a way that it is still interactive and accessible to the user. The two remaining blocks are intended for a table and a filter. The table and filter are both included to decrease information overload. The table does this by controlling what is being visualized in the three charts and the filter by decreasing the amount of minimal cut sets shown in the table. Furthermore, the filter is also included to decrease the amount of processing power required by limiting the amount of minimal cut sets. In Figure 4.4, the layout of the five blocks is shown with descriptions explaining what each block should contain and visualize. The next section will describe the intended interactions between all blocks, this has also been visualized in Figure 4.5.

Interface Interaction

Multiple interactions are necessary for full functionality of the interface. One of those interactions is the option to decrease the amount of minimal cut sets shown by applying filters upon the table. The filters can be altered by clicking on the arrows, filling in an amount or order and by moving the slider in the block 'filters', which can be seen in Figure 4.4. Secondly, the user can choose for which minimal cut sets they want to see more information, this can be done by clicking on the checkboxes in the table. The checkboxes underneath 'Chart' correspond to 'Extra Info MCS' and the checkboxes underneath Chart 2 correspond to 'Contribution of MCS to TE' meaning that if one of these checkboxes is selected the minimal cut set will appear in one of these blocks. This interaction is necessary because otherwise the user will always see a pie chart of every single minimal cut set of the data which can be overwhelming. Thirdly, to get a visualization shown in block 'Minimal Cut Sets including event A' the user should select an event by clicking on an event title in the column 'Minimal Cut Set' or by clicking on an event which is displayed in block 'Extra Info MCS'. Another interaction is that the user can select the gate for which of the minimal cut sets will be shown by clicking on the arrow next to the title of the gate or top event. Lastly, the table can get sorted on contribution and order by clicking on the arrows in the header of the table.

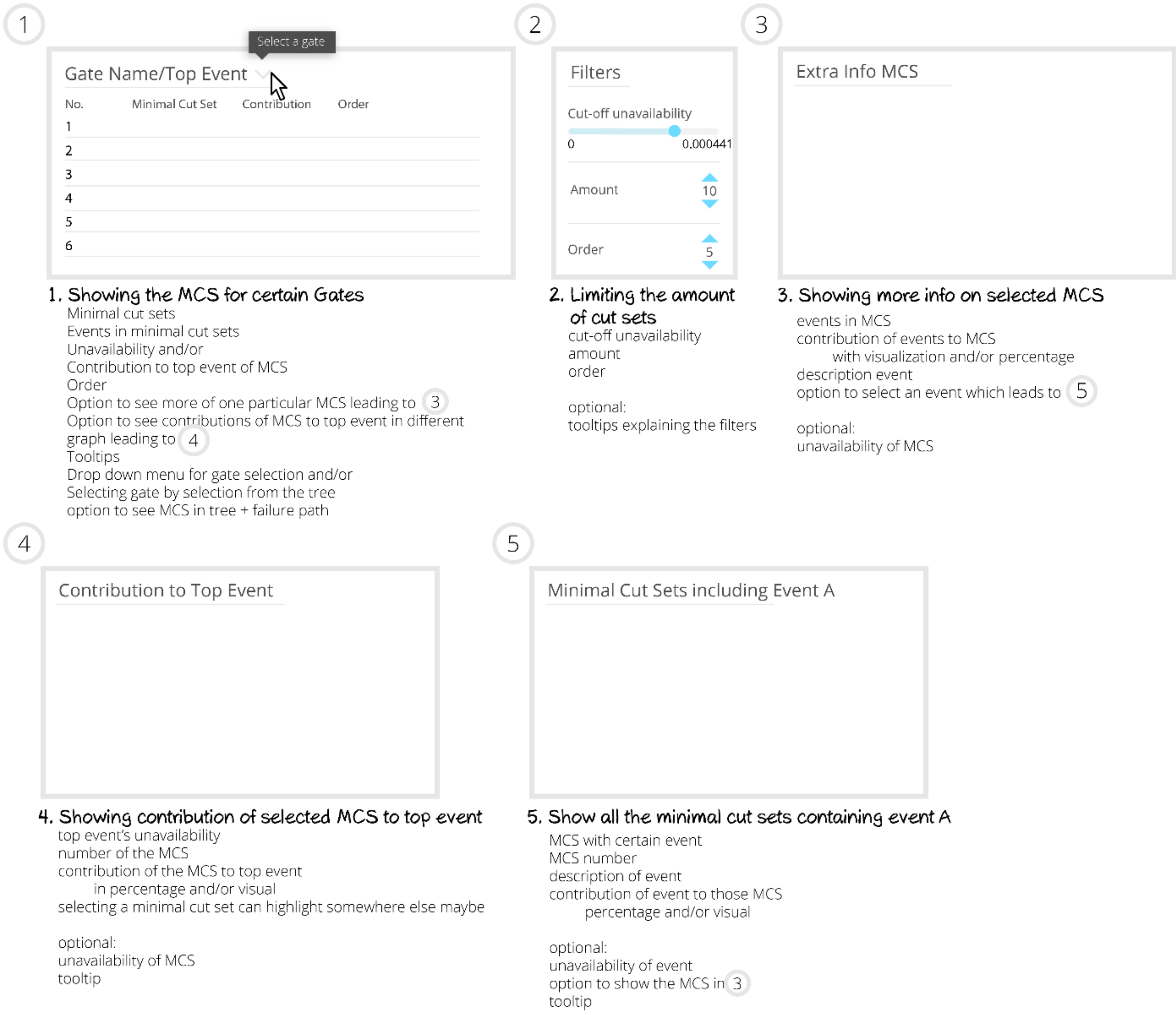


Figure 4.4: Interface Design Explanation.

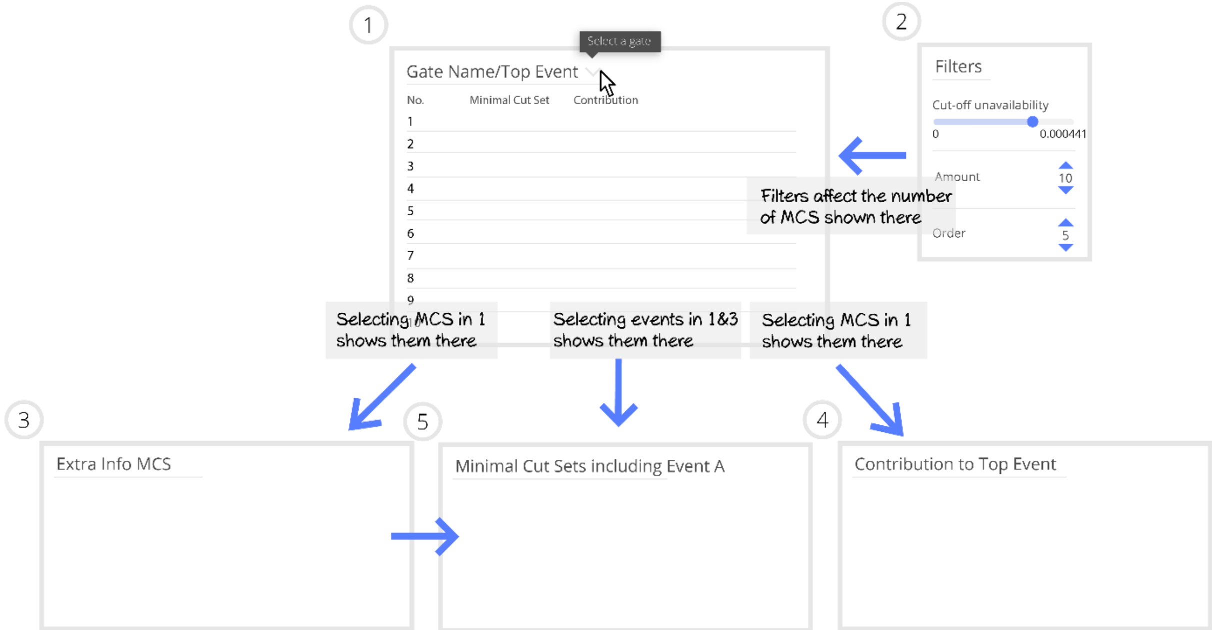


Figure 4.5: Interface design and explanation of interactivity between blocks.

4.4 Result

This chapter will explain the first product idea as result from this ideation phase. Each block of the interface will be clarified regarding the content and a complete overview of this product idea can be found in Figure 4.6.

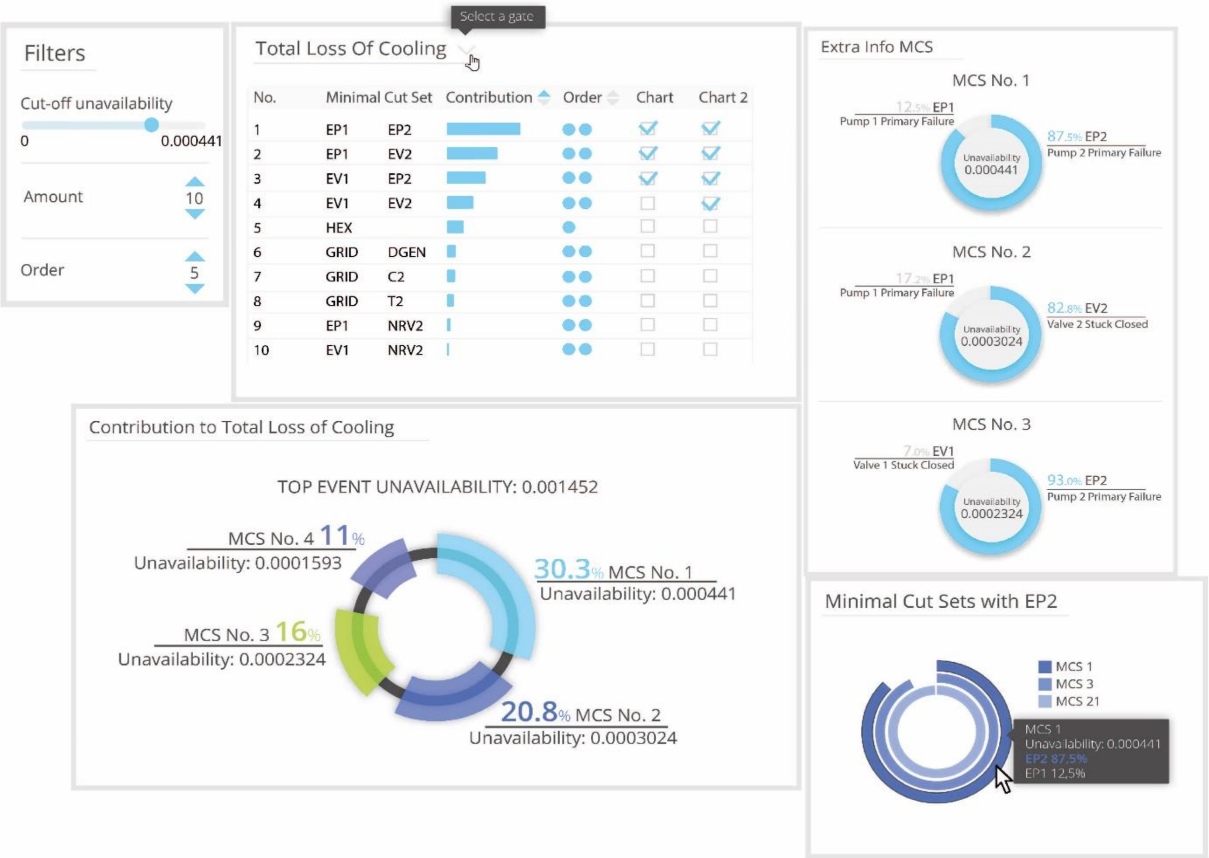


Figure 4.6:Total Product Concept.

Table – Block 1

The key ingredient in this layout is the table, it shows all the minimal cut sets which can be seen in Figure 4.6. A table has been chosen because a table can easily extend in size while still giving a clear overview. This unlimited extent in size is necessary knowing that fault trees can contain hundreds or thousands of minimal cut sets.

The content of the table consists of seven columns. At first, the number of the minimal cut set. Minimal cut sets do not possess an identifier like a title, therefor this number has been added to keep track of what minimal cut set is displayed in the table and charts. The number is based on the order of the minimal cut sets in the data which is loaded. Then there is the column minimal cut set, which shows all the events that are part of the minimal cut set. Only the title is being displayed so that the size of each cell will be kept as small as possible. In fact, showing only the title of an event is the way all analysed fault tree analysis programs presented their minimal cut sets. Hence, this makes the table even more accessible for users that are used to current presentations of minimal cut sets.

Next up is the contribution, this column shows the unavailability of each minimal cut set in length. Where current presentations of minimal cut sets display unavailability in scientific notation which is unfavourable for users who are less familiar with this notation. In fact, 90% found bars easier to find the highest or lowest and also more appealing than the scientific notation or decimal numbers. This resulted from a survey, see Appendix C for the results of this survey.

Subsequently, the column order shows the order (the amount of events in a minimal cut set) of the minimal cut set by representing each event with a circle. Not numbers but circles are chosen because it is easier to find the highest or the lowest. This was also a result from the survey which can be found in Appendix C.

Lastly, the two columns containing all the checkboxes. In general, checkboxes are used to select and unselect things which is exactly the function these two columns should have. Users will recognize these boxes as a checkbox and that selecting (clicking on) result in some sort of action.

Filters – *Block 2*

The block with filters has a simple and recognizable interface. The slider used in this sketch has been used in several applications as is visualized in Figure 4.7. Also, arrows indicate to increase and decrease the amount.

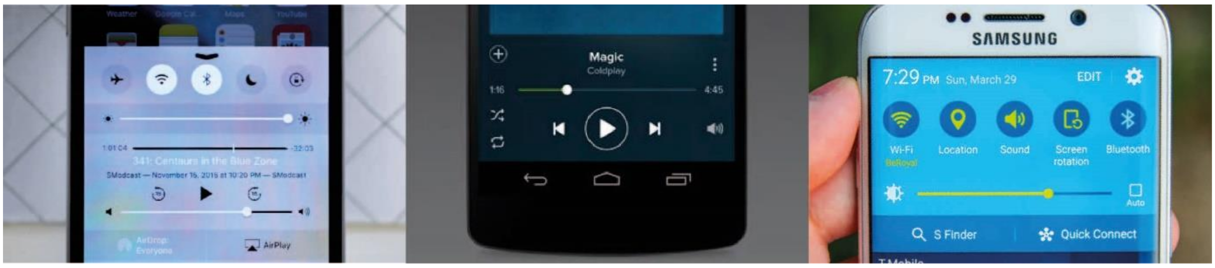


Figure 4.7: IPhone 6 Interface, Spotify Interface and Samsung Interface.

Extra Info MCS – *Block 3*

This chart its main goal to display all the events in the minimal cut set. Each part of the donut chart represents an event in the minimal cut set where the length of the arc represents the unavailability of each event. Meaning that the total donut covers the sum of the unavailability of each event. Aside the arc, the event title, description and a percentage is displayed. This percentage is calculated as in Equation 4.5 where k is the number of events in the minimal cut set.

Equation 4.5

$$\frac{P(Event_1\ fails)}{\sum_{n=0}^{k} P(Event_n\ fails)} * 100\%$$

Contribution of MCS to TE – *Block 4*

Each arc in this chart represents a minimal cut set, the length of the arc is the contribution of the minimal cut set to the top event. How this is calculated is explained in Equation 4.1. The black circle is always a complete circle because it represents the unavailability of the top level event. Each arc is accompanied by the percentage, the number of the minimal cut set to keep track of which one it is and the unavailability.

Event in Multiple Cut Sets – *Block 5*

Each arc represents a minimal cut set where the event appears in. The length of the arc is calculated the same as in ‘Extra info MCS’ but for this chart the other events are omitted. The colours are gradations from dark purple to light purple, the colour is added to emphasize the unavailability of the minimal cut set; the darker the colour, the higher the unavailability. Moreover, the placement of the arc emphasizes the unavailability of the minimal cut set. The highest unavailability is always the outer arc thus the biggest.

The three charts will be further referred to as chart 1, chart 2 and chart 3. Chart 1 being the Extra Info MCS, chart 2 Contribution of MCS to TE and chart 3 Event in multiple cut sets.

5 Realization

After the first product concept was developed, this concept has been transformed into a working online data visualization tool. This chapter will first explain this data visualization tool, after which the design choices made during this realization phase will be described. Lastly, the technical implementation of the data visualization tool will be discussed.

5.1 Result

The realization phase resulted in a working online tool which can be accessed through this link: http://www.djhuistra.nl/DataViz/datavis/index2.html. When the online tool is opened, it looks like in Figure 5.1. The only part already visible, apart from the layout, is the table. As explained in the ideation phase, a lot of interaction is necessary for full functionality of the interface. Therefore, the three blocks containing the charts are empty at the beginning. Visualizations will be shown in these blocks solely by interacting with the table.

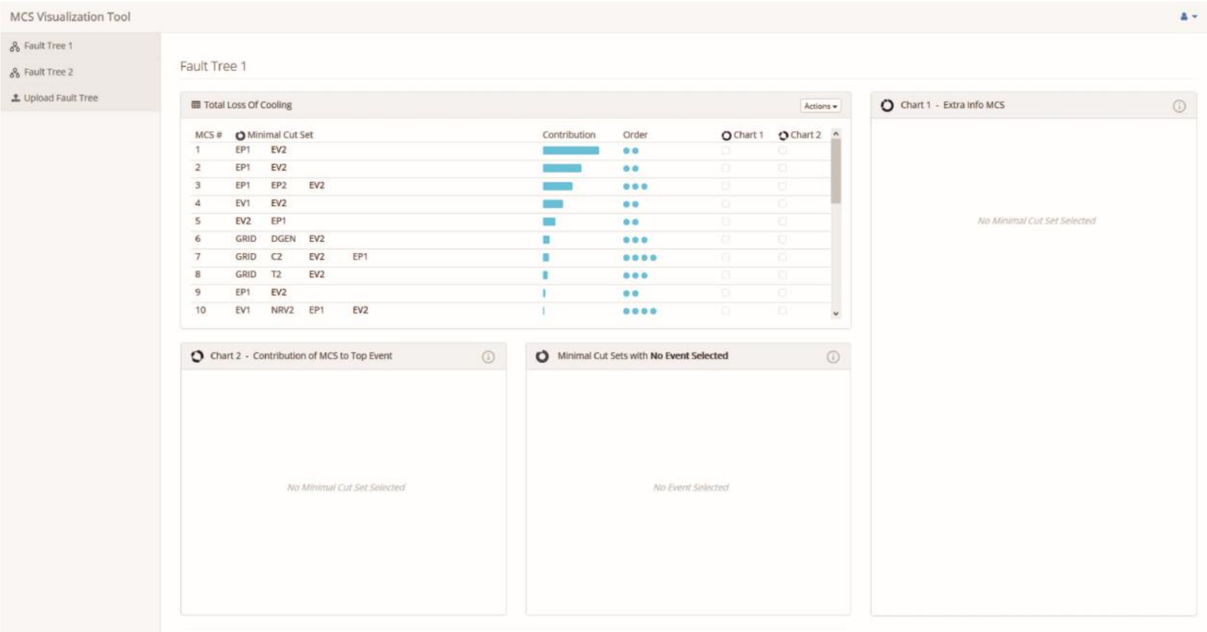


Figure 5.1: Start Screen Data Visualization Tool.

In Figure 5.2 the data visualization is shown after several interactions took place. Five minimal cut sets are selected for chart 1 with the checkboxes and four for chart 2. Also, event EP1 is selected in the table which resulted in the visualization of chart 3. As can be seen in Figure 5.2, the sketches from the Ideation phase have been implemented in the boxes. Also, most of the interactions have been implemented. Every minimal cut set can be (un)selected by clicking on the checkboxes. Similarly, an event can be selected and unselected by clicking on the event in the table. In contrast to chart 1 and 2, only one event can be selected and not multiple ones. When nothing is selected anymore, the tool will look like Figure 5.1 again.

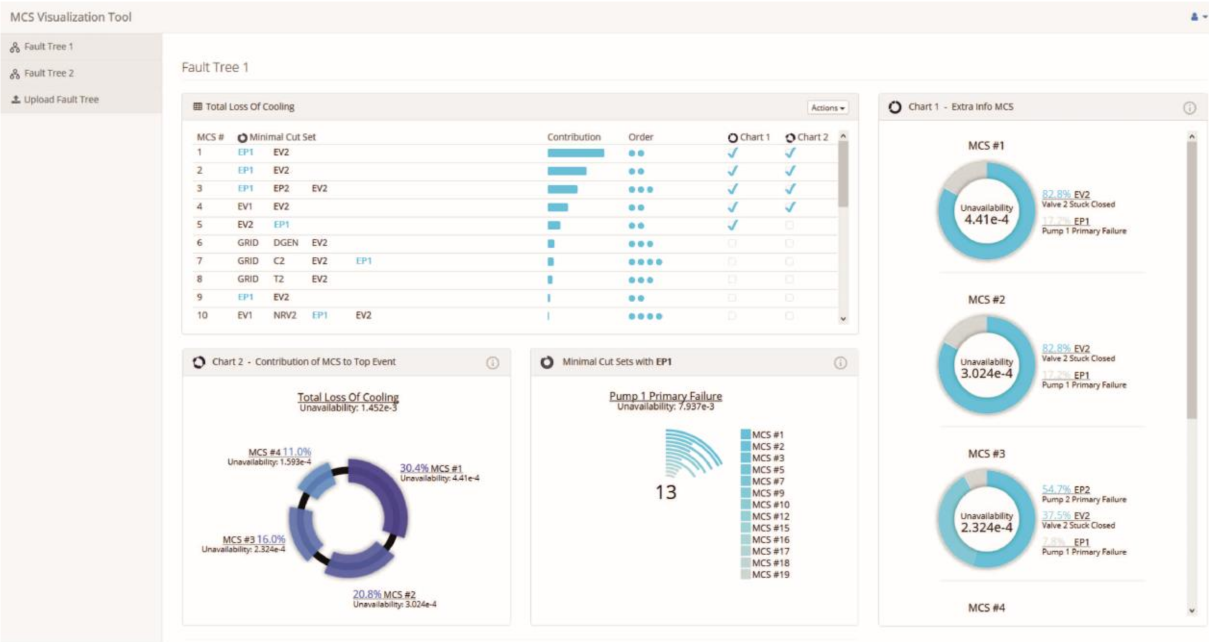


Figure 5.2: Impression of Data Visualization Tool with Selections.

Moreover, tooltips are implemented for chart 1 and 3 as shown in Figure 5.3. The tooltip for chart 1 shows the unavailability of the event that is hovered on. For chart 3 it shows the unavailability of minimal cut set and all the events in that minimal cut set with their relative contribution to the minimal cut set.

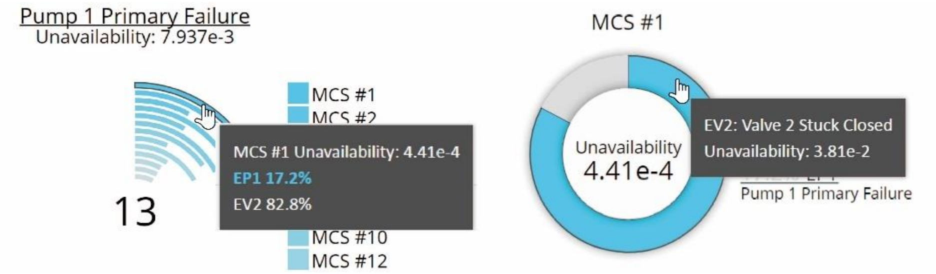


Figure 5.3: Tooltips of chart 3 and chart 1.

A big difference with the result from the Ideation phase is the presence of the menu on the left and on top. Note that these menus are not working yet. However, for future use and for appeal these have been implemented. The intention of the menu on the left is to navigate between different fault trees if there are multiple ones uploaded and to upload new data. The upper right menu is intended for logging in or when logged in you can see your user profile and adjust some settings. This menu is shown in Figure 5.4.

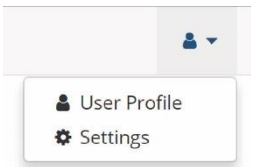


Figure 5.4: Upper Right Menu when clicked on.

5.2 Design Choices

During the realization of the data visualization tool, several additional design choices have been made. These design choices have been made in order to adapt to a large amount of cut sets, to increase intuitiveness and to increase visual appeal. Three requirements are not met because of the limited time frame. The next section describes what those decisions and requirements are and elaborates on why these have been made.

5.2.1 Adaptable to Large Amounts

Changing Legends

The legend of chart 2 must change when there is not enough space to display all the text based on the centres of the arcs. Therefore, a threshold value is set. It is a Boolean that checks if there is enough space between the arcs and the number of arcs that are displayed. The first condition is that the length of the space between arcs plus the last length of the arcs should be higher than 0.55 and the second condition is that there should be less than 11 minimal cut sets selected. If these conditions are not met, the legend will be tiled horizontally. Figure 5.5 shows what happens to the chart with each condition and when the conditions are combined. As can be seen in Figure 5.5, just having one condition is no option because text can get placed on top of each other or there is an immense amount of text circled around the chart.

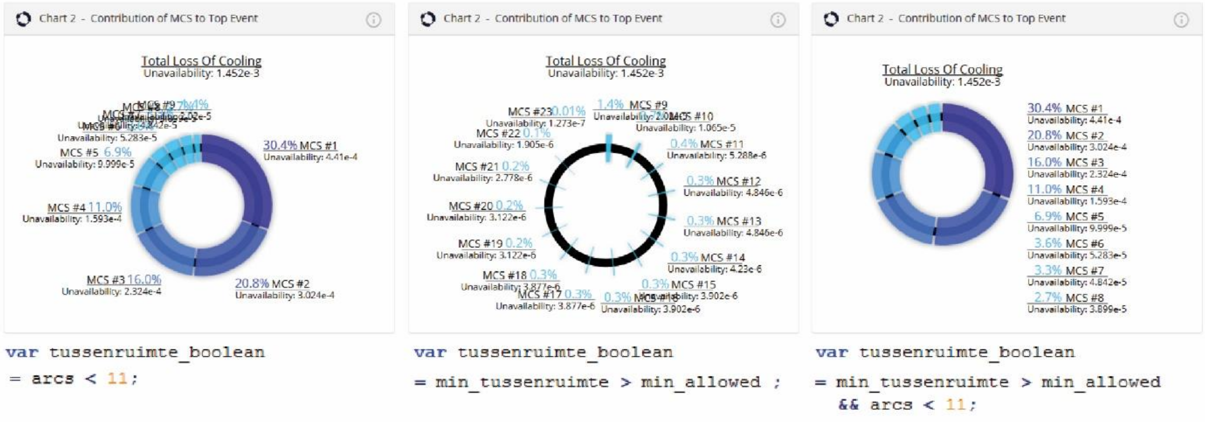


Figure 5.5: Chart 2 with different legend conditions.

Changing Blocks

The size of the blocks with charts must scale based on the amount of minimal cut sets. For chart 2 and 3 this entails that the size of the block should change in height if the legend will exceed the size of this block. If there are 15 minimal cut sets selected, the legend will not fit inside the block, in the same way this causes a problem for chart 2. Instead of increasing the size, chart 1 will get scrollable in the vertical axis whenever the height of all charts exceeds the height of the block. For further development, the same principle could be applied to chart 2 and 3 since these charts can get undesirably long. But for the time frame of this project this has not been implemented.

Changing Appearance

Also, the appearance of chart 3 must change according to the amount of cut sets. First the width of each arc decreases but as soon as an event is present in more than 14 minimal cut the arcs get significantly thin which makes it hard to distinguish the different lines and hovering over the arcs to see the tooltip is notably more difficult. This will only get worse when the number of minimal cut sets increases. Thus, the chart must change its appearance; as soon as it hits fifteen minimal cut sets the chart will increment with the same arc width as it would be with fourteen minimal cut sets. Figure 5.6 shows the differences in arc widths for the different amount of minimal cut sets.

Figure 5.6: Chart 3 visualizing 4, 7 and 20 minimal cut sets.

Numbers

Another small difference compared to the sketches is that the unavailability is presented in the scientific notation. The scientific notation allows for numbers to get very small without that the notation increases in size.

5.2.2 Intuitive Design

Affordance

Affordance has been added by means of visual elements to increase the intuitiveness. These visual elements are pointed out in Figure 5.7.

First, when the user hovers over any event in the table, that event title will appear in blue as well as the events with the same title in other minimal cut sets. Also, the cursor changes appearance. Both to indicate that the user can click on the events. To inform the user about what would happen, the corresponding block, chart 3, gets highlighted in blue. Whenever the user has clicked on an event, every event with that title stays blue and get bold to indicate the event that is selected.

Additionally, whenever the user hovers over a checkbox or the header of the checkbox columns the corresponding block gets highlighted. However, during interim evaluation, some students found it difficult to see the connection between the checkboxes and the charts when only the blue highlights were applied. Therefore, the header of the checkboxes is also visible in the header of the corresponding charts.

Moreover, there are three small other visual elements implemented. First, a row that turns in to grey when hovering on it to keep track of the row when a certain minimal cut set must be selected. Secondly, the cursor changes its appearance whenever the mouse is on clickable objects or on objects possessing of a hover action. Finally, text is added in each chart when nothing is selected. This text is '*No Event Selected*' or '*No Minimal Cut Set Selected*', to indicate that a selection has to be made to get something visible in those blocks.

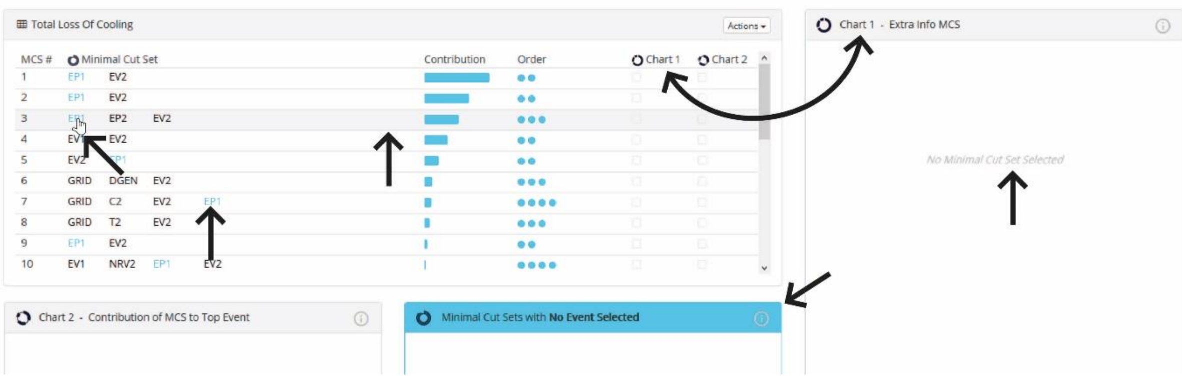


Figure 5.7: Visual Elements in Data Visualization Tool.

Guidance

Information buttons are added to guide the user when first trying out the data visualization tool so it is understood how to use the tool. Also, to inform the user about how certain calculations have been made. For instance, to explain how the 'contribution' has been calculated. Each chart has been given an information button to explain what is being visualized in that chart. The information is only displayed when the user clicks on the button so that there is no information overload.

Moreover, information pops up whenever the user hovers over a header of the table. Each header explains what is depicted in the column and if there is interactivity this will be mentioned and explained.

5.2.3 Visual Appeal

Look & Feel

To give the impression that the data visualisation tool is a tool a bootstrap template has been used. The benefits of such a template is that it contains predefined style settings which can be used to quickly get a familiar look and feel for websites or apps.

The layout of this template turns the screen into blocks, where the blocks always have the same ratio of the screen. There is a menu bar at the top and a menu on the left of the screen. These two menus and the styling of the blocks give the tool a more familiar impression. Moreover, the layout of this dashboard makes it easier to see that the table is the most important and that the user should start to interact with this table. Besides of having the advantage that the design of each block is predefined, this template has a responsive layout meaning that the sizes of the blocks adapt to the size of the window. Thus, zooming and resizing the window will cause no problems with the layout. The visualization tool without bootstrap template can be seen in Figure 5.8.

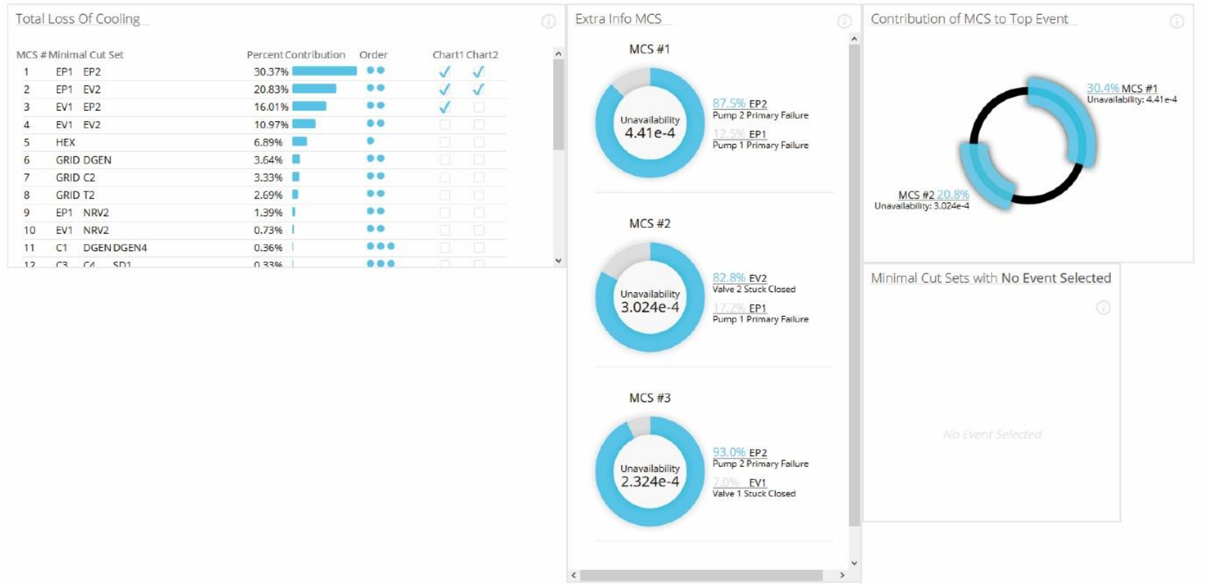


Figure 5.8: Data Visualization Tool without Template.

Colours

The tool uses several different colour pallets than the sketches. First of all, the hue of the arc in chart 3 have changed from purple in the sketches to blue. Blue has been chosen for more overall consistency. In contrast to the sketches, where chart 2 assigned random colours to the minimal cut sets, the tool assigns a colour to each minimal cut set. This colour is picked from an array of colours from purple to light blue, purple to the first minimal cut set and light blue to the last. This colour pallet has been chosen because it is less time consuming.

5.2.4 Unachieved Requirements

The data visualization does not meet three requirements from Chapter 4.1. First, there is no option to select a gate for which the minimal cut sets will be visualized. Secondly, there is no option to limit the amount of minimal cut sets loaded from the data in the tool. Thirdly, there is no option to sort the minimal cut sets on order or unavailability.

However, all of these options can be achieved by altering the data files. The data can be reduced and ordered in Excel and there are fault tree analysis programs available, among which RiskSpectrum, that have options to limit the amount of minimal cut sets before it is being calculated. This can be limited by a cut-off unavailability, an amount of minimal cut sets or by order.

Since there is a way that all of those functionalities can be achieved, these options are less innovative and only created as requirements to increase the functionality and the ease of use. Thus, these have been given a lower priority and considering the time frame, the decision has been made to exclude these options.

5.3 Technical Implementation

The implementation of the tool started with deciding what kind of tool should be used for the development of the data visualization tool. This decision will be explained first. Secondly, the code of the tool will be described and the fundamental functions will be explained by means of pseudo code. Next, the external source code that has been exploited as will be discussed. Eventually, the predefined format of the data input will be explained.

5.3.1 Development Tool

The D3 library has been used, it is a JavaScript library intended to visualize data [29]. D3 makes it possible to bind data to Document Object Models and then apply transformations to it which makes it easier, for example, to create a bar chart based on data. Another advantage is that with D3, attributes can be specified as functions of data. Using the D3 library resulted in that the tool has been developed in HTML, JavaScript and CSS.

5.3.2 Coding

Coding the tool has been done in html, JavaScript and CSS. For a more accessible code throughout the implementation phase, the total tool consists of several files to describe the functions of the table, charts, layout, the data and the styling. The foundation of the data visualization is the index2.html file, in this script the total layout is described as well as the table. Then there are five JavaScript files, these files contain functions to draw chart 1, chart 2, chart 3, circles and highlights. Furthermore, for the appearance of the tool, one CSS file is created. Last of all, external code is implemented for the aesthetics and functionality. The following section will describe the code for creating the table and charts. These codes can be found in Appendix D, apart from the external code.

Tabulate()

The function to draw the table is specified in index2.html. The function is called tabulate and acts upon the data it receives. First of all, the header of the table is made. The titles for the headers always remain the same, the only responsive part of the header is that the width of the second column depends on the amount of events in a minimal cut set. For the body, multiple columns are acting upon the data as can be seen in the pseudo code below.

```
var max_amount_events = mcsdata columns - 1;

var tabulate = function (mcsdata, eventdata, topeventdata, columns) {
```

```
    var headers = ["MCS #", "Minimal Cut Set", "Contribution", "Order", "Chart1",
"Chart2"];
    for (each header) {
        Create a header cell
        Add the header
        if (second column) {
            give it a colspan equal to max_amount_events
        }
    }

    for (each row in mcsdata) {
        Create a row
    }

    for (each row) {
        for (each column) {
            Create cell

            if (first column) {
                return row_number
            }
            if (second column && column <= max_amount_events) {
                return text_from_data
            }
            if (column = max_amount_events + 1) {
                add bars based on minimal cut set unavailability
            }
            if (column = max_amount_events + 3) {
                create checkboxes1
            }
            if (column = max_amount_events + 4) {
                create checkboxes2
            }
        }
    }
    for (each row) {
        var eventnames = [];
        for (var x = 1; x < max_amount_events + 1; x++) {
            if (there is an event title in data) {
                add that event to eventnames[]
            };
        }
        circles(j, eventnames.length, rows, color);
    }
}
```

This code is the foundation of the tool because every chart is called via this code. In the pseudo code above can be seen that the function circles() is called. Moreover, as soon as the state of a checkbox changes the function for chart 1 is called as can be seen in the pseudo code below. But before the chart gets drawn again it must be removed otherwise on every click a chart will be added upon the previous chart. The same goes for the text. Chart 2 is called with the same method.

```
for (each checkbox1) {
    if (change) {
        var checkedLines1 = [];
        for (each checkbox1 input)

            if (input is checked) {
                add row number to checkedLines1
            }
    });
    remove chart
    remove text
    chart1(mcsdata, eventdata, checkedLines1, max_amount_events);
});
```

The function for chart 3 is called whenever the user clicks upon an event title in the table and when the user clicks on an already selected event the chart will be removed as described in the pseudo code below.

```
for (each event in table) {

    if (clicked on) {
        if (highlighted events) {
            turn events to black
        }
        if (clicked event is already displayed) {
            remove chart 3
            add 'No Event Selected'
        } else {
            select all events with the same titles and highlight those
            remove chart 3
            chart3(mcsdata, eventdata, specifiedevent, max_amount_events);
        }
    }
}
```

Circles()

This function draws circles based on the amount of events in the minimal cut set. The variable length is the number of events in the minimal cut set and j is the row number.

```
var circles = function (j, length, rows, color) {
    for (each event) {
        select row j
        add a circle
    }
}
```

Chart1()

Chart 1 draws pie charts based on the amount of checkboxes are selected and which. The array hoeveelsteMCS contains the row number of each checkbox that is selected.

```
var chart1 = function (mcsdata, eventdata, hoeveelsteMCS, max_amount_events) {
    for (each checked checkbox1) {
        for (var x = 1; x <= max_amount_events; x++) {
            if (there is an event) {
                add event to eventnames[];
                add event unavailability to eventprop[];
            }
        }
        create pie chart based on eventprop[];
    }
}
```

Chart2()

The function chart2 draws arcs for each minimal cut set that is selected by the second checkbox. The function acts upon the array whatMCS which contains the row number of each minimal cut set that is selected. For each minimal cut set an arc will be drawn as described in the pseudo code below.

```
var chart2 = function (mcsdata, whatMCS, total, topeventdata, prev_checkedLines2) {

    draw black arc
    for (each minimal cut set selected) {
        draw arc
    }
}
```

For a normal pie or donut chart there is a function d3.layout.pie() which determines where each arc should be placed based on an array of numbers (continuously in a circle, from biggest to smallest). However, this chart is different, each arc needs to be given a start and end angle. The first arc always start at 0 and ends based upon it's unavailability relative to the total unavailability. For every other arc it is much more complicated. How this is calculated can be seen in the pseudo code below as well as in Figure 5.9.

```
var arc_lengte = 0;
var excess_space = 1;

for (each arc) {
    excess_space = excess_space - (minimal cut set unavailability / total
unavailability);
}

space = (excess_space / all arcs);

for (each arc) {

    if (first arc) {
        draw arc with
        .startAngle(0)
        .endAngle((minimal cut set unavailability / total unavailability) * 2 *
pi);
    } else {
        draw arc with
        .startAngle(arc_lengte + (tussen_lengte * drawn_arcs) * 2 * pi)
        .endAngle(arc_lengte + (tussen_lengte * drawn_arcs) + (minimal cut set
unavailability / total unavailability)) * 2 * pi);
    }
    arc_lengte = arc_lengte + (minimal cut set unavailability / total
unavailability)
}
```

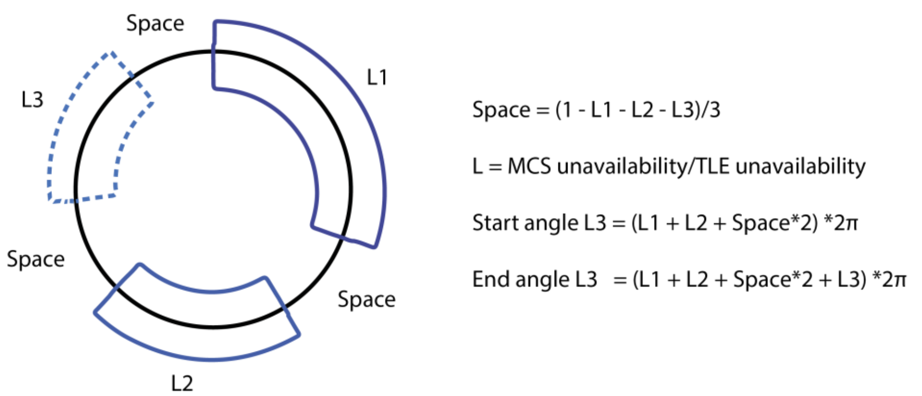


Figure 5.9: Start and End Angle calculation for Chart 3.

Chart3()

The function chart3 needs a variable specifiedevent as input. This is the event that is selected by the user when the user clicked on an event in the table. The function will check for each minimal cut set is the specified event is in there. Each minimal cut set with that event will get an arc, where the inner and outer radius must always be smaller than the previous one.

```
var chart3 = function (mcsdata, eventdata, specifiedevent, max_amount_events) {

    for (each minimal cut set) {
        if (specified is in minimal cut set) {
            add the row number to howmanyarcs
        }
    }
```

```
    for (each minimal cut set with specified event) {
        draw an arc
        with inner and outer radius smaller than previous arc
        and end angle = (specified event unavailability / total) * 2pi
    }
}
```

Bluehighlights()

The function bluehighlights describes what needs to happen whenever the user hovers over one of the checkboxes. The pseudo code below describes this function, the part for checkbox2 is the same and therefore omitted.

```
var bluehighlights = function (color) {
    if (mouseover on checkbox1) {
        select box of chart 1
        add color
    }
    if (mouseover on table header: Chart1) {
        select box of chart 1
        add color
    }
    if (mouseout) {
        return to normal style
    }
}
```

Also, for chart 3 the same highlights are provided, this is described in the function tabulate().

5.3.3 External Code

Apart from the D3 library, several other external codes have been implemented to increase the functionality and for aesthetic purposes.

Layout

The most prominent visible part of the tool is from Bootstrap [30]. Bootstrap offers many templates for web design for free. These can be used for any purpose, even commercial ones. The template is called SB Admin 2, it contains source code for responsive menu's, chart plugins, design and more, see Figure 5.10. However, for this data visualization only the responsive layout and layout design is used as can be seen in Figure 5.10. The source code can be found and downloaded on Bootstrap [31].

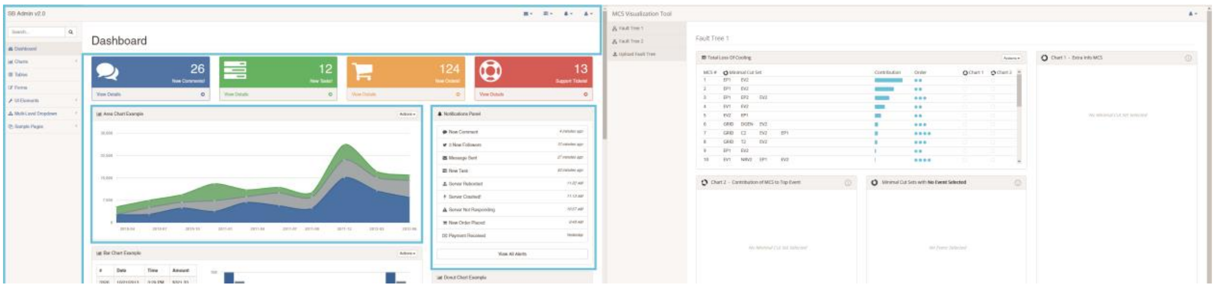


Figure 5.10: On the left the SB Admin template highlighting the similarities with the data visualization tool on the right.

Drop Shadow

Another bit of external code has been used for shadows since, simple CSS is not capable of applying shadow to path elements. It is a JavaScript code which can be used to apply shadows onto path elements. This drop shadow has been used to make the visualizations similar to the sketch in Figure 5.11. The code creates a variable filter which defines the drop shadow and this filter can be applied by giving the svg elements a filter attribute, which links to this created filter. The source code of the filter can be found on stackoverflow [32]. To display the filter on an element an attribute must be added to the element see the code below.

```
// Add a drop shadow to the circle
group.append("circle")
     .attr("r", outRadius)
     .attr("fill", "white")
     .attr("filter", "url(#dropshadow)");
```

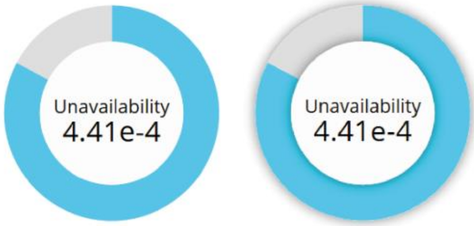


Figure 5.11: Donut Chart without and with Drop Shadow.

jQuery

Additionally, the jQuery library has been included [33]. It is a JavaScript library which contains numerous of functions which make JavaScript coding easier. It makes it easier because for the same task, JavaScript requires several lines of code to accomplish it where with jQuery a single line of code is enough. For this data visualization, several functions have been used in order to get the width and height of elements, the coordinates of elements and to set the html contents of elements without much effort and code. These functions are: .width(), .height(), .offset() and .html().

Moreover, a jQuery plugin is used [34]. This plugin makes the header of a table float, locked, sticky or fixed without special CSS. This plugin is used to make the header of the table in the data visualization float, meaning that it will not scroll as soon as the user is scrolling through the table and thus the header is always visible for the user. Note that this function was not working in the version of the tool used for the survey.

Font & Icons

Furthermore, the data visualization tool exploits some resources to enable fonts and icons. First, a font is used, Open Sans, that is not a web safe font, meaning that Windows, Mac and Linux are not likely to support this font. Therefore, this font is loaded from Google Fonts [35]. Google offers a free directory of open source designer web fonts which enables anyone to use non-web safe fonts. In the same way, icons are used. The table icon in the table header is available from a font called Material Icons which is also available through the Google directory [36]. As last, the info, person, settings and upload icons are from a font called Font Awesome which is created by Dave Gandy [37]. He offers a collection of icons which can easily be customized and used for the web. The icons from those sources can be seen in Figure 5.12.

Figure 5.12: Icons used from Material Icons and Font Awesome.

5.3.4 Data format

The format of the data is important considering that the tool acts upon this data, thus the data must be read in a way that connections between minimal cut sets and events are available. Moreover, it is important that the format stores the minimum amount of data so that processing time can be kept as low as possible as well as the amount of data that needs to be stored. At last, the data format should meet the requirements in Chapter 4.1. This all, resulted in that the tool needs three data sheets as input.

Data Files

The first csv file describes the minimal cut sets by containing the unavailability of the minimal cut set and the titles of the events in the minimal cut set. The second csv file contains all the information on the events: the title of the event, the description and the probability. The third csv file is quite small and only indicates the unavailability and the name of the top event, which is necessary for calculating the contribution of the minimal cut set to the top events probability. Since the total unavailability does not belong to one of the minimal cut sets or events this needs to be put into a separate file.

1

	A	B	C	D	E	F
1	**MCSUnavailability**	**Event1**	**Event2**	**Event3**	**Event4**	
2	*(MCS unavailability)*	*(event title)*	*(event title)*			
3	*(MCS unavailability)*	*(event title)*	*(event title)*			→
4	*(MCS unavailability)*	*(event title)*	*(event title)*	*(event title)*		
5	*(MCS unavailability)*	*(event title)*	*(event title)*	*(event title)*	*(event title)*	
6			↓			
7						

2

	A	B	C	D	E	F
1	*(event title)*	*(event title)*	*(event title)*	*(event title)*	*(event title)*	
2	*(event description)*	*(event description)*	*(event description)*	*(event description)*	*(event description)*	→
3	*(event unavailability)*	*(event unavailability)*	*(event unavailability)*	*(event unavailability)*	*(event unavailability)*	

3

	A	B
1	**TotalUnavailability**	**TLEName**
2	*(TLE unavailability)*	*(TLE name)*

Figure 5.13: Excel Format of each Data File.

No Limit

The datasheet should have no limit when it comes to the amount of cut sets and the amount of events. Therefore, for each minimal cut set a row can be added and for each event in the minimal cut set a new column can be made, see the format 1 in Figure 5.13.
For each event a new column has to be made see format 2 in Figure 5.13. Vertical alignment is chosen because D3 assumes that the first row of the data contains the column names. This means that the title of the event is the column name which makes it easier to access the data of each event since only the title of the event is needed.

Minimum Amount of Data

To get the minimum amount of data stored three separate files are needed because then data is repeated the least. Also with three files there are repeated items in the cells which can be seen in Figure 5.13, the event title is stored in both files (1 and 2). This is necessary because each event needs to be linked to one or more minimal cut sets and each event has one description and one unavailability. This method is taking up less space than a method where each unavailability of the event is included in the same row as the minimal cut set. Consequently, an unavailability can be saved repeatedly when events occur in more than one minimal cut set see Figure 5.14.

	A	B	C	D	E	F
1	**MCSUnavailability**	**Event1**	**Event2**	**....**	**Unav.Event1**	**Unav.Event2**
2	0.000441	EP1	EP2		0.007937	0.05556
3	0.0003024	EP1	EV2		0.007937	0.0381
4	0.0002324	EV1	EP2		0.004182	0.05556
5	0.0001593	EV1	EV2		0.004182	0.0381
6	0.00002020	EP1	NRV2		0.007937	0.002546

Figure 5.14: Format with Repetition of the Unavailability.

The tool only accepts csv files, therefore the data sheets need to be saved as csv files.
It loads data based on the amount of columns and rows in mydatamcs. Each column must contain an event otherwise the data visualization does not accept it.

6 Evaluation

Besides the intermediate feedback during the ideation and realization phase, a final evaluation has been performed. A survey has been made and sent to students at the University of Twente and experts and intermediate experts on fault tree analysis, the survey can be found in Appendix E. The following sections will describe the online survey and the results based on each user need.

6.1 Online Survey

The main goals of the survey are to evaluate the tool based on the user needs from Chapter 4.2 and to evaluate if the tool is more appealing and useful than current visualizations of minimal cut sets provided by fault tree analysis.

The technique used for the evaluation is an online survey in the hope that more experts can be reached. The survey consists of four parts: an intro with general questions, three tasks, an overall evaluation of the tool and expert questions. The survey can also be accessed through the following link https://utwentebs.eu.qualtrics.com/jfe/form/SV_6f06MGbYmKzkpoh.

The survey starts with a little introduction to thank and inform the participant. After that, some general questions are asked. The goal of these questions is to get some demographics on the participant so that the opinions of experienced, intermediate-experienced and non-experienced can be analysed individually.

Subsequently, the participant should execute three tasks with the data visualization tool. The tool can be accessed through a link which is provided by the survey. Each task focuses on one chart of the tool. After every task, the participant should answer questions related to the task and some general questions about the chart itself. The general questions are focused on getting the participant's opinion on the appeal, the ease of use, the intuitiveness and the information shown.
Next, questions will be asked about the entire data visualization tool. These are also focused on the user needs. During these questions the participant is allowed to freely use the data visualization tool to try out everything.

Afterwards, the participant will be asked if they have used any kind of tool for fault tree analysis. If they have, they will be asked to answer some more questions. These questions are aimed at getting opinions on how this tool visualizes minimal cut sets compared to current visualizations of minimal cut sets.

The online survey has been made available to students and experts/intermediate experts on fault tree analysis. Apart from the experts, students can also fill in the survey because even though they might not understand the jargon used, they can give their opinion on the visual aspects or usability.

6.2 Results

This section first describes the participant's characteristics of the survey responses. Secondly, the results for each user need and the results compared to current visualizations of minimal cut sets will be described. Lastly, the feedback obtained from the responses will be discussed.

6.2.1 Responses

The survey resulted in 24 responses. Five participants are extremely familiar with minimal cut sets and fault tree analysis. Three of them work as consultants in the field of Risk & Reliability and two of them are project engineers. Four are slightly familiar with minimal cut sets whereof one works as an assistant professor, the rest is student. The last fifteen participants are students who are not familiar with minimal cut sets. The age of the participants lies within the range of 19 to 34.

6.2.2 Intuitive

Affordance

To evaluate the affordance, the visual clues that were used during the tasks were evaluated by asking if they were useful and if other clues were found useful by the participant. The visual clues that have been evaluated are the blue highlighted text, the cursor that changes appearance, blue highlights on charts and the grey highlight on the row. In Figure 6.1 can be seen how many participants found these clues useful.

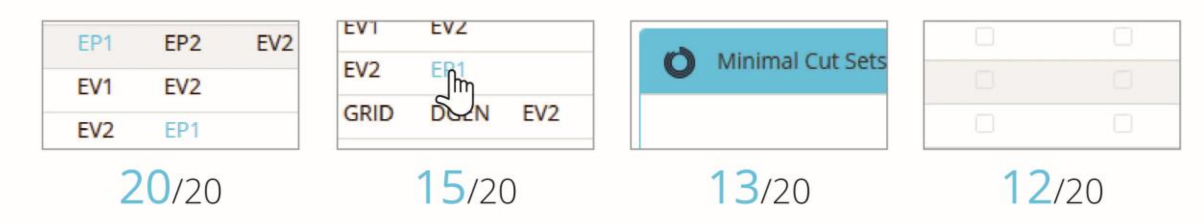
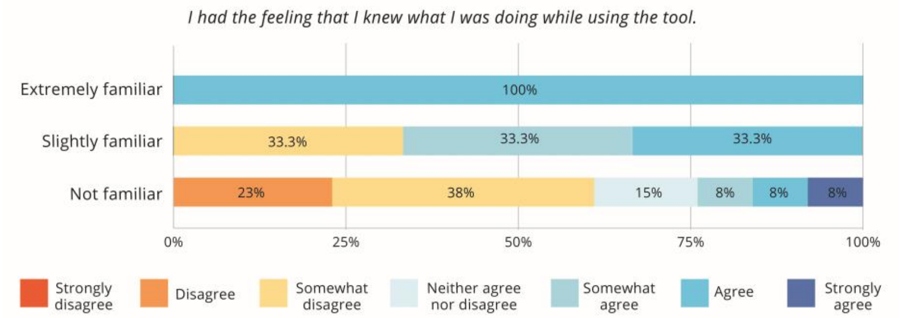


Figure 6.1: Visual Clues with indication of how many participants found these useful in blue.

Other useful visual clues that the participants found useful are the info buttons (2 participants), the info that appeared when hovering over the titles of the columns (3 participants), the tooltips (2 participants) and the animated charts (1 participant).

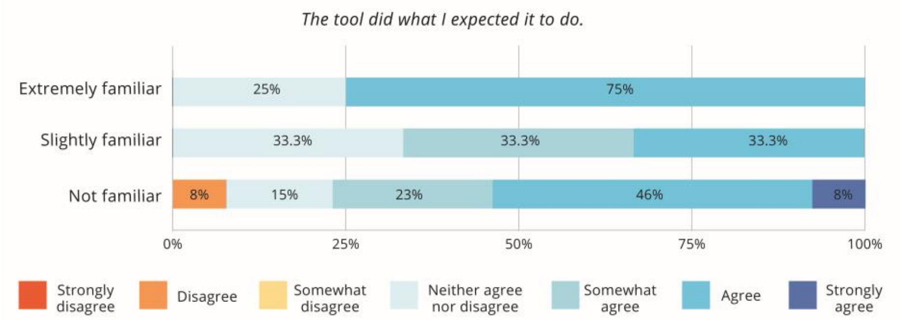
Predictability

To evaluate the predictability of the tool, the participant has been given two statements: I had the feeling that I knew what I was doing while using the tool and the tool did what I expected it to do. The results can be found in Graph 6.1 and Graph 6.2.



Graph 6.1: Results of the Statement: I had the feeling that I knew what I was doing while using the tool.

The extremely familiar group all agreed to this question but the less familiar the participants are with minimal cut sets, the less they agreed. This can be caused by their lack of knowledge of all the jargon presented.



Graph 6.2: Results of the Statement: The tool did what I expected it to do.

In contrast to the previous statement, the not familiar group agreed to that the tool did what they expected it to do for more than 70%. Agreed being somewhat agree, agree or strongly agree.

Responsiveness

To evaluate the responsiveness of the tool, every available action with the tool will be evaluated.

First of all, the actions by clicking. As soon as there is being clicked on a checkbox, an event or an information button, a chart or textbox appears or disappears. Chart 3 appears with an animation which catches the attention of the user even more. Likewise, in chart 2 the last piece of the donut that has been added appears with an animation to make clear which piece is selected as last. This animation effect has not been added to chart 1 because of the time period.

If something has been selected there are also indications for this. First, the checkboxes themselves indicate if something is still visible or not with a checkmark. Similarly, the info icons are darker when selected. Moreover, the event that is being presented in chart 3 is blue and bold as long as it is visible in chart 3.

Also, if nothing is selected, there is text in the boxes that says that nothing is selected which subsequently indicates that the user should select something. Another hint is in de header of chart 3 which also indicates if there is no event selected.

Next, the hovers. For every hover possible, something will immediately appear. Moreover, when hovering over chart 1 or 3, a tooltip appears. The arc that has been selected will get an outline to indicate for which the tooltip is shown. For chart 2, there is no tooltip. However, when hovering over one minimal cut set this one will be accentuated and the other minimal cut sets will be less visible. This can be seen in Figure 6.2.

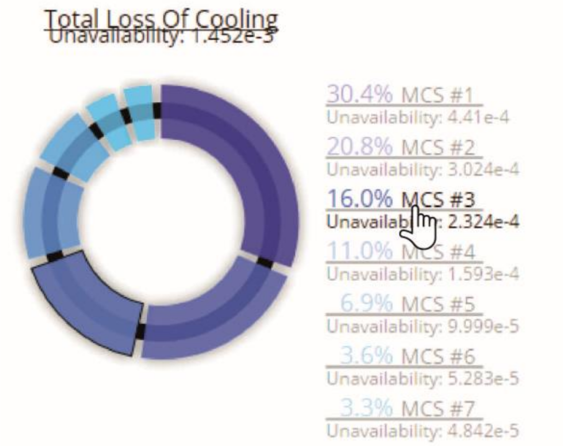
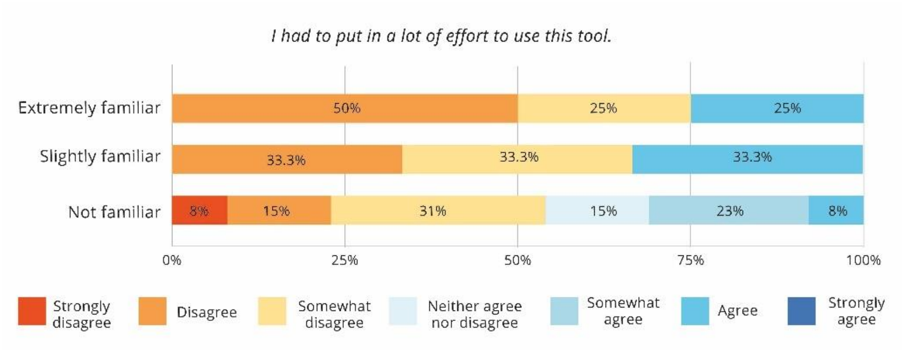


Figure 6.2: Chart 2 with Highlighted and Less Visible Parts.

Thus, the tool shows that it is responsive since feedback is immediately given to the user for every action the user can perform.

Efficiency

To evaluate the efficiency of the tool, the following statement has been presented: I had to put in a lot of effort to use this tool. Note in Graph 6.3, red, orange and yellow is disagreeing with the statements which is positive for the efficiency.



Graph 6.3: Results of the Statement: I had to put in a lot of effort to use this tool.

More than 50% of each group did not have to put in a lot effort to use this tool. Furthermore, only three persons agreed with the statement whereof had experienced difficulties because the selection boxes were not visible and the elements did not fit in the boxes.

6.2.3 Ease of Use

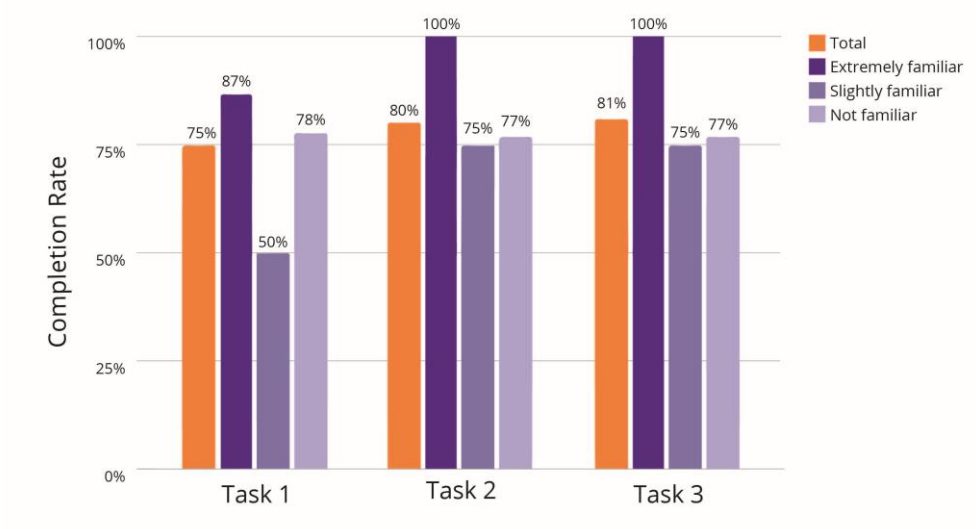
Effectiveness

To measure effectiveness a successful completion rate is calculated. This is calculated as in Equation 6.1.

Equation 6.1

$$Succesfull\ Completion\ Rate\ = \frac{Number\ of\ tasks\ completed\ succesfully}{Total\ number\ of\ tasks\ undertaken}\times 100\%$$

In Graph 6.4 the completion rate can be seen of the total number of participants and the groups with different familiarity of minimal cut sets. In the Graph, it can be seen that the more familiar someone is with minimal cut sets the better they are at completing the tasks with the tool. However, the groups extremely familiar and slightly familiar consisted only out of five and four participants.



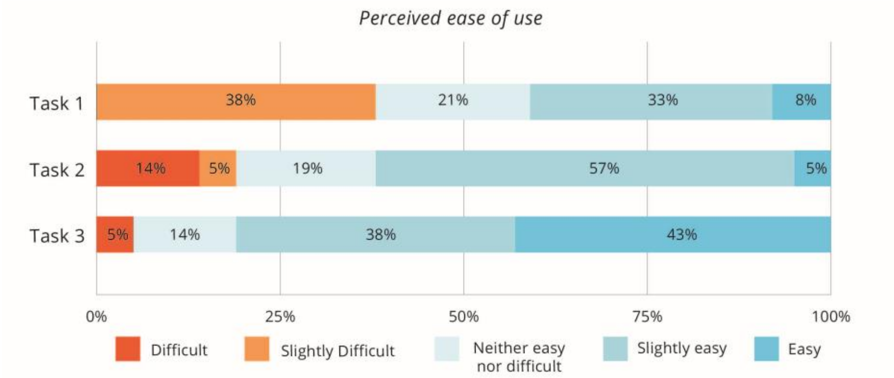
Graph 6.4: Completion Rate per Group.

Task 2, the ones who had it wrong made mistakes with adding up the percentages but did get something visible or did not get something visible because the checkboxes were not visible.

Task 3, only four persons did not fill in the right answer. Three of them answered with 20 which is the answer to another event EV2 which is quite like EP2. The other, did not understand the question.

Satisfaction

To measure satisfaction the Single Ease Question [38] is asked after every completion of a task. It can be used to assess the overall ease of the completion of the task with a 5-point Likert scale. After this question the participant was asked why they gave the answer to the SEQ. The results of the Single Ease Question can be found in Graph 6.5.



Graph 6.5: Results of the Question: How easy was it to do execute the task?

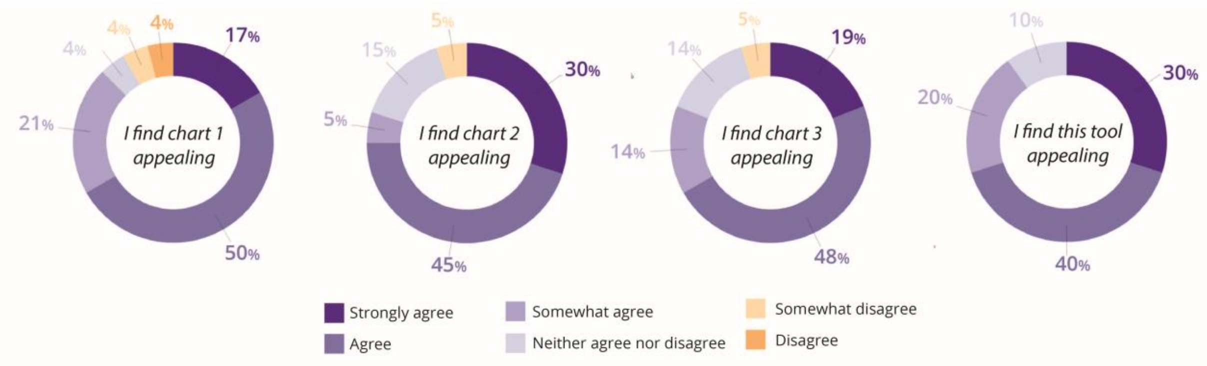
The first task was the most difficult for the participants. The participants that answered slightly difficult, had difficulties with the size of the window; they had to scroll to the right in the table to access the checkboxes and they did know where to look or click at first. The participants that answered neither easy nor difficult gave those same reasons. One quote 'I had to get accustomed to the interface and how to operate it, once I got the hang of it was rather easy :)'.
Even though there were 15 participants that are unfamiliar with minimal cut sets, only three of them answered the first task wrong and only two mentioned that they found it difficult because the acronyms were unknown to them.
The second task was easier in general, but there were three participants who answered difficult. Their reason was that they could not locate the checkbox for Chart 2 and because the percentages needed to be added up. All of the 'complaints' were about confusion of the task itself and difficulty because the percentages needed to be added up. Compared to task 1, no one complaint about not knowing where to look or click.

Task three was the easiest for the participants. By looking at the reasons that the participants gave for their answers, most of them answered that the task was very intuitive and only one click was needed. Thus, they knew what they had to do. Only one participant found it difficult and gave as reason that he/she did not understand it. The task was found neither easy nor difficult because of unknown terms (events), not sure what the number in the middle mean.

There is a learning effect present. One answered 'knowing where to look after doing the previous question made it easier to find the check boxes for this one', another 'I am more familiar with the interface now'. This effect can also be seen in Graph 6.5, the more tasks that have been completed, the easier the next task is.

6.2.4 Appeal

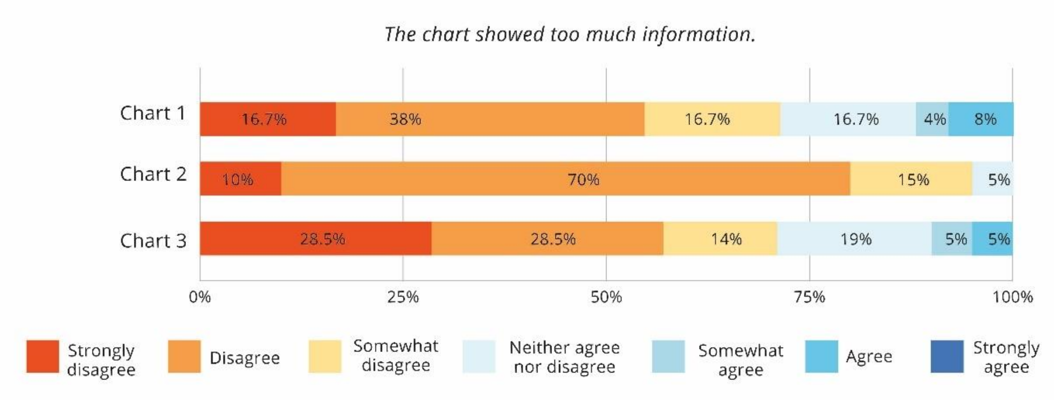
Each chart is being perceived as appealing for 80% or even more, and 90% of the participants found the total tool appealing. Appealing meaning that they somewhat agreed, agreed or strongly agreed with the statement inside the donut charts in Graph 6.6.



Graph 6.6: Perceived Appeal for each Chart and the Total Tool.

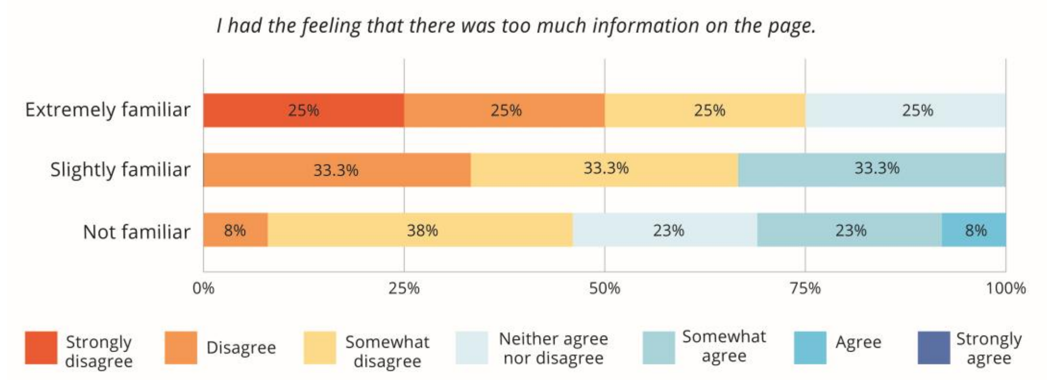
6.2.5 Avoid Information Overload

For each chart, the participant had to answer the statement: the chart showed too much information. This can be seen in Graph 6.7. For each chart, more than 70% disagreed with the statement. Chart 2 was perceived as the chart with the least amount of information overload since no one agreed with the statement.



Graph 6.7: Results of the Statement: The chart showed too much information.

Another statement has been evaluated about the information shown on the entire interface. These results can be seen in Graph 6.8. There is a correlation between the familiarity with minimal cut sets and the feeling of information overload. The less familiarity the more they had the feeling of information overload. This is quite logical if you think about the fact that the whole interface is about minimal cut sets but you know nothing about that.



Graph 6.8: Results of the Statement: I had the feeling that there was too much information on the page.

6.2.6 Compared to Current FTA Tools

Five participants have experience with other fault tree analysis tools. These tools are Excel, Isograph FT+, Fault Tree Analysis (NRG), WinNUPRA, Riskspectrum and DFTCalc. Compared to how these tools visualizes minimal cut sets, this tool is found to be more appealing as can be seen in Figure 6.3.

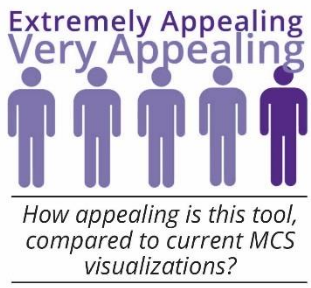


Figure 6.3: Results of the Question: How appealing is this tool, compared to current MCS visualizations?

Also, the participants agreed to that this data visualization is more useful for presenting minimal cut sets. Moreover, four agreed that this tool showed more interesting information on minimal cut sets and one did not agree nor disagree.

6.2.7 Feedback

A lot of feedback resulted from the survey. the most relevant and repeated feedback will be mentioned here.

One point that stood out the most is the problem with visibility of the charts and table. Before executing the survey, it was known that these would not scale automatically but it was not expected that it was causing such a big problem. So, a lot of participants mentioned that the tool should be responsive. Also, mentioned by many participants is the fact that the colours of the charts were hard to distinct. Although, the colours were appealing. Three participants mentioned that chart 2 should be changed or an option to change it. Namely, changing the chart by removing the empty spaces between the arcs, or an option to switch between those. This would make it easier to see the total percentage of the arcs. Another suggestion is to put the total percentage of the arcs in the middle.

There were also other interesting suggestions for improvement. Among which that the last sentence of the info button of chart 1 should read: The bluer and bigger the arc, the higher the relative contribution of the event to the minimal cut set. This was mentioned by an ‘expert’. Moreover, an (un)hide option for the minimal cut sets in chart 3, because if there are many minimal cut sets it can be annoying. Also, a hover over the bars and circles in the table could show the actual number or percentage or unavailability. Likewise, a hover on the checkboxes giving a short instruction telling what to do. To increase the functionality even more an export option could be added to save charts to images. Furthermore, a search bar can be added to search or filter the results. Finally, an option to change the order of minimal cut sets alphabetically, contribution or order could be added.

7 Conclusion & Future Work

7.1 Conclusion

This research has shown that using common graph types, visual clues and interaction techniques can be used in such a way that minimal cut sets are visualized intuitive and more appealing. The goal of this project was to create a stand-alone tool that visualizes the results of fault tree analysis with only data as input. By doing a state of the art analysis inspiration was found. This and interviews led to the first product concept which was then translated into a stand-alone tool. The data visualization tool uses visual clues, common graphs and interaction techniques to visualize minimal cut sets. The evaluation has shown that this tool is appealing. Moreover, experts found this visualization of minimal cut sets more appealing than current visualizations. Also, the tool was found to be intuitive through its visual clues. However, the tool lacked intuitiveness caused by the absence of responsiveness. Furthermore, the tool is easier to use when the tool has been used before.

7.2 Future Work

Further research is necessary to make intuitive and appealing visualizations of other available fault tree analysis techniques possible. Also, more research has to be done to explore the best data format. Such a format should be able to exchange data from fault tree analysis software to this tool. Also, this tool can be developed. When further research has found ways of visualizing other analysis techniques, this can be implemented into this stand-alone tool. Multiple tabs can be used to navigate through the different types of analysis techniques. Moreover, the tool already shows an interface where multiple minimal cut sets from different fault trees can be displayed, this can be made functional. Data can be uploaded, users can log in and look at previously saved projects. To make the tool even more practical, charts can be saved as image files. This data visualization tool can increase its intuitiveness by having great responsiveness. To increase the intuitiveness even more, more interactions can be implemented. For instance, an option where the user can (de)select every checkbox. Finally, all the feedback and the unachieved requirements can be integrated in order to increase the functionality and to adjust the tool to the needs of its users.

8 Appendices

8.1 Appendix A: Criteria & Scores FTA Software

Interaction: The quality and quantity of the interaction possible with the fault tree and the results. An interaction of value means that the interaction gives the user more insight or helps the user to easily explore the data or the fault tree.
1 = No interaction possible
2 = At least one interaction is of value
3 = At least two interactions are of value
4 = At least three interactions are of value
5 = At least four interactions are of value

Navigation: The ease of getting results and switching between results and the fault tree.
1 = Very difficult
2 = Difficult
3 = Not easy, nor difficult
4 = Easy
5 = Very easy

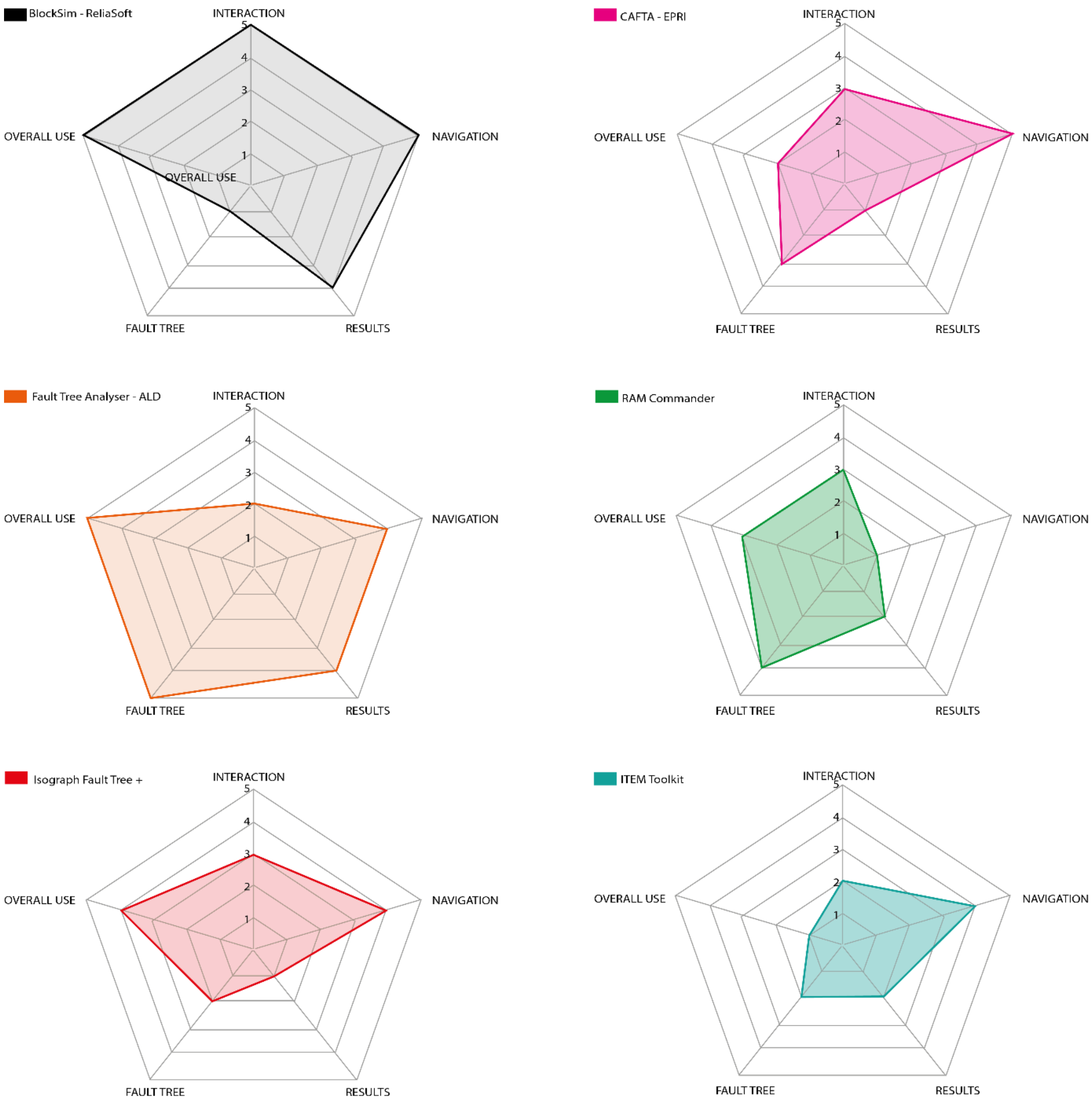
Results: The effectiveness and aesthetics of the representations of the fault tree analysis results.
1 = The results are very difficult to interpret and poorly visualized.
2 = The results are difficult to interpret and the visual elements do not help in understanding the results.
3 = The results are neither easy or hard to interpret and the visual elements hardly help in understanding the results.
4 = The results are easy to interpret and did contain some visual elements that help in understanding the results.
5 = The results are very easy to interpret and nicely visualized.

Fault tree: The effectiveness and aesthetics of the representation of the fault tree.
1 = The fault tree is very ineffective and not appealing represented.
2 = The fault tree is ineffective and not so appealing represented.
3 = The fault tree is neither bad nor good represented.
4 = The fault tree is appealing and effectively represented.
5 = The fault tree is very appealing and effectively represented.

Overall use: The effectiveness and efficiency of the program.
1 = Every time using this program, it is still difficult using the program and knowing where to find something.
2 = Even after trying multiple times, it was hard to remember where to find something and how to use the program.
3 = After trying the program more than three times, it was possible to remember where to find something and how to use the program.
4 = After trying the program once or twice, it was easy to use the program and knowing where to find something.
5 = It was instantly clear how to use the program and knowing where to find something.

Table A.1: FTA Software scores for the criteria

Program	Interaction	Navigation	Results	Fault Tree	Overall Use	TOTAL
Isograph Fault Tree+	3	4	1	2	4	14
Fault Tree Analyser - ALD	2	4	4	5	5	20
ITEM Toolkit	2	4	2	2	1	11
RAM Commander	3	1	2	4	3	13
CAFTA - EPRI	3	5	1	3	2	14
BlockSim - ReliaSoft	5	5	4	2	5	21



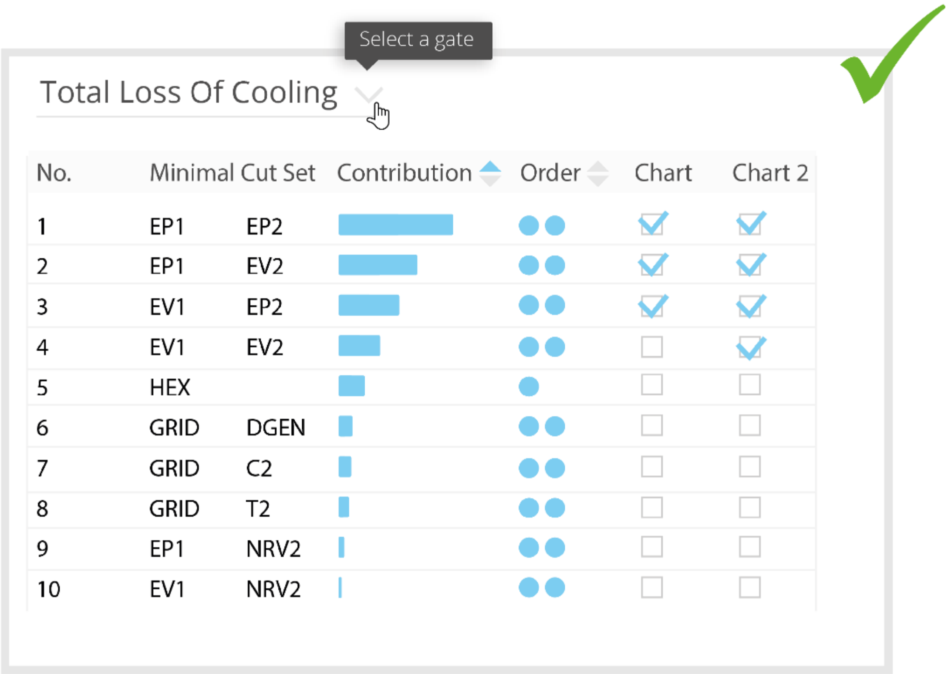
Graph A.1: Radar Chart showing the score for each software separately.

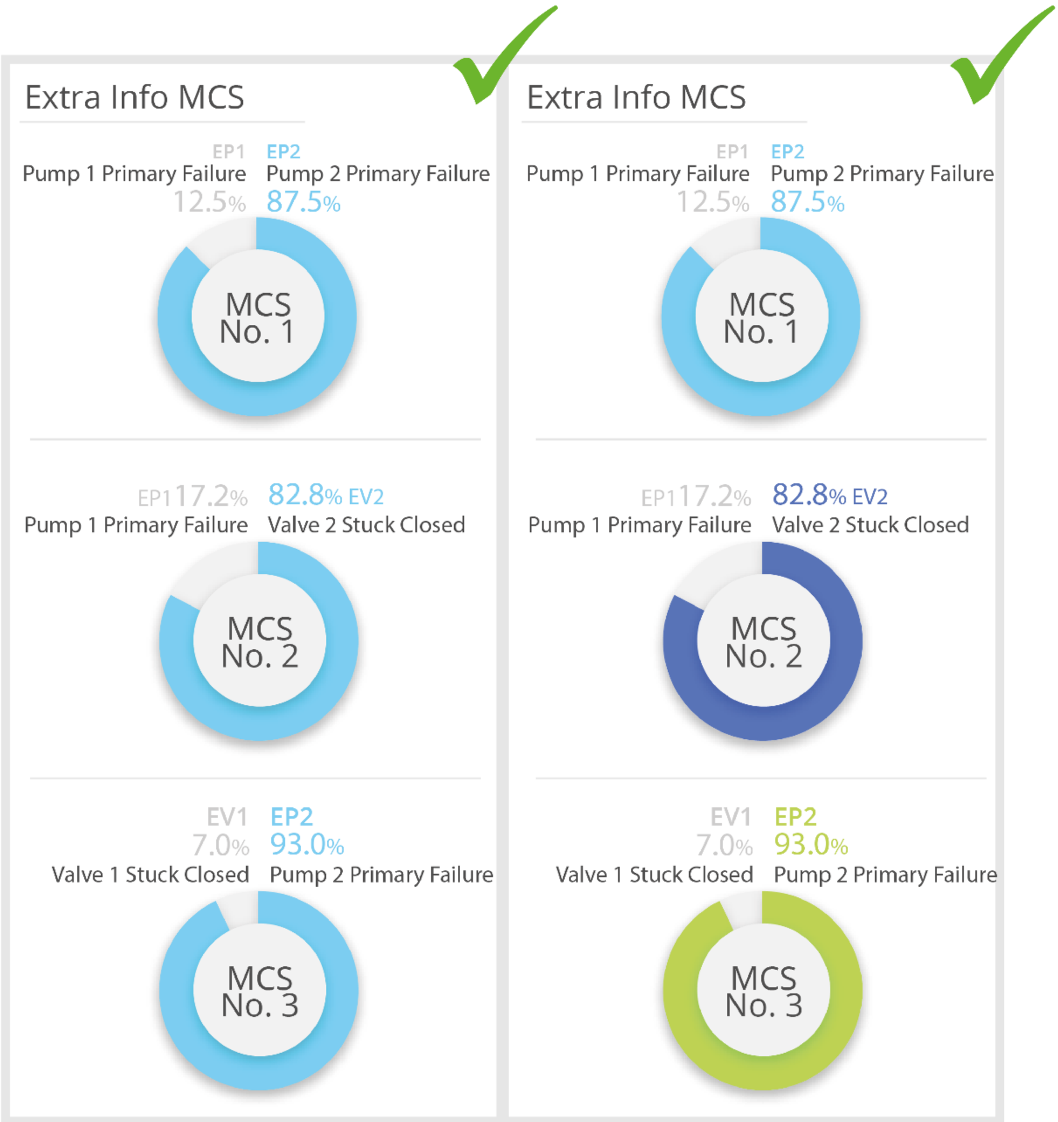
		Isograph Fault Tree +	Fault Tree Analyser - ALD	ITEM Toolkit	RAM Commander	CAFTA - EPRI	BlockSim - ReliaSoft
Qualitative methods	MCS	+	+	+	+	+	+
	MPS	-	-	-	-	-	-
	CCF	-	+	-	+	+	-
Stochastic Measures	(un)Reliability	+	+	+	+	+	+
	(un)Availability	+	-	+	+	+	+
	ENF	+	-	+	+	-	+
	MTTF	+	-	+	-	-	+
	MTTR	+	-	+	-	+	-
	MTBF	+	-	+	-	-	-
Importance Measures	MCCS	±	±	±	+	-	+
	Birnbaum	+	-	+	-	+	-
	Fussel-Vesely	+	-	+	+	+	-
	Barlow-Proschan	+	-	+	-	-	-
	RRW	+	-	+	+	+	-
	RAW	+	-	-	+	+	-
	Sequential	+	-	-	-	-	-
Sensitivity	High/Low Sensitivity	-	-	-	+	-	-

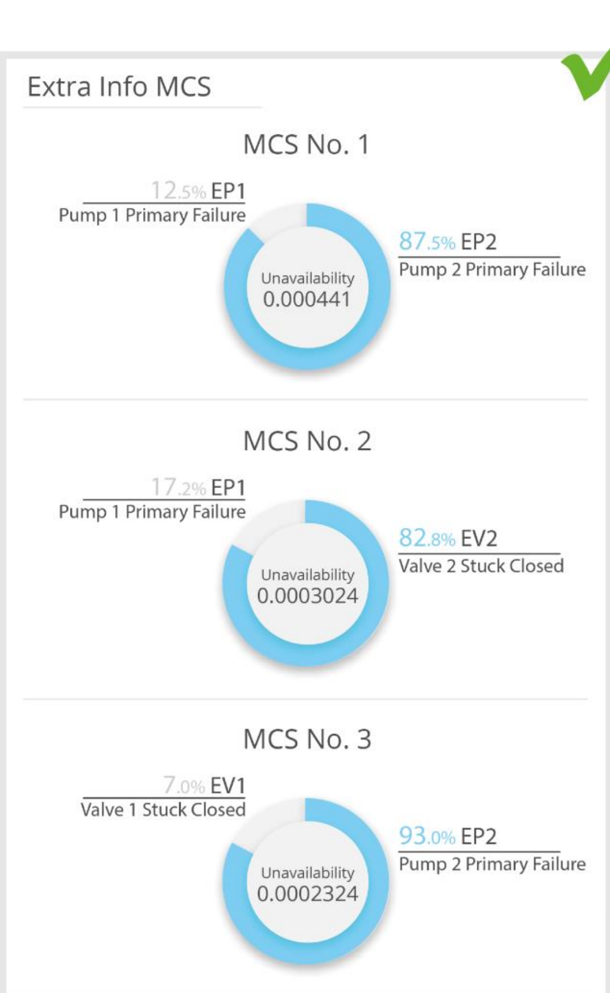
Table A.2: Qualitative and Quantitative Analysis per Software.

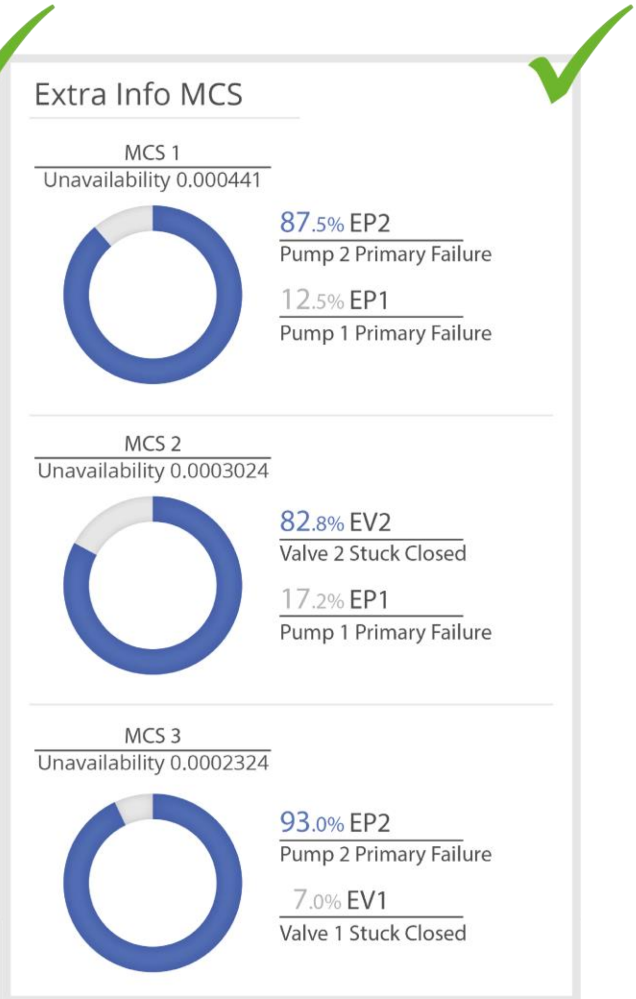
± shows the minimal cut set sizes but does not rank them.

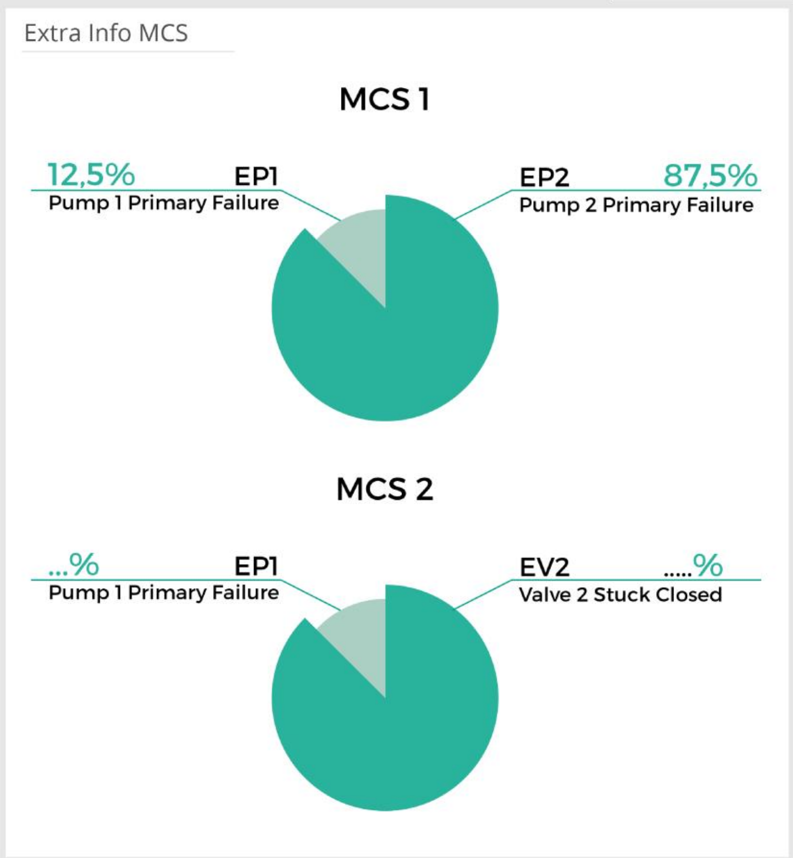
8.2 Appendix B: Sketches

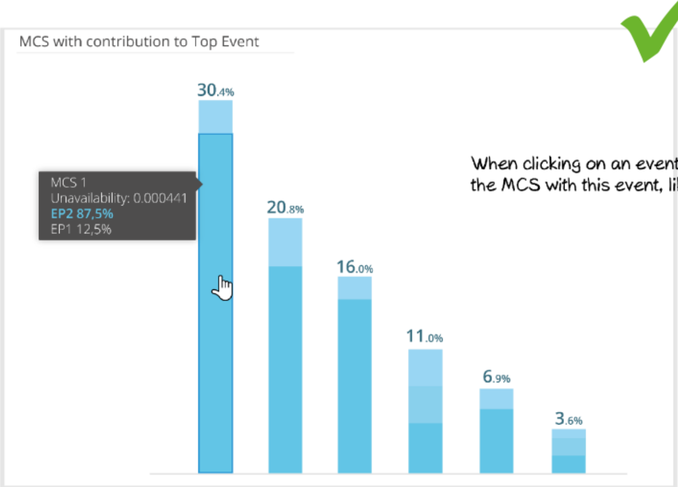




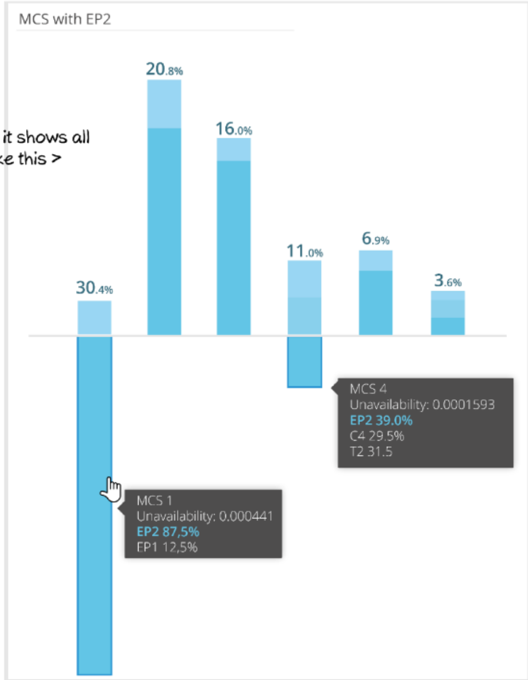

Extra Info MCS
MCS No. 1
12.5% EP1
Pump 1 Primary Failure
Unavailability
0.000441
87.5% EP2
Pump 2 Primary Failure
MCS No. 2
17.2% EP1
Pump 1 Primary Failure
Unavailability
0.0003024
82.8% EV2
Valve 2 Stuck Closed
MCS No. 3
7.0% EV1
Valve 1 Stuck Closed
Unavailability
0.0002324
93.0% EP2
Pump 2 Primary Failure

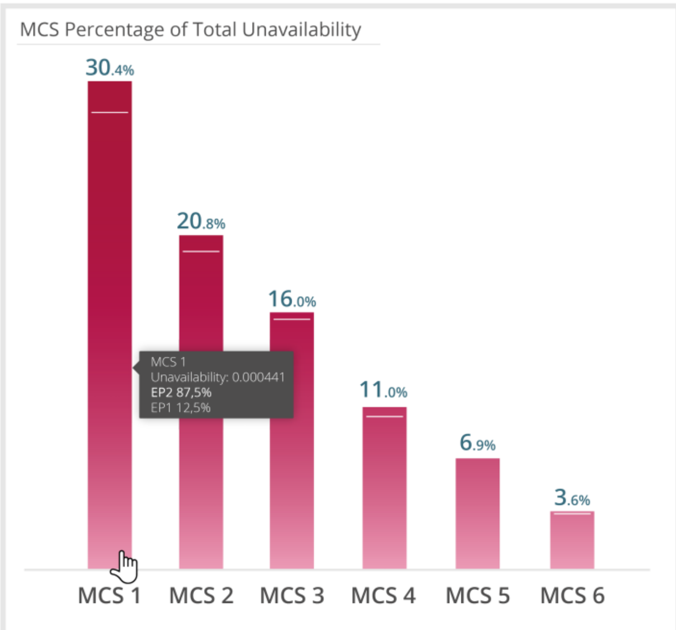

Extra Info MCS
MCS 1
Unavailability 0.000441
87.5% EP2
Pump 2 Primary Failure
12.5% EP1
Pump 1 Primary Failure
MCS 2
Unavailability 0.0003024
82.8% EV2
Valve 2 Stuck Closed
17.2% EP1
Pump 1 Primary Failure
MCS 3
Unavailability 0.0002324
93.0% EP2
Pump 2 Primary Failure
7.0% EV1
Valve 1 Stuck Closed

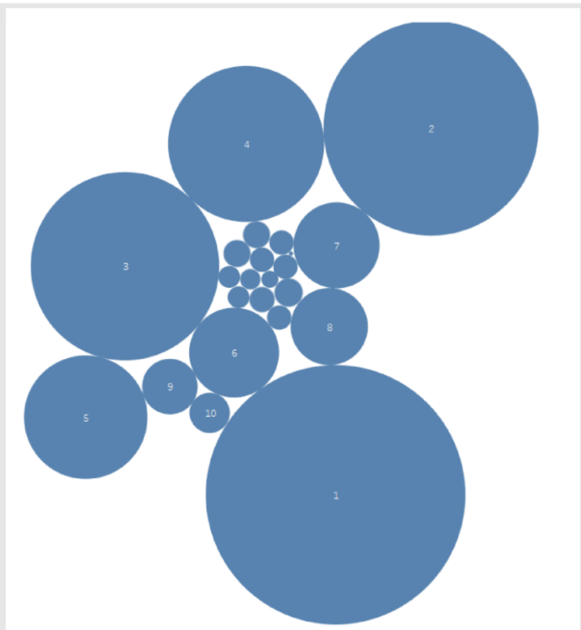

Extra Info MCS
MCS 1
12,5% EP1
Pump 1 Primary Failure
EP2 87,5%
Pump 2 Primary Failure
MCS 2
...% EP1
Pump 1 Primary Failure
EV2 .....%
Valve 2 Stuck Closed



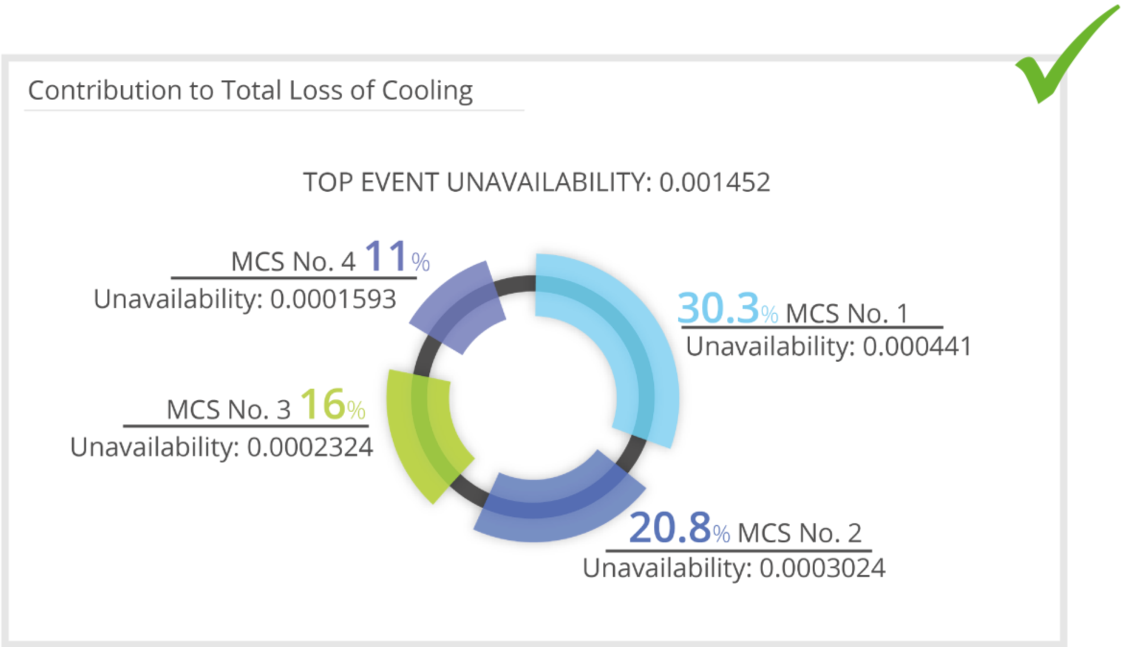
When clicking on an event it shows all the MCS with this event, like this >

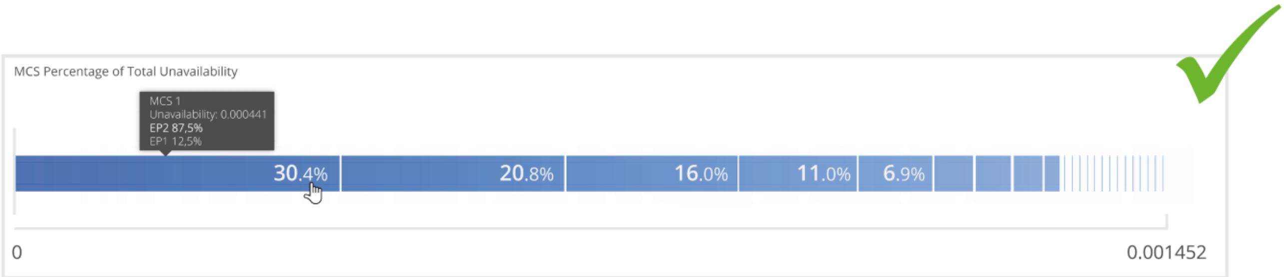


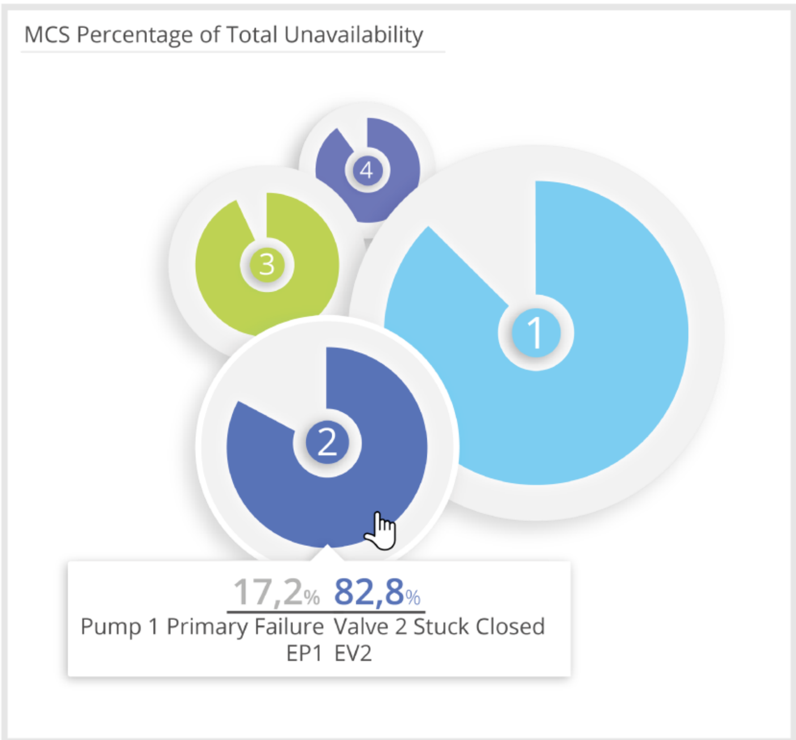


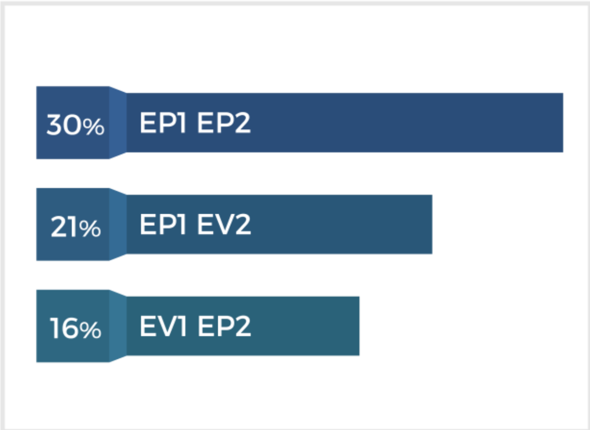


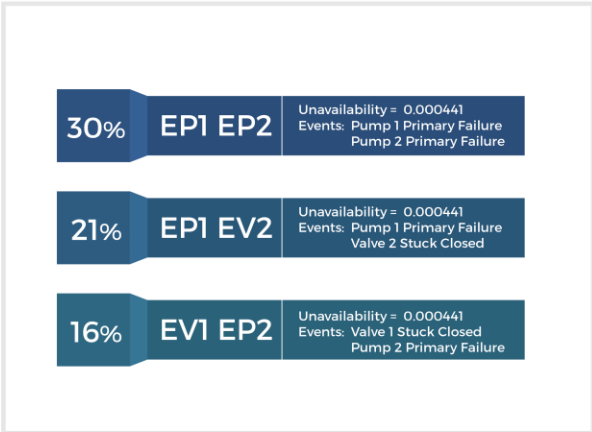
The size of the round is the unavailability of the minimal cut set, the bigger the surface, the higher the unavailability.


Contribution to Total Loss of Cooling
TOP EVENT UNAVAILABILITY: 0.001452
MCS No. 4 11%
Unavailability: 0.0001593
30.3% MCS No. 1
Unavailability: 0.000441
MCS No. 3 16%
Unavailability: 0.0002324
20.8% MCS No. 2
Unavailability: 0.0003024

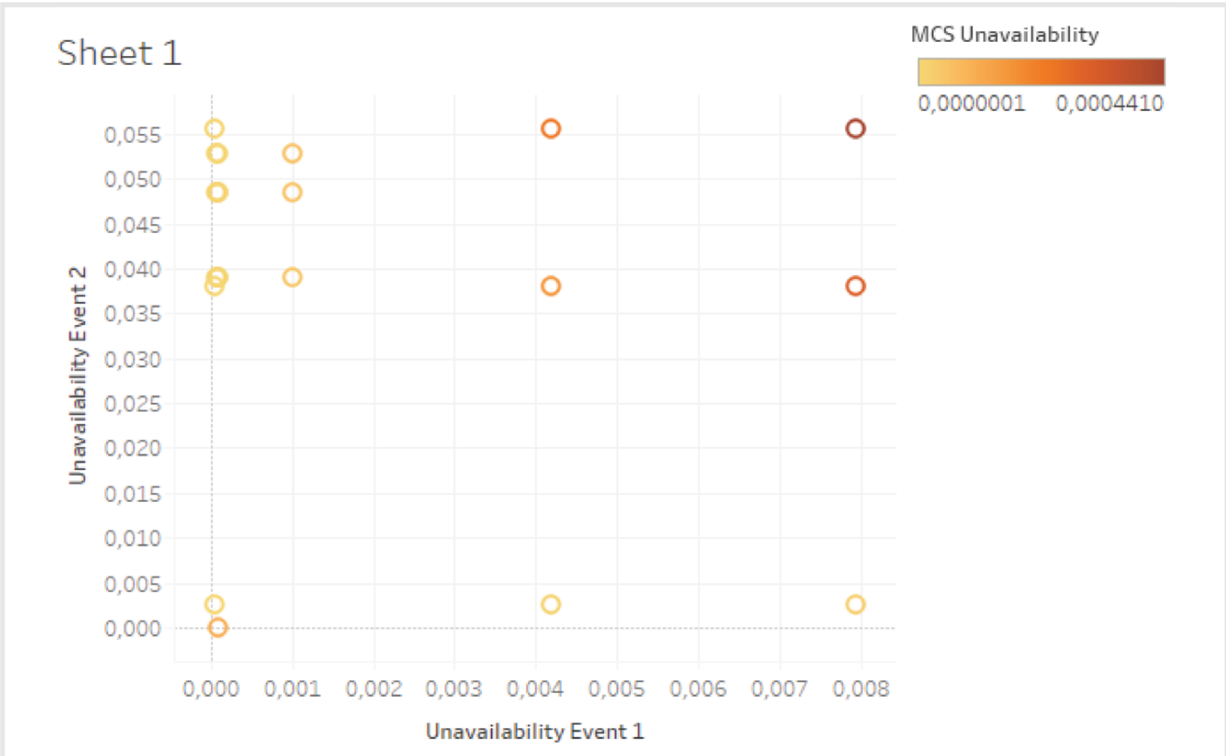

MCS Percentage of Total Unavailability
MCS 1
Unavailability: 0.000441
EP2 87,5%
EP1 12,5%
30.4%
20.8%
16.0%
11.0%
6.9%
0
0.001452

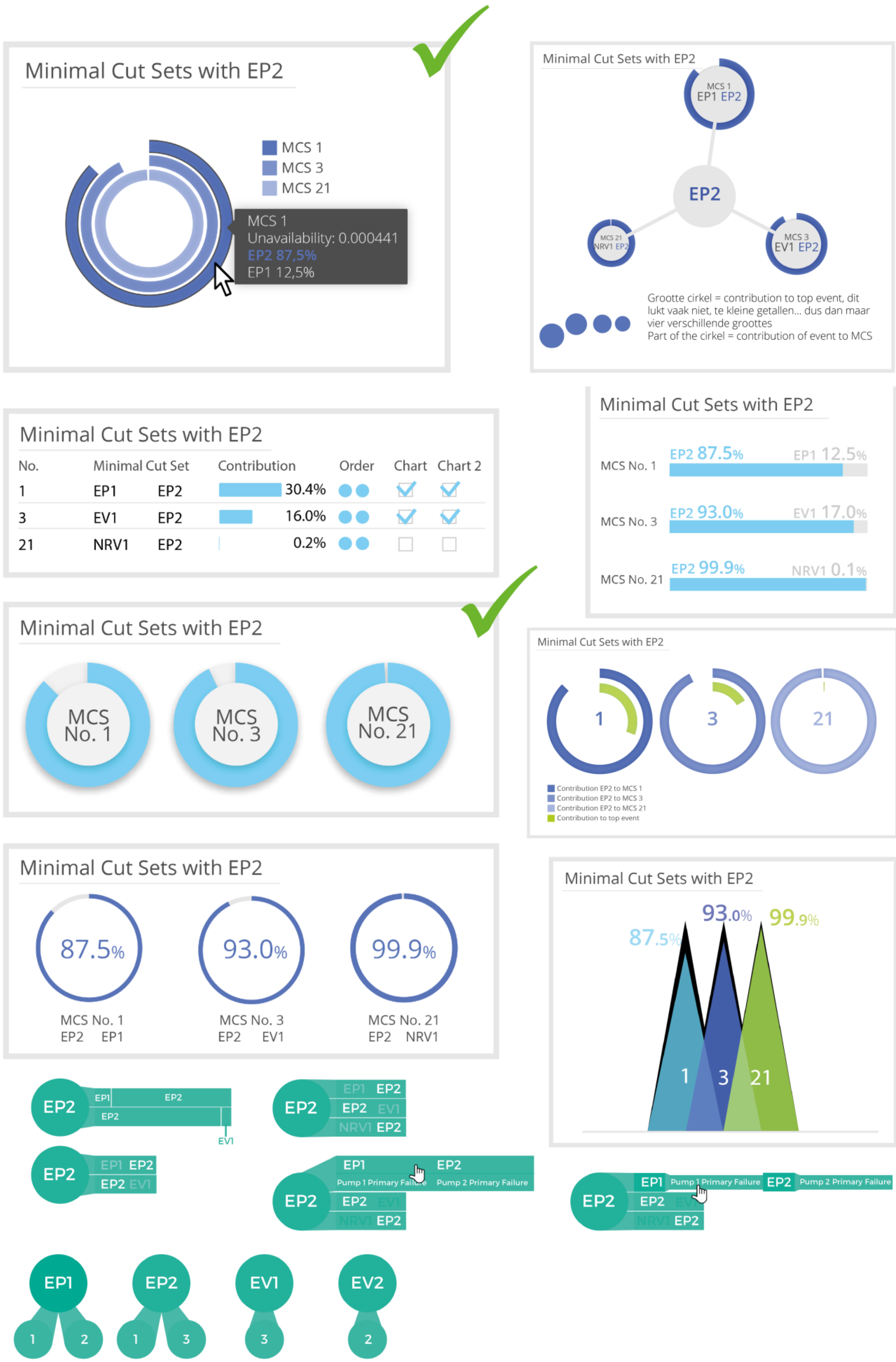

MCS Percentage of Total Unavailability
4
3
1
2
17,2% 82,8%
Pump 1 Primary Failure Valve 2 Stuck Closed
EP1 EV2





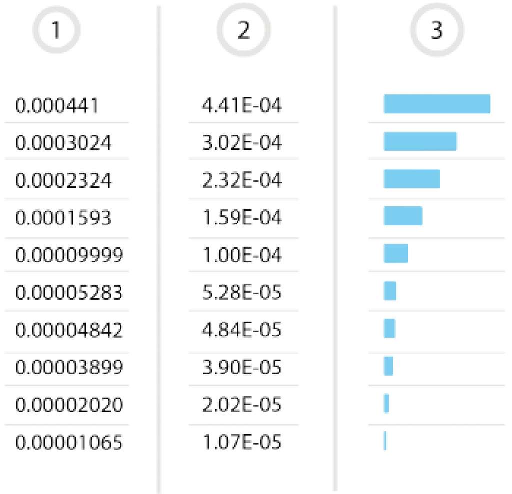
Each bar represents a minimal cut set, the percentage is the contribution to the top level event's probability.




Minimal Cut Sets with EP2
MCS 1
MCS 3
MCS 21
MCS 1
Unavailability: 0.000441
EP2 87,5%
EP1 12,5%
Minimal Cut Sets with EP2
MCS 1
EP1 EP2
EP2
MCS 21
NRV1 EP2
MCS 3
EV1 EP2
Grootte cirkel = contribution to top event, dit lukt vaak niet, te kleine getallen... dus dan maar vier verschillende grootes
Part of the cirkel = contribution of event to MCS
Minimal Cut Sets with EP2
No. Minimal Cut Set Contribution Order Chart Chart 2
1 EP1 EP2 30.4%
3 EV1 EP2 16.0%
21 NRV1 EP2 0.2%
Minimal Cut Sets with EP2
MCS No. 1 EP2 87.5% EP1 12.5%
MCS No. 3 EP2 93.0% EV1 17.0%
MCS No. 21 EP2 99.9% NRV1 0.1%
Minimal Cut Sets with EP2
MCS No. 1
MCS No. 3
MCS No. 21
Minimal Cut Sets with EP2
1
3
21
Contribution EP2 to MCS 1
Contribution EP2 to MCS 3
Contribution EP2 to MCS 21
Contribution to top event
Minimal Cut Sets with EP2
87.5%
93.0%
99.9%
MCS No. 1
EP2 EP1
MCS No. 3
EP2 EV1
MCS No. 21
EP2 NRV1
Minimal Cut Sets with EP2
87.5%
93.0%
99.9%
1
3
21
EP2
EP1
EP2
EP2
EV1
EP2
EP1 EP2
EP2 EV1
NRV1 EP2
EP2
EP1 EP2
EP2 EV1
EP2
EP1
EP2
Pump 1 Primary Failure
Pump 2 Primary Failure
EP2 EV1
NRV1 EP2
EP2
EP1
Pump 1 Primary Failure
EP2
Pump 2 Primary Failure
EP2 EV1
NRV1 EP2
EP1
1
2
EP2
1
3
EV1
3
EV2
2

8.3 Appendix C: Survey for Comparing Numbers

Look at the image below and try to figure out which row represents the lowest of them all and the highest. Do this for each column (1,2,3)



Which column did you find easier for finding the lowest and the highest?

- ❍ 1
- ❍ 2
- ❍ 3

Which column do you find the most appealing?

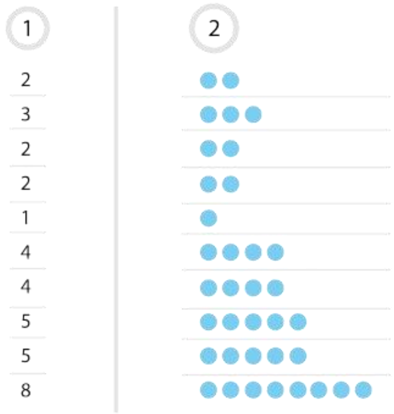
- ❍ 1
- ❍ 2
- ❍ 3

Results

Column Number	**Easier**	**Most Appealing**
1	0	0
2	2	2
3	19	19

The participants consisted out of extremely, slightly and not familiar people with minimal cut sets.

Look at the image below and try to figure out which row represents the lowest of them all and the highest. Do this for each column (1,2)



Which column did you find easier for finding the lowest and the highest?

❍ 1
❍ 2

Which column do you find the most appealing?

❍ 1
❍ 2

Results

Column Number	**Easier**	**Most Appealing**
1	2	3
2	19	18

The participants consisted out of extremely, slightly and not familiar people with minimal cut sets.

8.4 Appendix D: Coding

8.4.1 Index2.html

```
<!DOCTYPE html>
<html lang="en">

<head>

    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="">
    <meta name="author" content="">

    <title>DataViz</title>

    <!-- Bootstrap Core CSS -->
    <link href="../vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
    <!-- MetisMenu CSS -->
    <link href="../vendor/metisMenu/metisMenu.min.css" rel="stylesheet">
    <!-- Custom CSS -->
    <link href="../dist/css/sb-admin-2.css" rel="stylesheet">
    <!-- Morris Charts CSS -->
    <link href="../vendor/morrisjs/morris.css" rel="stylesheet">
    <!--Import D3, jQuery and floatthead-->
    <script src="http://d3js.org/d3.v3.min.js"></script>
    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/floatthead/2.0.3/jquery.floatThead.min.js"></scrip
t>

    <!-- Import javascript functions-->
    <script src="Chart1.js"></script>
    <script src="Chart2.js"></script>
    <script src="Chart3.js"></script>
    <script src="circles_order.js"></script>
    <script src="bluehighlights.js"></script>

    <!--Import Open Sans font from google fonts-->
    <link href="https://fonts.googleapis.com/css?family=Open+Sans:400,300" rel="stylesheet">
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
    <link rel="stylesheet" href="index.css">

</head>

<body>

    <div id="wrapper">
        <!-- Navigation -->
        <nav class="navbar navbar-default navbar-static-top" role="navigation" style="margin-
bottom: 0">
            <!-- Top Menu -->
            <div class="navbar-header">
                <a class="navbar-brand" href="index2.html">MCS Visualization Tool</a>
            </div>

            <ul class="nav navbar-top-links navbar-right">
                <li class="dropdown">
                    <a class="dropdown-toggle" data-toggle="dropdown" href="#">
                        <i class="fa fa-user fa-fw"></i> <i class="fa fa-caret-down"></i>
                    </a>
                    <ul class="dropdown-menu dropdown-user">
                        <li><a href=""><i class="fa fa-user fa-fw"></i> User Profile</a>
                        </li>
                        <li><a href=""><i class="fa fa-gear fa-fw"></i> Settings</a>
                        </li>
                    </ul>
                </li>
            </ul>

            <!-- Left Menu -->
            <div class="navbar-default sidebar" role="navigation">
                <div class="sidebar-nav navbar-collapse">
                    <ul class="nav" id="side-menu">
```

```
                        <li>
                            <a href="" class="active">
                            <img src="faulttree.png" alt="Icon Chart 2" height="15"
width="15">  Fault Tree 1</a>
                        </li>
                        <li>
                            <a href="">
                            <img src="faulttree.png" alt="Icon Chart 2" height="15"
width="15">  Fault Tree 2</a>
                        </li>
                        <li>
                            <a href="">
                            <i class="fa fa-upload fa-fw"></i> Upload Fault Tree</a>
                        </li>
                    </ul>
                </div>
            </div>
        </nav>

        <!-- Content Layout -->
        <div id="page-wrapper">
            <div class="row">
                <div class="col-lg-12">
                    <h1 class="page-header">Fault Tree 1</h1>
                </div>
            </div>

            <!-- Table -->
            <div class="row">
                <div class="col-lg-8">
                    <div class="panel panel-default">
                        <div class="panel-heading">
                            <i class="fa fa-table fa-fw"></i>
                            <text class= "TitelTopEvent"> </text>
                            <div class="pull-right">
                                <div class="btn-group">
                                    <button type="button" class="btn btn-default btn-xs
dropdown-toggle" data-toggle="dropdown">
                                        Actions
                                        <span class="caret"></span>
                                    </button>
                                    <ul class="dropdown-menu pull-right" role="menu">
                                        <li><a href="index2.html">Refresh</a>
                                        </li>
                                    </ul>
                                </div>
                            </div>
                        </div>
                        <div class="panel-body">
                            <div class="Tabel">
                                <div class="wrapper" style="height:300px;  overflow-y:auto">
                                </div>
                            </div>
                        </div>
                    </div>

                    <div class="panel ">
                        <div class="row">

                            <!-- Chart 2 -->
                           <div class="col-lg-6">
                                <div class="panel panel-default box2">
                                    <div class="panel-heading">
                                        <img src="iconChart2-2.png" alt="Icon Chart 2"
height="20" width="20">   Chart 2  -  Contribution of MCS to Top Event
                                        <div class="popup"
onclick="myFunction(myPopup2,icontabel2)"><i  id="icontabel2"class="material-
icons">&#xe88f;</i>
                                            <span class="popuptext2" id="myPopup2">
                                                Info Chart 2: Contribution of MCS to Top Event
                                                <br> By clicking on a checkbox underneath
Chart 2, the corresponding minimal cut set will be displayed in this chart. The percentage is
the contribution of the minimal cut set to the top event's unavailability. This is calculated
as P(MCS)/P(TLE)*100%.
                                                <br> The color of the arc represents the
unavailability of the minimal cut set. The darker and more purple the color, the higher the
unavailability of the minimal cut set.
```

```
                                        </span>
                                    </div>
                                </div>
                                <div class="panel-body" id="idChart2" >
                                <svg class = "Chart2"></svg>
                                </div>
                            </div>
                        </div>

                        <!-- Chart 3 -->
                        <div class="col-lg-6">
                            <div class="panel panel-default box3">
                                <div class="panel-heading">
                                    <img src="iconChart3.png" alt="Icon Chart 3"
height="20" width="20">    Minimal Cut Sets with  <span class="eventheader"
id="eventheader" style="font-weight: 800">No Event Selected</span>
                                    <div class="popup"
onclick="myFunction(myPopup3,icontabel3)"><i  id="icontabel3"class="material-
icons">&#xe88f;</i>
                                        <span class="popuptext3" id="myPopup3">
                                            Info Chart 3: Minimal Cut sets with Event A
                                            <br> When an event is selected from the table
it will be visualized in this chart. For each minimal cut set this event appears in, an arc
will be drawn. The length of the arc is determined by the probability of this event compared
to the other events in the minimal cut set. Namely, the probability of Event A divided by the
sum of all the events' probabilities in that minimal cut set.
                                            <br> The color of the arc represents the
unavailability of the minimal cut set. The bluer the color, the higher the unavailability of
the minimal cut set.
                                        </span>
                                    </div>
                                </div>
                                <div class="panel-body" id="idChart3""style=" display:
flex; flex-direction: row; flex-wrap: wrap; justify-content: center; align-items: center;">
                                    <svg class="Chart3"></svg>
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
            </div>

            <!-- Chart 2 -->
            <div class="col-lg-4">
                <div class="panel panel-default box1">
                    <div class="panel-heading">
                        <img src="iconChart1.png" alt="Icon Chart 2" height="20"
width="20">   Chart 1  -  Extra Info MCS
                            <div class="popup"
onclick="myFunction(myPopup1,icontabel1)"><i  id="icontabel1"class="material-
icons">&#xe88f;</i>
                                <span class="popuptext" id="myPopup1">
                                    Info Chart 1: Extra info MCS
                                    <br> By clicking on a checkbox underneath Chart 1, the
corresponding minimal cut set will be displayed in this chart. Each event in this minimal cut
set is visualized as a part of the donut. The bluer and bigger the arc, the higher the
probability of that event.
                                </span>
                            </div>
                    </div>
                    <div class="panel-body">
                        <div class = "tekstnodig" style="height:750px; overflow-y:auto; ">
                            <svg class = "Chart1" style="height=350px; width:400px">
</svg>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>

    <!-- jQuery -->
    <script src="../vendor/jquery/jquery.min.js"></script>
    <!-- Bootstrap Core JavaScript -->
    <script src="../vendor/bootstrap/js/bootstrap.min.js"></script>
```

```
    <!-- Metis Menu Plugin JavaScript -->
    <script src="../vendor/metisMenu/metisMenu.min.js"></script>
    <!-- Custom Theme JavaScript -->
    <script src="../dist/js/sb-admin-2.js"></script>

    <script>

    // Add text to chart 1.
    // Give it a height otherwise the text will not be displayed.
    var canvas1 = d3.select(".Chart1")
        .attr("height",350);
        canvas1.append("text")
        .text("No Minimal Cut Set Selected")
        .attr("transform",
"translate("+($(".tekstnodig").width())/2+","+$(".Tabel").height()/2+")")
        .attr("class", "none1");
    // Add svg for chart 2.
    var canvas2 = d3.select(".Chart2")
        .attr("width", $("#idChart2").width())
        .attr("height",350);
        canvas2.append("text")
        .attr("transform",
"translate("+($(".Chart2").width())/2+","+$(".Chart2").height()/2+")")
        .text("No Minimal Cut Set Selected")
        .attr("class", "none2");
    // Add svg for chart 3s
    var canvas3 = d3.select(".Chart3")
        .attr("width", $("#idChart3").width())
        .attr("height",350);
        canvas3.append("text")
        .text("No Event Selected")
        .attr("id", "none3")

.attr("transform","translate("+($(".Chart3").width())/2+","+$(".Chart3").height()/2+")");

    // When the user clicks on the infobuttion, open the popup.
    function myFunction(popuptext,icon) {
        popuptext.classList.toggle("show");
        icon.classList.toggle("active");
    // If the page has been zoomed in or out, it will display it according to the size of the
block if loaded again.
        var width = $(".tekstnodig").width();
        d3.select(popuptext).style("width", width+"px");
    }

    // Each header needs a span for info.
    for( var q = 0; q<7; q++){
        d3.select('body').append("span")
            .attr("class", "table_info"+q + " table_info" );
    }

    var tabulate = function (mcsdata,eventdata,topeventdata, columns) {
        // Add title above table.
        var titel = d3.select(".TitelTopEvent").text(topeventdata[0].TLEName);

        // Add a table, header and body element.
        var table = d3.select('.wrapper').append('table')
        table.attr("class", "table table-hover dataTable no-footer dtr-inline");
        var titles = table.append('thead')
        var tbody = table.append('tbody')

        var total = topeventdata[0].TotalUnavailability;
        var color = "#42d1f4"; //blue color
        var checkedLines2 =[];
        var prev_checkedLines2 = [];
        // Determine the largest cut set by calculating the amount of headers minus the
unavailability of the mcs.
        var max_amount_events = d3.keys(mcsdata[0]).length-1;

        // Append a row to the theader and give it headers/titles.
        // For the minimal cut set columns it should have a span of based on the amount of max
events in a mcs.
        // Give each of the headers a class so when hovering on it, it displays different text
for each header.
        // Give each span a class to add info text to span.
        titles.append('tr')
            .selectAll('th')
```

```
            .data(["MCS #", "Minimal Cut Set", "Contribution", "Order", "Chart1","Chart2"])
            .enter()
            .append('th')
            .attr("colspan", function (d, i) {
                if (i == 1) return max_amount_events;
            })
            .append("text")
            .attr("class", function(d,i){
                return "table_header"+i +" table_header";
            })
            .attr("id", function(d,i){
                return "table_header"+i;
            })
            .text(function (d) {
                return d;
            })
            .on("mouseover", function(d,i){
                d3.select(".table_info"+i)
                .style("top",$(this).offset().top+"px")
                .style("left", $(this).offset().left+$(this).width()+10+"px")
                .style("visibility","visible");
            })
            .on("mouseout", function(d,i){
                d3.select(".table_info"+i).style("left","70px").style("visibility","hidden");
            })

        // Add chart-icons to the headers.
        table_header1.innerHTML = "<img src='iconChart3.png' alt='Icon Info Chart 3' height =
'15' width='15'/> Minimal Cut Set "  ;
        table_header4.innerHTML = "<img src='iconChart1.png' alt='Icon Info Chart 1' height =
'15' width='15'/> Chart 1 "  ;
        table_header5.innerHTML = "<img src='iconChart2-2.png' alt='Icon Info Chart 2' height
= '15' width='15'/> Chart 2 "  ;

        // Create rows.
        var rows = tbody.selectAll('tr')
            .data(mcsdata)
            .enter()
            .append('tr')

        // Create cells and add numbers and events.
        var x = -1;
        var rownumber=0;
        var cells = rows.selectAll('td')
            .data(function (row) {
                return columns.map(function (column) {
                    return {column: column, value: row[column]}
                })
            })
            .enter()
            .append('td')
            .text(function (d, i) {
                if(i==0){
                    rownumber++;
                    return rownumber;
                }
                else {
                    return d.value;
                }

            })
            .attr("class", function (d, i) {
                if(i>0 && i <=max_amount_events && d.value.length>0){
                    return "underline-text"
                }
                if (i == max_amount_events+1) {
                    return "bar"
                }
                if (i == max_amount_events+2) {
                    x++;
                    return "order" + x
                }
                if (i == max_amount_events+3) {
                    return "chart1";
                }
                if (i == max_amount_events+4) {
                    return "chart2";
```

```
            }
        });

    // Give highlight effects to the event titles.
    // On mouseclick draw chart 3
    var specifiedevent;
    rows.selectAll(".underline-text")
        .attr("id", function(d,i){return d.value})
        .on("click",function(d,i){
            // First, remove all the events that are blue by changing the class.
            $('.event-selected').attr("class","underline-text");
            // When clicking on the same event, the chart will be removed.
            if(specifiedevent == d.value){
                d3.selectAll("g.chart3").remove();
                specifiedevent = 0;
                eventheader.innerText = "No Event Selected";
                none3.innerHTML = "No Event Selected";
            }else{
                // Select all events with the same id name and color them by changing the
class.
                d3.selectAll("#"+d.value).attr("class",'event-selected');
                // d.value Is the name of the event the user clicks on.
                specifiedevent = d.value;
                d3.selectAll("g.chart3").remove();
                d3.selectAll(".tooltip3").remove();
                chart3(mcsdata, eventdata, specifiedevent,max_amount_events);
            }
        })
        // When hovering on an event all of those events will be blue.
        // When hovering on an event the border of chart 3 will be blue.
        .on("mouseover", function(d,i){
            d3.select(".box3").style("border-color",color);
            d3.select(".box3").select(".panel-heading").style("background-color",color);
            d3.selectAll("#"+d.value).style("color","#42d1f4");
        })
        // Turn the colors back on mouseout.
        .on("mouseout", function(d,i){
            d3.select(".box3").style("border-color","#e6e6e6");
            d3.select(".box3").select(".panel-heading").style("background-
color","#f5f5f5");
            d3.selectAll("#"+d.value).style("color","black");
        })



    // Add bars to the table the length is based on the mcs unavailability.
    rows.selectAll("td.bar")
        .data(function (row) {
            return columns.map(function (column) {
                return {column: column, value: row[column]}
            })
        })
        .append("svg")
        .attr("width", "100")
        .attr("height", "20")
        .append("rect")
        .attr("width", function (d, i) {
            return d.value * 200000;
        })
        .attr("height", "13")
        .attr("rx", 2)
        .attr("ry", 2)
        .attr("transform","translate(0,5)")
        .attr("fill", color);

    // Add circles to table based on the amount of events.
    // For each row determine the amount of events.
    for (var j = 0; j < mcsdata.length; j++) {
        var eventnames = [];
        for (var x = 1; x < max_amount_events+1; x++) {
            var String = "mcsdata[" + j + "].Event" + x;
            var eventname = eval(String);
            if (eventname.length > 0) {
                eventnames.push(eventname);
            };
        }
        // For each row make circles based on the amount of events.
```

```
            circles(j, eventnames.length,rows,color);
        }

        // Maak voor alle data een input element aan van type checkbox.
        // Als value krijgen ze een rij nummer.
        for(var x = 1; x<3; x++){
            var label = rows.selectAll("td.chart"+x)
                .data(function (d, i) {
                    return [i];
                })
                .append("label")
                .attr("class", "regular-checkbox")

                label.append("input")
                .attr("type", "checkBox")
                .attr("class", "regular-checkbox"+x)
                .attr("value", function (d) {
                    return d;
                });
            label.append("span");
        }

        // Check if a checkbox1 changes, each time execute chart1().
        d3.selectAll("input.regular-checkbox1").on("change", function () {
            var checkedLines1 = [];
            // Check every checkbox if they are checked.
            d3.selectAll("input.regular-checkbox1").each(function (d) {
                if (d3.select(this).node().checked == true) {
                    checkedLines1.push(d3.select(this).attr("value"));
                }
            });
            d3.selectAll("g.single").remove();
            d3.selectAll(".tooltip1").remove();
            chart1(mcsdata, eventdata, checkedLines1, max_amount_events);
        });

        // Check if a checkbox2 changes, each time execute chart1().
        d3.selectAll("input.regular-checkbox2").on("change", function () {
            prev_checkedLines2 = checkedLines2;
            checkedLines2 = [];

            //  Check every checkbox if they are checked.
            d3.selectAll("input.regular-checkbox2").each(function (d) {
                if (d3.select(this).node().checked == true) {
                    checkedLines2.push(d3.select(this).attr("value"));
                }
            });
                d3.selectAll("g.total").remove();
                if(checkedLines2.length>0){
                    chart2(mcsdata,checkedLines2,total,topeventdata, prev_checkedLines2);
                }
                if(checkedLines2.length==0){
                    canvas2.append("text")
                    .attr("transform",
"translate("+$(".Chart2").width()/2+","+$(".Chart2").height()/2+")")
                    .text("No Minimal Cut Set Selected")
                    .attr("class", "none2")
                }

        });

        // Highlighting chart 1 and 2 when hovering on checkbox or header.
        bluehighlights(color);

        // Add info text to each of the headers of the table.
        $(".table_info0").html("<text class= infotext> MCS #: stands for the minimal cut set
number, each minimal cut set has been given a number based on their unavailability. The
minimal cut set with the highest unavailability has been given number one.</text>");
        $(".table_info1").html("<text class= infotext> Minimal Cut Set: in the columns
underneath, all the event titles are listed that occur in the minimal cut sets. Each row
defines one individual minimal cut set. On each event title can be clicked, this will affect
the chart shown in 3. Minimal Cut Sets with … More information on Chart 3 here.</text>");
        $(".table_info2").html("<text class= infotext> Contribution: this column indicates the
same number as percent but now it is represented by a bar.</text>");
        $(".table_info3").html("<text class= infotext> Order: each circle indicates an event
in the minimal cut set; if there are three circles this means that the minimal cut set is of
order three and thus containing three events.</text>");
```

```
        $(".table_info4").html("<text class= infotext> Chart 1: when clicking on a checkbox
below, a chart of the minimal cut set in the same row will appear in the corresponding 'view
box'. Chart 1 corresponds to 1. Extra Info MCS. More information on Chart 1 here.</text>");
        $(".table_info5").html("<text class= infotext> Chart 2: when clicking on a checkbox
below, a chart of the minimal cut set in the same row will appear in the corresponding 'view
box'. Chart 2 corresponds to 2. Contribution of MCS to Top Event. More information on Chart 2
here.</text>");

        return table;

    }

    // Load the data and draw EVERYTHING.
    d3.csv('mydataMCS2.csv', function (mcsdata) {
        d3.csv('mydataEvents.csv', function (eventdata) {
            d3.csv('mydataTLE.csv', function (topeventdata) {
                var columns = d3.keys(mcsdata[0]);
                columns.push("Contribution", "Order", "Chart1","Chart2");
                tabulate(mcsdata, eventdata, topeventdata, columns)
            })
        })
    })


    <!-- $(document).ready(function () {  -->
      <!-- var $table = $('.wrapper');  -->

         <!-- $table.floatThead({  -->
            <!-- // useAbsolutePositioning: true,  -->
             <!-- scrollContainer: function ($table) {  -->
                 <!-- return $table.closest('.wrapper');  -->
            <!-- }  -->
         <!-- });  -->
     <!-- });  -->

    // Dropshadow with adjustable color and opacity.
    // Code from https://stackoverflow.com/questions/12277776/how-to-add-drop-shadow-to-d3-js-
pie-or-donut-chart
    var defs = d3.select("body").append("defs")
    var defs = canvas2.append("defs")

    var filter = defs.append("filter")
        .attr("id", "dropshadow")
    filter.append("feGaussianBlur")
        .attr("in", "SourceAlpha")
        .attr("stdDeviation", 4)
        .attr("result", "blur");
    filter.append("feOffset")
        .attr("in", "blur")
        .attr("dx", 0)
        .attr("dy", 0)
        .attr("result", "offsetBlur")
    filter.append("feFlood")
        .attr("in", "offsetBlur")
        .attr("flood-color", "black")
        .attr("flood-opacity", "0.5")
        .attr("result", "offsetColor");
    filter.append("feComposite")
        .attr("in", "offsetColor")
        .attr("in2", "offsetBlur")
        .attr("operator", "in")
        .attr("result", "offsetBlur");

    var feMerge = filter.append("feMerge");
    feMerge.append("feMergeNode")
        .attr("in", "offsetBlur")
    feMerge.append("feMergeNode")
        .attr("in", "SourceGraphic");

</script>
</body>
</html>
```

8.4.2 Chart 1.js

```
// Function that draws a pie charts of minimal cut sets.
var chart1 = function (mcsdata, eventdata, hoeveelsteMCS, max_amount_events) {

    var inRadius = 50;
    var outRadius = 75;
    var legendRectSize = outRadius / 7;
    var legendSpacing = 30;
    var group_height = 20;
    var tooltip = d3.select('body')
        .append('div')
        .attr('class', 'tooltip1');
    tooltip.append('div')
    .attr('class', 'eventname');
    tooltip.append('div')
    .attr('class', 'unavailability');

    // For each minimal cut set make a pie chart.
    for (var j = 0; j < hoeveelsteMCS.length; j++) {

        var eventnames = [];
        var eventpropabilities = [];
        var events = [];

        var group = canvas1.append("g")
            .attr("class", "single")

        // Add background shadow.
        group.append("circle")
            .attr("r", outRadius)
            .attr("fill", "white")
            .attr("filter", "url(#dropshadow)");

        // Add the events and unavailability to the arrays.
        for (var x = 1; x <= max_amount_events; x++) {
            var string1 = "mcsdata[" + hoeveelsteMCS[j] + "].Event" + x;
            var eventname = eval(string1);
            if (eventname.length > 0) { //only add something to the array if there is
something
                eventnames.push(eventname);
                var string_eventprop = "eventdata[1]." + eventnames[x-1];
                var eventprop = eval(string_eventprop);
                eventpropabilities.push(eventprop);
            }
        }

        // Make an array of objects, each object is an event.
        for (var x = 0; x < eventnames.length; x++) {
            var string2 = "eventdata[1]." + eventnames[x];
            var string3 = "eventdata[0]." + eventnames[x];
            events.push({
                "eventname": eventnames[x],
                "prop": eval(string2),
                "description": eval(string3)
            })
        }

        var total = d3.sum(eventpropabilities);

        // Sorting based on unavailability.
        events.sort(function (a, b) {
            return b.prop - a.prop
        });
        eventpropabilities.sort(function (a, b) {
            return b - a
        });

        // Color scale from blue to grey.
        var color = d3.scale.linear()
            .domain([events[0].prop , events[events.length-1].prop])
            //.interpolate(d3.interpolateHcl)
            .range(["#42d1f4", "#ddd"])
            //.range(["#ff6347", "#46e3ff"]);
            //.range(["#6959ff",'#59ffbf']); //same colors as chart2

        var arc = d3.svg.arc()
            .innerRadius(inRadius)
            .outerRadius(outRadius);
```

```
        var pie = d3.layout.pie()
            .value(function (d, i) {
                return events[i].prop;
            })

        // Select all the arcs to bind it to the data.
        var arcs = group.selectAll(".arc")
            .data(pie(events))
            .enter()
            .append("g")
            .attr("class", "arc");

        // The pie(data) must be binded with the paths.
        arcs.append("path")
            .attr("d", arc)
            .attr("fill", function (d, i) {
                return color(d.value);
            })

        // When hover on arc give it a stroke and show tooltip.
        // Display tooltip on mouse position.
        arcs.on("mouseover", function (d,i) {
            d3.select(this).style("stroke", "#4d4d4d");
            tooltip.select('.eventname').html(d.data.eventname+": "+d.data.description);
            tooltip.select('.unavailability').html("Unavailability: " +
d3.format("e")(d.value));
            tooltip.style('display', 'block');
        })
        .on('mousemove', function (d) {
            var event = d3.mouse(d3.select('body').node());
            tooltip.style('top', (event[1] + 10) + 'px').style('left', (event[0] + 10) +
'px');
        })
        .on("mouseout", function (d) {
            d3.select(this).style("stroke", "none");
            tooltip.style('display', 'none');
        })

        // Add a white circle on top with shadow.
        // Add MCS # and unavailability as word and numerical.
        group.append("circle")
        .attr("r", inRadius)
        .attr("fill", "#fcfcfc")
        .attr("filter", "url(#dropshadow)")

        group.append("text")
        .text(function (d, i) {
            return "MCS #" + (eval(hoeveelsteMCS[j]) + 1)
        })
        .attr("text-anchor", "middle")
        .attr("transform", "translate(0," + (-outRadius - 20) + ")")
        .attr("font-size", 16);

        group.append("text")
        .text("Unavailability")
        .attr("text-anchor", "middle")
        .attr("font-size", 13);

        group.append("text")
        .text(function (d, i) {
            return d3.format("e")(mcsdata[hoeveelsteMCS[j]].MCSUnavailability)
        })
        .attr("text-anchor", "middle")
        .attr("transform", "translate(0,20)")
        .attr("font-size", 20)

        // If there are more than 3 events, add 20 pixels for each.
        var tmp_outRadius = outRadius;
        if (events.length >= 3) {
            tmp_outRadius += (((events.length) - 3) * 14);
        }

        // Add line between charts.
        group.append("line")
        .attr("x1", -outRadius)
        .attr("y1", tmp_outRadius + 20)
```

```
        .attr("x2", 250)
        .attr("y2", tmp_outRadius + 20)
        .attr("stroke", "#e6e6e6")
        .attr("stroke-width", 1)
        .attr("fill", "none");

        // Put all the elements of the legend in one group.
        var legend_g = group.append("g").attr("class", "scrolbaarmaken");

        // Create a legend for each unavailability.
        // Add percentage, event title, event description and a line.
        var legend = legend_g.selectAll('.legend')
            .data(eventpropabilities)
            .enter()
            .append('g')
            .attr('transform', function (d, i) {
                var height = legendRectSize + legendSpacing;
                var offset = height * (eventpropabilities.length - 1) / 2;
                var horz = outRadius + legendRectSize;
                var vert = i * height - offset;
                return 'translate(' + horz + ',' + vert + ')';
            });
        legend.append('text')
        .text(function (d, i) {
            return d3.format(".1%")(eventpropabilities[i] / total)
        })
        .attr("fill", color)
        .attr("class", "percentage");

        legend.append('text')
        .attr('x', "50")
        .text(function (d, i) {
            return events[i].eventname
        })
        .attr("class", "eventtitle");

        legend.append("text")
        .attr('y', "14")
        .text(function (d, i) {
            return events[i].description
        })
        .attr("class", "description");

        legend.append("line")
        .attr("x1", 0)
        .attr("y1", 2)
        .attr("x2", function (d, i) {
            return 75;
        })
        .attr("y2", 2)
        .attr("stroke", "black")
        .attr("stroke-width", 0.5)
        .attr("fill", "none");

        // Give the div element a height so that it is scrollable.
        group.each(function () {
            group.attr("transform", "translate(" + ($(".tekstnodig").width() / 2 -
this.getBBox().width / 2 + outRadius) + "," + (this.getBBox().height / 2 + group_height) +
")")
            group_height += this.getBBox().height + 30;
        });
    }

    // Change the size attribute according to how many cut sets are displayed.
    var chart_size = canvas1.attr("height",  group_height);

    // delete 'No Minimal Cut Set Selected'
    d3.selectAll("text.none1").remove();

    // Add 'No Minimal Cut Set Selected' if there is nothing to display.
    if (hoeveelsteMCS.length == 0) {
        canvas1.attr("height", 350)
        .append("text")
        .text("No Minimal Cut Set Selected")
        .attr("transform", "translate(" + ($(".tekstnodig").width()) / 2 + "," +
$(".Tabel").height() / 2 + ")")
        .attr("class", "none1");
```

```
    }
}
```

8.4.3 Chart 2.js

```
// Function that draws arcs based on the amount of cut sets.
var chart2 = function (mcsdata, whatMCS,total,topeventdata,prev_checkedLines2) {

    var outrad = 80;
    var radverschil = 12;
    var numberofMCS = mcsdata.length;
    var unavailability = [];
    var pi = Math.PI * 2;
    var arcs = whatMCS.length;
    var tussen_lengte = 1;
    var height = 35;
    var min_allowed = 0.055;
    var tooltip = d3.select('#idChart2')
        .append('div')
        .attr('class', 'tooltip2');
    tooltip.append('div')
    .attr('class', 'mcsnumber');
    tooltip.append('div')
    .attr('class', 'unavailability');

    // Make an array with the all the MCS unavailability.
    mcsdata.map(function (d) {
        unavailability.push(+d.MCSUnavailability);
    })

    // Function that calculates the percentage of MCS to TLE unavailability.
    var percent = function (d, i) {
        return mcsdata[i].MCSUnavailability / total;
    }

    var group = canvas2.append("g")
        .attr("class", "total")

    // Make a black arc.
    var blackarc = d3.svg.arc()
        .innerRadius(outrad - radverschil * 0.9)
        .outerRadius(outrad)
        .startAngle(0)
        .endAngle(Math.PI * 2);

    // Color scale, each minimal cut set gets its own color.
    var color = d3.scale.linear()
        .domain([ mcsdata[mcsdata.length-1].MCSUnavailability, mcsdata[0].MCSUnavailability])
       // .interpolate(d3.interpolateHsl)
       // .range(["#b7ff32","#452cd4"]);
       // .range(["#32ff70","#452cd4"]); //purple to green
        .range(["#42d1f4","#452cd4"]);  //purple to blue

    var build_arcs = function () {

        var arc_lengte = 0;

        // Calculate the space available between each arc.
        for (var x = 0; x < arcs; x++) {
            tussen_lengte = tussen_lengte - percent(0, whatMCS[x]);
        }

        tussen_lengte = (tussen_lengte / arcs);

        // Draw the arcs.
        for (var x = 0; x < arcs; x++) {
            // The first arc is placed at startAngle 0.
            // The rest gets a space.
            if (x == 0) {
                var arc = d3.svg.arc()
                    .innerRadius(outrad - radverschil * 2)
                    .outerRadius(outrad + radverschil)
                    .startAngle(0)
                    .endAngle(function (d, i) {
```

```
                        return (percent(0, whatMCS[x])) * pi;
                    });
            } else {
                var arc = d3.svg.arc()
                    .innerRadius(outrad - radverschil * 2)
                    .outerRadius(outrad + radverschil)
                    .startAngle(function (d, i) {
                        return (arc_lengte + (tussen_lengte * (x))) * pi;
                    })
                    .endAngle(function (d, i) {
                        return ((arc_lengte + (tussen_lengte * (x)) + percent(0, whatMCS[x]))
* pi)
                    ;
                    });
            }

            // Only if there is a new arc it gets a class for animation.
            group.append("path")
            .attr("fill", color(mcsdata[whatMCS[x]].MCSUnavailability))
            .attr("filter", "url(#dropshadow)")
            .style("opacity", 0.9)
            .attr('class', function(d){
                if(x==arcs-1 && prev_checkedLines2.length < arcs){
                    return 'arc_chart2 new_arc arc'+x
                }else{
                    return 'arc_chart2 arc'+x;
                }
            })
            .attr("d", arc);

            // Initialize a boolean to check if there is enough room to display the legend
based on the centre of the arc.
            var min_tussenruimte = tussen_lengte + percent(0, whatMCS[(arcs - 1)]);
            var tussenruimte_boolean = min_tussenruimte > min_allowed && arcs < 11;
            var tussenruimte_boolean = min_tussenruimte > min_allowed ;
            var tussenruimte_boolean = arcs < 11;
            var offset = outrad;

            // Make a group legend.
            // If there is a new arc only give that legend a new_arc class to animate it.
            // Add the percentage, MCS#, a line and unavailability to each legend.
            var legend = group.append('g')
            .attr('class', function(d){
                if(x==arcs-1 && prev_checkedLines2.length < arcs){
                    return 'legend_chart2 new_legend legend'+x
                }else{
                    return 'legend_chart2 legend'+x
                }
            })

            legend.append("text")
            .attr("class", "percentage")
            .attr("transform", function (d) {
                if (Boolean(tussenruimte_boolean)) {
                    return "translate(" + ((arc.centroid(d)[0]) * 1.7) + "," +
((arc.centroid(d)[1]) * 1.7) + ")";
                } else {

                    var horz = outrad+75;
                    var vert = x * height - offset;
                    return 'translate(' + horz + ',' + vert + ')';
                };
            })
            .attr("text-anchor", function(d){ if (Boolean(tussenruimte_boolean)) { return
"middle"} else{ return "end" }})
            .text(function (d, i) {
                if((d3.format(".1f")(percent(0, whatMCS[x]) * 100))==0){
                    return d3.format(".2f")(percent(0, whatMCS[x]) * 100) + "%";
                } else{
                return d3.format(".1f")(percent(0, whatMCS[x]) * 100) + "%";
                }
            })
            .attr("fill", color(mcsdata[whatMCS[x]].MCSUnavailability))

            legend.append("text")
            .attr("text-anchor", function(d){ if (Boolean(tussenruimte_boolean)) { return
"middle"}})
```

```
            .attr("class", "eventtitle")
            .attr("transform", function (d) {
                if (Boolean(tussenruimte_boolean)) {
                    if (arc.centroid(d)[0] > 0) {
                        return "translate(" + ((arc.centroid(d)[0]) * 1.7 + 50) + "," +
((arc.centroid(d)[1]) * 1.7) + ")";
                    } else {
                        return "translate(" + ((arc.centroid(d)[0]) * 1.7 - 50) + "," +
((arc.centroid(d)[1]) * 1.7) + ")";
                    }
                } else {

                    var horz = outrad +30+ 50;
                    var vert = x * height - offset;
                    return 'translate(' + horz + ',' + vert + ')';
                };
            })
            .text(function (d, i) {
                return "MCS #" + (eval(whatMCS[x]) + 1);
            })

            legend.append("text")
            .attr("text-anchor", function(d){ if (arc.centroid(d)[0] < 0 &&
Boolean(tussenruimte_boolean)) { return "end"}})
            .attr("class", "description")
            .attr("transform", function (d) {
                if (Boolean(tussenruimte_boolean)) {
                    if (arc.centroid(d)[0] > 0) {
                        return  "translate(" + ((arc.centroid(d)[0]) * 1.7-22) + "," +
((arc.centroid(d)[1]) * 1.7+14) + ")";
                    } else {
                        return "translate(" + ((arc.centroid(d)[0]) * 1.7 +22) + "," +
((arc.centroid(d)[1]) * 1.7+14) + ")";
                    }
                } else {

                    var horz = outrad + 30;
                    var vert = x * height - offset+14;
                    return 'translate(' + horz + ',' + vert + ')';
                };
            })
            .text(function (d, i) {
                return "Unavailability: " +
(d3.format('e')(eval(mcsdata[whatMCS[x]].MCSUnavailability)));
            })

            legend.append("line")
            .attr("x1", function (d, i) {
                if (Boolean(tussenruimte_boolean)) {
                    if (arc.centroid(d)[0] < 0) {
                        return arc.centroid(d)[0] * 1.7 + 22;
                    } else {
                        return arc.centroid(d)[0] * 1.7 - 22;
                    }
                } else {
                    return outrad + 30;
                }
            })
            .attr("y1", function (d) {
                if (Boolean(tussenruimte_boolean)) {
                    return (arc.centroid(d)[1]) * 1.7 + 2;
                } else {
                    return x * height - offset + 2;
                }
            })
            .attr("x2", function (d, i) {
                if (Boolean(tussenruimte_boolean)) {
                    if (arc.centroid(d)[0] < 0) {
                        return arc.centroid(d)[0] * 1.7 - 75;
                    } else {
                        return arc.centroid(d)[0] * 1.7 + 75;
                    }
                } else {
                    return outrad + 135;
                }
            })
            .attr("y2", function (d) {
```

```
                if (Boolean(tussenruimte_boolean)) {
                    return (arc.centroid(d)[1]) * 1.7 + 2;
                } else {
                    return x * height - offset + 2;
                }
            })
            .attr("stroke", "black")
            .attr("stroke-width", 0.5)
            .attr("fill", "none");

            // Update the current length of the arcs that are already drawn.
            arc_lengte = arc_lengte + percent(0, whatMCS[x]);

        }

        // Add top event name and unavailability.
        if(Boolean(tussenruimte_boolean) == false){
            group.append("text")
                .text(topeventdata[0].TLEName)
                .attr("class", "chart2_title")
                .attr("text-anchor","middle")
                .attr("transform", "translate(0,"+(-outrad-50)+")");
            group.append("text")
                .text("Unavailability: "+d3.format("e")(topeventdata[0].TotalUnavailability))
                .attr("class", "chart2_unav")
                .attr("text-anchor","middle")
                .attr("transform", "translate(0,"+(-outrad-35)+")");
            group.attr("transform", "translate("+ (outrad+($(".Chart2").width()-
$(".total").width())/2)+ ","+ (outrad +85) +")");
        }else{
            group.append("text").text(topeventdata[0].TLEName).attr("class",
"chart2_title").attr("text-anchor","middle").attr("transform", "translate(0,"+(-outrad*1.7-
45)+")");
            group.append("text").text("Unavailability:
"+d3.format("e")(topeventdata[0].TotalUnavailability)).attr("class",
"chart2_unav").attr("text-anchor","middle").attr("transform", "translate(0,"+(-outrad*1.7-
30)+")");
            group.attr("transform", "translate(" + $(".Chart2").width() / 2 + "," +
($(".Chart2").height() / 2 +30) + ")");
        }
    }
    //Add the black arc.
    group.append("path")
    .attr("d", blackarc)

    // Execute the function defined above.
    build_arcs();

    // Remove the 'No Minimal Cut Set Selected'
    d3.selectAll("text.none2").remove();
    // Add the 'No Minimal Cut Set Selected' if there is nothing to display.
    if( $(".total").height() > 350){
        d3.select(".Chart2").attr("height", function(){
         return $(".total").height()+20;
        })
    }else{
        d3.select(".Chart2").attr("height", 350);
    }

    // Add strokes and change opacity of legends and arcs on mouseover.
    d3.selectAll('.legend_chart2')
        .on('mouseover', function(d,i){
            d3.selectAll('.arc_chart2').style('opacity', 0.85);
            d3.select('.arc'+i).style('opacity', 0.95).style("stroke", "black");
            d3.selectAll('.legend_chart2').selectAll('text').style('opacity',0.4 );
            d3.select('.legend'+i).selectAll('text').style('opacity',1);
        })
        .on('mouseout', function(d,i){
            d3.selectAll('.arc_chart2').style('opacity', 0.9).style("stroke", "none");
            d3.selectAll('.legend_chart2').selectAll('text').style('opacity',1 );
        })

    d3.selectAll('.arc_chart2')
        .on('mouseover',function(d,i){
            d3.selectAll('.arc_chart2').style('opacity', 0.85);
            d3.select(this).style('opacity', 0.95).style("stroke", "black");
            d3.selectAll('.legend_chart2').selectAll('text').style('opacity',0.4 );
```

```
            d3.select('.legend'+i).selectAll('text').style('opacity',1);

        })
        .on('mouseout',function(d){
            d3.selectAll('.arc_chart2').style('opacity', 0.9).style("stroke", "none");
            d3.selectAll('.legend_chart2').selectAll('text').style('opacity',1 );
        })  ;
}
```

8.4.4 Chart 3.js

```
// Function that draw multiple arcs based on how often an event occurs in minimal cut sets.
var chart3 = function (mcsdata, eventdata, specifiedevent, max_amount_events) {

    // Change the header of this block by adding the event name.
    eventheader.innerText = specifiedevent;
    // Display nothing ('No Event Selected').
    none3.innerHTML = "";

    var outRadius = 100;
    var pi = Math.PI * 2;
    var legendRectSize = 16;
    var legendSpacing = 2;
    var howmanyarcs = [];

    var tooltip = d3.select('#idChart3')
        .append('div')
        .attr('class', 'tooltip3');
    tooltip.append('div')
    .attr('class', 'mcsnumber');
    tooltip.append('div')
    .attr('class', 'unavailability');

    // Check for each row, thus MCS, if the event is in there.
    // If so, add it to the array.
    for (var x = 0; x < mcsdata.length; x++) {
        if (Object.values(mcsdata[x]).indexOf(specifiedevent) > -1) {
            howmanyarcs.push(x);
        }
    }

    // Color scale from blue to grey.
    var color = d3.scale.linear()
        .domain([0, howmanyarcs.length-1])
        //.interpolate(d3.interpolateHcl)
        .range(["#42d1f4", "#ddd"]);
        //.range(["#452cd4", "#e6e6e6"]); // purple to grey

    // Determine the width of each arc based on the amount of arcs.
    if (howmanyarcs.length <= 4) {
        var arc_width = 18;
    } else if (howmanyarcs.length > 4 && howmanyarcs.length < 15) {
        var arc_width = 75 / howmanyarcs.length;
    } else {
        var arc_width = 75 / 14;
        outRadius = 100 + arc_width * (howmanyarcs.length - 14)
    }

    var group = canvas3.append("g")
        .attr("class", "chart3")

        // For each minimal cut set draw an arc.
        for (var x = 0; x < howmanyarcs.length; x++) {

            // Put the events of the minimal cut set in the array.
            var eventnames = [];
            for (var j = 1; j <= max_amount_events; j++) {
                var eventname = eval("mcsdata[" + howmanyarcs[x] + "].Event" + j);
                if (eventname.length > 0) {
                    eventnames.push(eventname);
                }
            }

            // Add the unavailability for each event.
            var eventprobabilities = [];
            for (var k = 0; k < eventnames.length; k++) {
```

```
                var String = "eventdata[1]." + eventnames[k];
                var eventprop = eval(String);
                eventprobabilities.push(eventprop);
            }

            var total = d3.sum(eventprobabilities);
            var spec_event_prob = eval("eventdata[1]." + specifiedevent);

            // Define each arc, the length is the probability of the event compared to all of
them.
            var arc = d3.svg.arc()
                .innerRadius(outRadius - arc_width + 2 - arc_width * x)
                .outerRadius(outRadius - arc_width * x)
                .startAngle(0)
                .endAngle((spec_event_prob / total) * pi);

            group.append("path")
            .attr("fill", color(x))
            .attr("d", arc)
            .attr("class", "arc_chart3");
            console.log('color of x is '+x+color(x));
        }

        // Show the number of in how many mcs the event is present.
        group.append("text")
        .text(function (d) {
            return howmanyarcs.length
        })
        .attr("class", "bignumber");

    // Show the title and description and unavailability of the event.
    group.append("text")
    .text(eval("eventdata[0]." + specifiedevent))
    .attr("class", "chart3_title")
    .attr("transform", "translate(" + -outRadius + "," + (-outRadius - 45) + ")")
    group.append("text")
    .text("Unavailability: " + d3.format("e")(eval("eventdata[1]." + specifiedevent)))
    .attr("class", "chart3_unav")
    .attr("transform", "translate(" + -outRadius + "," + (-outRadius - 30) + ")")

    var maketooltip = d3.select(".chart3").selectAll("path")
        .on("mouseover", function (d, i) {
            d3.select(this).style("stroke", "#4d4d4d"); // #00ffd2 blueish color
            var probs = [];

            // Load all the eventnames for the arc the mouse is on
            for (var n = 1; n <= max_amount_events; n++) {
                var eventname2 = eval("mcsdata[" + howmanyarcs[i] + "].Event" + n);
                if (eventname2.length > 0) {
                    var eventprop2 = eval("eventdata[1]." + eventname2);
                    probs.push(eventprop2);
                }
            }

            var totalprob = d3.sum(probs);

            // Add a div for each event in the minimal cut set.
            // Add the MSC # and unavailability.
            // Add the event title and percentage to each div.
            // The specified event gets the color of the arc.
            for (var m = 1; m <= max_amount_events; m++) {
                d3.selectAll(".eventname" + m).remove();

                var eventname2 = eval("mcsdata[" + howmanyarcs[i] + "].Event" + m);

                tooltip.append('div')
                .attr('class', "eventname" + m);

                if (eventname2 == specifiedevent) {
                    d3.select(".eventname" + m)
                    .style("color", color(i))
                    .attr('id', "specifiedevent");
                }

                if (eventname2.length > 0) {
                    var eventprop2 = eval("eventdata[1]." + eventname2);
```

```
                    tooltip.select(".eventname" + m).html(eventname2 + "  " +
d3.format(".1%")(eventprop2 / totalprob));
                } else {
                    tooltip.select(".eventname" + m).html("");
                }
            }

            tooltip.select('.mcsnumber').html("MCS #" + (howmanyarcs[i] + 1) + "
Unavailability: " + d3.format("e")(mcsdata[howmanyarcs[i]].MCSUnavailability));
            tooltip.style('display', 'block');

        })
        .on('mousemove', function (d) {
            tooltip.style('top', (d3.event.layerY + 10) + 'px')
            .style('left', (d3.event.layerX + 10) + 'px');
        })
        .on("mouseout", function (d) {
            d3.select(this).style("stroke", "none");
            tooltip.style('display', 'none'); ;
        })

    // Add a legend with rectangle and MCS#.
    var legend = group.selectAll('.legend_chart3')
    .data(howmanyarcs)
    .enter()
    .append('g')
    .attr('class', 'legend_chart3')
    .attr('transform', function (d, i) {
        var height = legendRectSize + legendSpacing;
        var horz = outRadius + legendRectSize;
        var vert = i * height - outRadius;
        return 'translate(' + horz + ',' + vert + ')';
    });

    legend.append('rect')
    .attr('width', legendRectSize)
    .attr('height', legendRectSize)
    .attr('fill', function (d, i) {
        return color(i)
    });

    legend.append("text")
    .attr('x', legendRectSize + legendSpacing)
    .attr('y', legendRectSize - legendSpacing)
    .text(function (d, i) {
        return "MCS #" + (howmanyarcs[i] + 1)
    })

    // A white arc with all values bound except the endAngle.
    // SVG path string for a given angle, we pass an object with an endAngle
    // property to the `arc` function, and it will return the corresponding string.
    var wittearc = d3.svg.arc()
        .innerRadius(outRadius - arc_width * howmanyarcs.length)
        .outerRadius(outRadius + 10)
        .startAngle(pi);

    var foreground1 = group.append("path")
        .datum({
            endAngle: pi
        })
        .style("fill", "white")
        .attr("d", wittearc);

    foreground1.transition()
    .duration(2000)
    .attrTween("d", function (d) {
        var interpolate = d3.interpolate(0, d.endAngle);
        return function (t) {
            d.endAngle = interpolate(t);
            return wittearc(d);
        }
    });

    // Place the group in the middle of the svg.
    group.attr("transform", "translate(" + (outRadius + ($(".Chart3").width() -
$(".chart3").width()) / 2) + "," + (outRadius + 67) + ")");
```

```
    // Change the height of the div .Header3 according to the legend and arc size.
    var border_size = d3.select(".Chart3")
        .style("height", function () {
            if ($(".chart3").height() > 350) {
                return ($(".chart3").height() + 50) + "px";
            }

        })

    // Replace the text in the middle.
    // Do this at the end, otherwise the placement of the total group would be shitty.
    d3.select('.chart3_title').attr("text-anchor", "middle")
    .attr("transform", "translate(0," + (-outRadius - 45) + ")");
    d3.select('.chart3_unav').attr("text-anchor", "middle")
    .attr("transform", "translate(0," + (-outRadius - 30) + ")");

}
```

8.4.5 Circles_order.js

```
// Function that draws circles based on length.
var circles = function (j, length, rows, color) {
    for (var x = 0; x < length; x++) {
        rows.select("td.order" + j)
        .append("svg")
        .attr("width", "14")
        .attr("height", "20")
        .append("circle", circle);

        var circle = d3.selectAll("circle")
            .attr("r", 5)
            .attr("transform", "translate(5,12)")
            .attr("fill", color);
    }
}
```

8.4.6 Bluehighlights.js

```
// Function that highlights the borders of chart 1 and 2 on mousemove.
var bluehighlights = function (color) {

    // When the mouse is over all checkboxes or header, highlight the border of chart 1.
    d3.selectAll(".chart1").selectAll(".regular-checkbox").on("mouseover", function () {
        d3.select(".box1").style("border-color", color);
        d3.select(".box1").select(".panel-heading").style("background-color", color);
    })
    d3.selectAll(".chart1").selectAll(".regular-checkbox").on("mouseout", function () {
        d3.select(".box1").style("border-color", "#e6e6e6");
        d3.select(".box1").select(".panel-heading").style("background-color", "#f5f5f5");
    })

    d3.select(".table_header4").on("mouseover", function () {
        d3.select(".box1").style("border-color", color);
        d3.select(".box1").select(".panel-heading").style("background-color", color);
        d3.select(".table_info4")
        .style("top", $(this).offset().top + "px")
        .style("left", $(this).offset().left + $(this).width() + 10 + "px")
        .style("visibility", "visible");
    })
    d3.select(".table_header4").on("mouseout", function () {
        d3.select(".box1").style("border-color", "#e6e6e6");
        d3.select(".box1").select(".panel-heading").style("background-color", "#f5f5f5");
        d3.select(".table_info4").style("left", "70px").style("visibility", "hidden");
    })

    // When the mouse is over all checkboxes or header, highlight the border of chart 2.
    d3.selectAll(".chart2").selectAll(".regular-checkbox").on("mouseover", function () {
        d3.select(".box2").style("border-color", color);
        d3.select(".box2").select(".panel-heading").style("background-color", color);
    })
    d3.selectAll(".chart2").selectAll(".regular-checkbox").on("mouseout", function () {
        d3.select(".box2").style("border-color", "#e6e6e6");
        d3.select(".box2").select(".panel-heading").style("background-color", "#f5f5f5");
    })
```

```
    d3.select(".table_header5").on("mouseover", function () {
        d3.select(".box2").style("border-color", color);
        d3.select(".box2").select(".panel-heading").style("background-color", color);
        d3.select(".table_info5")
        .style("top", $(this).offset().top + "px")
        .style("left", $(this).offset().left + $(this).width() + 10 + "px")
        .style("visibility", "visible");
    })
    d3.select(".table_header5").on("mouseout", function () {
        d3.select(".box2").style("border-color", "#e6e6e6");
        d3.select(".box2").select(".panel-heading").style("background-color", "#f5f5f5");
        d3.select(".table_info5").style("left", "70px").style("visibility", "hidden");
    })

}
```

8.5 Appendix E: Evaluation Survey

Welcome to this survey. Thank you for participating in this survey. The survey aims to evaluate the data visualization tool that has been developed for a graduation project. The goal of the data visualization tool is to visualize fault tree analysis, more specifically the minimal cut sets. This survey takes about 15 to 20 minutes to complete. Be assured that all answers you provide will be kept in confidentiality. Please click >>

Page break

What is your gender?
- ❍ Male
- ❍ Female

What is your age?

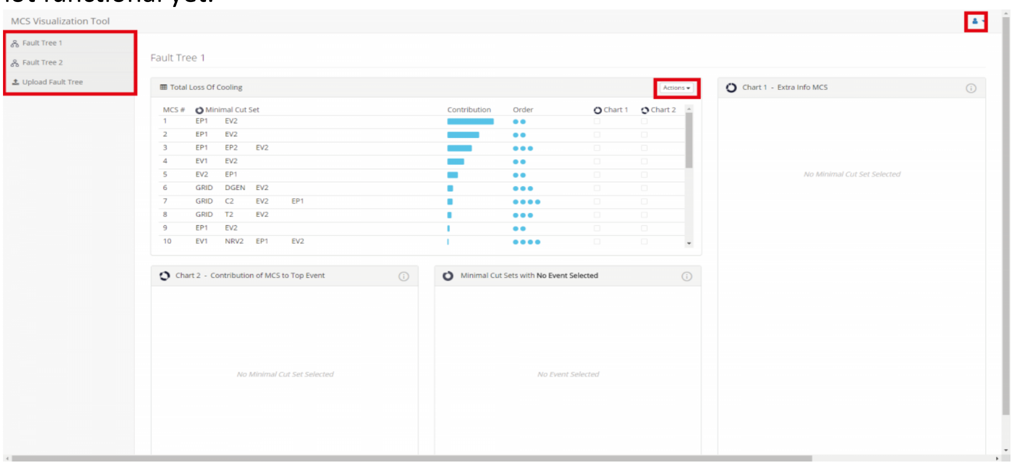
What is your profession?

How experienced are you with fault tree analysis?
- ❍ Not Experienced
- ❍ Slightly Experienced
- ❍ Moderately Experienced
- ❍ Extremely Experienced

How familiar are you with minimal cut sets?
- ❍ Not familiar
- ❍ Slightly familiar
- ❍ Moderately familiar
- ❍ Extremely familiar

Page break

You will be asked to do three tasks, for each task there are questions to answer. These tasks can only be accomplished by using the data visualization, the data visualization can be accesed by clicking here. Please open the link. The link should display the same as in the picture below. NOTE: Parts in red are not functional yet.



Task 1: Get a visualization in block 'Extra Info MCS' for minimal cut set 11.Find out what event in minimal cut set 11 has the highest unavailability.

What is the title of this event?
- EV2
- DGEN4
- DGEN
- C1
- I am not sure

What is the unavailability of this event?
- 5.288e-6
- 5.288e-2
- 3.18e-2
- 5e-3
- 9.99e-5
- I am not sure

What is the event's description?
- Valve 2 Stuck Closed
- Diesel Generator Failure 4
- Contact Breaker 1 Failure
- Diesel Generator Failure
- I am not sure

How easy was it to get to these answers?
- Easy
- Slightly easy
- Neither easy nor difficult
- Slightly difficult
- Difficult

What is the main reason that you gave the previous answer?

Page break

Did you use the tooltip?
- Yes
- No

The following statements are about the chart in block 'Extra Info MCS'For each of the statements below, select the answer that resembles your opinion the most.

	Strongly disagree	Disagree	Somewhat disagree	Neither agree nor disagree	Somewhat agree	Agree	Strongly agree
I find this chart appealing.	❍	❍	❍	❍	❍	❍	❍
I understand what is visualized in this chart.	❍	❍	❍	❍	❍	❍	❍
The chart showed too much information.	❍	❍	❍	❍	❍	❍	❍
The tooltip is useful.	❍	❍	❍	❍	❍	❍	❍

Do you have any suggestions for improving this chart?

Page break

Task 2: Get a visualization visible in block 'Contribution of MCS to Top Event'. Select MCS# 1, 2, and 3 visible in this chart.

What percentage do the first three minimal cut sets cover?
- ❍ less than 20%
- ❍ between 20% and 40%
- ❍ between 40% and 60%
- ❍ between 60% and 80%
- ❍ more than 80%

How easy was it to get to this answer?
- ❍ Easy
- ❍ Slightly easy
- ❍ Neither easy nor difficult
- ❍ Slightly difficult
- ❍ Difficult

What is the main reason that you gave the previous answer?

The following statements are about the chart in block 'Contribution of MCS to Top Event'For each of the statements below, select the answer that resembles your opinion the most.

	Strongly disagree	Disagree	Somewhat disagree	Neither agree nor disagree	Somewhat agree	Agree	Strongly agree
I find this chart appealing.	❍	❍	❍	❍	❍	❍	❍
I understand what is visualized in this chart.	❍	❍	❍	❍	❍	❍	❍
The chart showed too much information.	❍	❍	❍	❍	❍	❍	❍

Do you have any suggestions for improving this chart?

Page break

Task 3: Find out in how many cut sets the event EP2 appears in. Thus, select an event so that it appears in 'Minimal Cut Sets with No Event Selected'.

The event EP2 appears in how many cut sets?

How easy was it to get to this answer?

- ❍ Easy
- ❍ Slightly easy
- ❍ Neither easy nor difficult
- ❍ Slightly difficult
- ❍ Difficult

What is the main reason that you gave the previous answer?

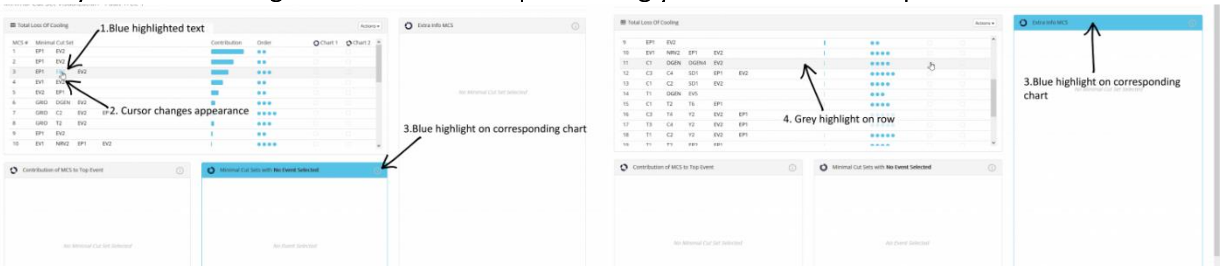
The following statements are about the chart in block 'Minimal Cut Sets with ...'For each of the statements below, select the answer that resembles your opinion the most.

	Strongly disagree	Disagree	Somewhat disagree	Neither agree nor disagree	Somewhat agree	Agree	Strongly agree
I find this chart appealing.	○	○	○	○	○	○	○
I understand what is visualized in this chart.	○	○	○	○	○	○	○
The chart showed too much information.	○	○	○	○	○	○	○
The tooltip is useful.	○	○	○	○	○	○	○

Do you have any suggestions for improving this chart?

Page break

Did any of the following visual elements help in finding your answers to the previous tasks?



Please select the elements that you found to be useful during the tasks.

❑ 1. Blue highlighted text
❑ 2. Cursor changes appearance
❑ 3. Blue highlights on charts
❑ 4. Grey highlight on row

Were there other elements of this tool that you found to be useful for finding the answers of the previous tasks? If yes, please describe these elements.

Page break

This is the last part of this survey. You can use and interact with the data visualization tool while answering the questions on this page.

For each of the statements below, select the answer that resembles your opinion the most.

	Strongly disagree	Disagree	Somewhat disagree	Neither agree nor disagree	Somewhat agree	Agree	Strongly agree
I find this tool attractive as a whole.	❍	❍	❍	❍	❍	❍	❍
While using this tool, I was overloaded with information.	❍	❍	❍	❍	❍	❍	❍
While using this tool, I had the feeling that there was too much information on the page.	❍	❍	❍	❍	❍	❍	❍
I understand how to use this tool.	❍	❍	❍	❍	❍	❍	❍
The tool did what I expected it to do.	❍	❍	❍	❍	❍	❍	❍
I had the feeling that I knew what I was doing while using the tool.	❍	❍	❍	❍	❍	❍	❍
I had to put in a lot of effort to use this tool.	❍	❍	❍	❍	❍	❍	❍

Page break

Have you used any kind of tool for fault tree analysis before?

❍ Yes

❍ No > *Go to the last question*

Are you familiar with different tools for fault tree analysis?

❍ Yes

❍ No

What is the name of this tool for fault tree analysis?If you are not free to answer, go to the next question.

What are the names of these tools for fault tree analysis? If you are not free to answer, go to the next question.
You can use the tool and interact with the data visualization tool while answering the following questions.
How appealing is this visualization of minimal cut sets compared to the visualization of minimal cut sets in the tool(s) you have used?

- ❍ Extremely appealing
- ❍ Very appealing
- ❍ Somewhat appealing
- ❍ Not so appealing
- ❍ Not at all appealing

I find this data visualization tool more useful for presenting minimal cut sets compared to the presentation of minimal cut sets in the tool(s) I am familiar with.

- ❍ Strongly agree
- ❍ Agree
- ❍ Somewhat agree
- ❍ Neither agree nor disagree
- ❍ Somewhat disagree
- ❍ Disagree
- ❍ Strongly disagree

This tool shows me more interesting information of minimal cut sets than tools I am familiar with.

- ❍ Strongly agree
- ❍ Agree
- ❍ Somewhat agree
- ❍ Neither agree nor disagree
- ❍ Somewhat disagree
- ❍ Disagree
- ❍ Strongly disagree

Do you have any suggestions for improving this data visualization for minimal cut sets? Multiple answers are welcome.

References

[1] W. Vesely, F. Goldberg, N. Roberts and D. Haasl, Fault Tree Handbook, Washington, D.C.: Systems and Reliability Research, Office of Nuclear Regulatory Research, U.S. Nuclear Regulatory Commission, 1981.

[2] M. Statematelatos and W. Vesely, Fault Tree Handbook with Aerospace Applications, 1st ed., Washington, D.C.: Office of safety and mission assurance NASA headquarters, 2002.

[3] C. f. C. P. Safety, Guidelines for Hazard Evaluation Procedures, 3rd Edition, John Wiley & Sons, 2008.

[4] C. Eckberg, WS-133B Fault Tree Analysis Program Plan, 1st ed., Ft. Belvoir: Defense Technical Information Center, 1964.

[5] W. Lee, D. Grosh, F. Tillman and C. Lie, "Fault Tree Analysis, Methods, and Applications - A Review," *IEEE Transactions on Reliability,* vol. 34, no. 3, pp. 194-203, 1986.

[6] M. Stoelinga, "Fault Tree Analysis - an introduction".

[7] "Fault Tree Analysis Diagrams Solution | ConceptDraw.com," [Online]. Available: http://www.conceptdraw.com/solution-park/engineering-fault-tree-analysis-diagrams.

[8] "Fault Tree Diagram Software, Create Fault Tree Diagrams Rapidly with Examples and Templates.," [Online]. Available: https://www.edrawsoft.com/fault-tree-diagram-software.php.

[9] J. Fussel, "Fault Trees - A State of the Art Discussion," *IEEE Trans. Reliab.,* vol. 4, no. 2, pp. 51-55, 1974.

[10] B. Fischhoff, "Fault trees: Sensitivity of estimated failure probabilities to problem representation.," *Journal of Experimental Psychology: Human Perception and Performance,* vol. 4, no. 2, pp. 330-344, 1978.

[11] Z. Nivolianitou, "Comparison of techiques for accident scenario analysis in harzardous systems," *Journal of Loss Prevention in the Process Industries,* vol. 17, no. 6, pp. 467-475, 2004.

[12] R. E. Barlow and F. Proschan, "Importance of System Components and Fault Tree Events," *Stochastic Processes and their Applications,* vol. 3, pp. 53-173, 1975.

[13] S. Beeson and J. D. Andrews, "Importance Measures for Non-Coherent-System," *IEEE Transactions on reliability,* vol. 52, no. 3, 2003.

[14] E. Ruijters and M. Stoelinga, "Fault tree analysis: A survey of the state-of-the-art in modeling, analysis and tools," *Computer Science Review,* Vols. 15-16, pp. 29-62, 2015.

[15] A. Engineering, "Fault-tree-analysis-software.com," [Online]. Available: http://www.fault-tree-analysis-software.com/fault-tree-analysis?type=Aerospace. [Accessed 2017].

[16] "rams-reliability-availability-maintainability-and-safety-software," 23 3 2017. [Online]. Available: http://aldservice.com/reliability-products/rams-software.html.

[17] "Fault Tree Analysis - Isograph," 14 3 2017. [Online]. Available: https://www.isograph.com/software/reliability-workbench/fault-tree-analysis/?gclid=CInR8K66utICFYY-GwodmxUJ2Q .

[18] "Fault Tree Analysis Diagrams Solution | ConceptDraw.com," 3 2017. [Online]. Available: http://www.conceptdraw.com/solution-park/engineering-fault-tree-analysis-diagrams#!howto.

[19] 23 3 2017. [Online]. Available: http://www.itemsoft.com/fault_tree.html?gclid=CO6Ku_yt7NICFRATGwodqEsC7Q.

[20] 23 3 2017. [Online]. Available: http://www.itemsoft.com/about_us.html.

[21] "EPRI Public Site," 3 2017. [Online]. Available: https://www.epri.com/#/pages/product/000000003002000020/EPRI Public Site.

[22] CAFTA 6.0 Demo User Manual.

[23] R. Publishing, "Reliasoft.com," [Online]. Available: http://www.reliasoft.com/synthesis/downloads.htm. [Accessed 3 4 2017].

[24] R. Publishing, "Reliasoft.com," [Online]. Available: http://www.reliasoft.com/products.htm. [Accessed 3 4 2017].

[25] Jeffrey A. Julius, *WinNUPRA USER'S MANUAL,* Washington: Scientech LLC, 2006, p. 299.

[26] J. Wiley, Guidelines for Chemical Process Quantitative Risk Analysis, 2nd ed., 1999.

[27] E. N. Mckay, UI is Communication: How to Design Intuitive, User Centered Interfaces by Focusing on Effective Communication, 1st ed., Morgan Kaufmann Publishers, 2013.

[28] W. C. Eells, "The Relative Merits of Circles and Bars for Representing Component Parts," *American Statistical Association,* vol. 21, no. 154, pp. 119-132, 1926.

[29] M. Bostock, "D3.js - Data-Driven Documents," [Online]. Available: https://d3js.org/. [Accessed 30 April 2017].

[30] "Start Bootstrap," [Online]. Available: https://startbootstrap.com/. [Accessed June 2017].

[31] "SB Admin 2 - Free Bootstrap Admin Theme," [Online]. Available: https://startbootstrap.com/template-overviews/sb-admin-2/. [Accessed June 2017].

[32] "How to add drop shadow to d3.js pie or donut chart," [Online]. Available: https://stackoverflow.com/questions/12277776/how-to-add-drop-shadow-to-d3-js-pie-or-donut-chart. [Accessed April 2017].

[33] [Online]. Available: https://code.jquery.com/jquery-3.2.1.slim.min.js. [Accessed June 2017].

[34] [Online]. Available: https://cdnjs.cloudflare.com/ajax/libs/floatthead/2.0.3/jquery.floatThead.min.js. [Accessed April 2017].

[35] "Fonts.googleapis.com," [Online]. Available: https://fonts.googleapis.com/css?family=Open+Sans:400,300. [Accessed April 2017].

[36] [Online]. Available: https://fonts.googleapis.com/icon?family=Material+Icons. [Accessed June 2017].

[37] D. Gandy, "Font Awesome, the iconic font and CSS toolkit," [Online]. Available: http://fontawesome.io/. [Accessed June 2017].

[38] J. Sauro, "MeasuringU: 10 Things To Know About The Single Ease Question (SEQ)," 30 10 2012. [Online]. Available: https://measuringu.com/seq10/. [Accessed 6 2017].