Graduation Report:

Interactive Fault Trees

Anh Tuan Nguyen (s1816187)

Creative Technology

University of Twente

Enschede, July 2020

Supervisors: Dr. Marielle Stoelinga

ABSTRACT

Fault trees are a popular tool to visualize the risk of models in many technical fields. However, non-experts often feel fault trees are confusing, overwhelming, and difficult-to-understand. This study aims to find out the root of the problem of why fault trees are not easy-to-understand for non-experts and to build up an interactive prototype to solve this problem. The scope of this study narrows down to the qualitative information of fault trees.

Based on a review of the literature and user interview, there are two problems of fault trees found: lack of efficiency and lack of both visual and textual clarity. To solve these two problems, an interactive prototype is created. The prototype has two parts: Static layout and interactive features. Even though the status layout did increase efficiency, it mainly solves the clarity problem of fault trees. On the other hand, in five created interactive features, four of them focus on solving the efficiency problem and one solves the clarity problem. The evaluation of the prototype conducted at the end of the study shows the significant benefits of the layout and two interactive features named Expansion Button and Emphasis. The rest of the interactive features still need to improve further.

CONTENTS

1 INTRODUCTION

Originally, around the 70s, risk communication became more popular since the need of communicating the judgments of experts on the risk of industrial products to people (1). However, this purpose changed rapidly over the years because stakeholders and the general public often refused to be “enlightened” about the risk. They desired to be listened to and involved more often in the communication. Hence, the adaption was made: risk communication focuses not only on the expert’s side but also on the understanding of stakeholders such as policymakers, businessmen and the general public (1; 2). Due to this adaption, the communication between experts and non-experts encountered a problem: non-experts often experienced difficulty understanding complex risk issues(1). This problem requires a new way to present the risk issues to non-experts and one of the solutions is risk visualization. Eppler and Aeschimann define risk visualization as “ the systematic effort of using images to augment the quality of risk communication along the entire risk management cycle.”(1).

Fault trees are one of the popular tools of risk visualization. They are often used to visualize the risk of models in many technical fields such as mechanical engineering, aerospace, and nuclear power (3). Fault trees belong to the family of tree diagrams (3). In reality, engineers often use fault trees as a communication tool in order to discuss the risk of models with other co-workers, managers, and non-experts.

However, non-experts often experience difficulties when reading fault trees. They easily get lost and feel confused with all the information displayed in fault trees. This problem causes a limited understanding of non-experts about the risks that are presented in fault trees, creating ineffective communication between experts and non-experts. Surprisingly, there are not so many studies researching this topic while the trend of risk communication that stakeholders want to be more involved keeps increasing. This is the motivation for us to conduct this study to find out the most effective display method to make fault trees easier-to-understand to non-experts.

The results of this study contribute a certain value for fault tree makers. For example, engineers who want to make non-expert friendly fault trees could use this study as a reference. Another audience of this study can be researchers who want to create their own fault tree guide-

lines for a company. Furthermore, since this topic is relatively new, future researchers also can continue and improve this study.

The research of this study is " What is the most effective display method to make fault trees easier-to-understand to non-experts?". To answer this question, we first want to find the reasons why fault trees are difficult-to-understand for non-experts and then design an interactive fault tree visualization to solve the problem. This study carefully reports that process. There are ten chapters in this study. Chapter 1 is the Introduction. Chapter 2 is Background, introducing some background knowledge about fault trees. Chapter 3 is Problem Statement, introducing the problematic fact about fault trees. Chapter 4 is Problem Validation, introducing the core problems of fault trees that are solved in this study and its validation. Chapter 5 is Ideation, describing how the idea for the final prototype is formed. Chapter 6 is Specification, describing the user requirements and description of the final prototype. Chapter 7 is Realisation, reporting the process of creating the final prototype. Chapter 8 is Evaluation, reporting the evaluation results of the final prototype on how well it solves the problem. Chapter 9 is Discussion, discussing the limitations of the study and future work. Chapter 10 is Conclusion, showing the answer to the research question.

2 BACKGROUND

2.1 Fault Trees

Fault trees are one type of logical diagram, specifically in the family of tree diagrams. The main purpose of fault trees is " to identify potential causes of system failures by systematically breaking down high-level system failures into their causal factors", using graphical technique (4). Engineers often use fault trees to understand failure propagation and root causes and diagnose a fault that has happened in order to access the safety requirements of systems (4; 5; 6). Also as a function of a visualization, fault trees are used as a communication tool between experts and non-experts. Fault trees are widely applicable in different fields such as mechanical engineering, aerospace, and nuclear power (3).

Fault trees can analyze both qualitative and quantitative data information. Qualitative analysis points out the combination of events that cause the top event to occur (5; 4). With the graphical presentation of tree diagrams, readers can understand systematically the relationships of these events. Furthermore, quantitative analysis focuses on the dependability computation, which describes how dependable the system is (4). The computation uses probabilistic information such as failure probabilities, rates, or frequencies. From the computation, engineers can draw three conclusions of the system: the Reliability shows the failure frequency of the system within a given period; the Availability shows the average percentage of time that the system is functioning well; the Mean Time to Failure shows the average time for a system to fail (4).

Fault trees have three kinds of symbols: gate, event, and transfer. As can be seen in Figure 2.1, each kind of symbol has many different versions which refer to a different meaning (6). In Figure 2, an example of a fault tree is presented in order to show how these symbols are connected (4). Fault trees are read from the top to the bottom. The top event locates at the top of the fault tree. In Figure 2, the relationship between the top event and the below events is interpreted as the road trip is stranded when both phone and car fail. Continually, the phone fails because of either no connection or no power. The interpretations are applied similarly to

Gate Symbol		Event Symbol		Transfer Symbol	
Symbol	**Meaning**	**Symbol**	**Meaning**	**Symbol**	**Meaning**
	AND Gate		Basic event		Transfer-In
	OR Gate		Incomplete Event		
	Exclusive OR Gate		Conditional Event		
	Priority AND Gate		Normal Event		Transfer-out
	Inhibit Gate		Intermediate Event		

Figure 2.1: All symbols used in fault trees

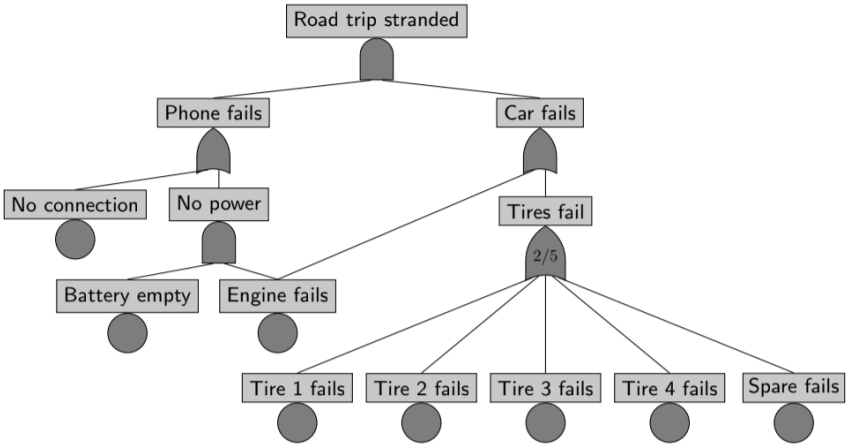


Figure 2.2: An example of a fault tree.

other nodes

3 PROBLEM STATEMENT

In practice, fault trees are often used as a communication tool in many important events. For example, in a presentation, fault trees are presented to different stakeholders such as technical and project managers in order to show the reliability of a model. This kind of presentation is cruel for testing the safety of a model because it is the moment engineers demonstrate their analytical results and managers have to make their judgment on whether the model is reliable enough in order to put it in operation. Unfortunately, non-experts such as the stakeholder in the example often experience difficulties when reading fault trees. They feel overwhelmed and easily get lost. Therefore, it is hard for them to make their own judgment on the reliability of the model. The problem with fault trees, regarding the ineffective information display, makes fault trees do not pursue greatly the purpose of communication tools. Surprisingly, there are not so many studies researching how easily non-experts understand fault trees, and more importantly, there is no guideline or framework that shows how to create visually effective and well-presented fault trees.

If the problem of ineffective display of information in fault trees is not solved, it can lead to serious safety problems while the model is in operation. The accident of the Fukushima Dai-ichi nuclear disaster in 2011 can prove is an example. One of the reasons causing that accident is the insufficient provision of information about the reliability and expected limit of operability of monitoring to operators (7). Thus, operators do not strictly follow the safety conditions while operating the monitoring.

Hence, the aim of this project is to find out the root of the reason why non-experts have a difficult time understanding fault trees and to create an interactive version of fault trees that assist non-experts to process the qualitative information of fault trees effortlessly.

4 PROBLEM VALIDATION

This chapter focuses on analyzing the problem of ineffective information display of fault trees. In the previous chapter, it is stated that non-experts have a difficult time understanding fault trees, which could lead to many safety problems while the model is in operation. So, in this chapter, the problem is analyzed deeply in order to answer the question: *why are fault trees not effective to display information to non-experts?*

This chapter started with the literature review on effective visualization in general. It suggests that effective visualizations can be evaluated in four different aspects: *Novelty, Informativeness, Efficiency,* and *Aesthetics*. In the context of visualization, according to many professional designers, *Aesthetics* has the main goal of using visual elements in order to increase Clarity. So, it is decided to evaluate *Aesthetics* in terms of *Clarity*. This chapter continues with the first conclusion stating whether the ineffective information display of fault trees is due to the lack of *Novelty, Informativeness, Efficiency,* or *Clarity*. However, the requirements of *Efficiency* and *Clarity* require more knowledge in order to conclude properly, so the chapter continues to look into visual principles in order to analyze the problem of these two requirements deeper. This chapter ends with the second conclusion that states the visual design mistakes causing the *Efficiency* and *Clarity* problem of the fault tree in the use case of (3). This conclusion for the use case is an example that can be generalized for the problem of fault trees in general.

4.1 Effective Visualizations

The limited amount of research on the effective information display of fault trees hinders our process of finding the root of the reason why non-experts do not understand the fault tree easily. Most of the online research is studying the guideline for effective data visualization but not modeling visualization. One of the few research found is *Effective Visualization in Modeling and Simulation* of (8). However, the authors also give a very vague evaluation method for concept and diagram visualization: evaluators often check the effective use of graphics in terms of visualizing the idea and concept. Even though the evaluation method of the authors is relatively general, it gives us a direction that graphics are one of the important factors for effective

modeling visualization.

Because of the limited research, design materials, and handbooks guiding how to evaluate an effective modeling visualization, we decided to go broader. The range of the search expanded to look for the requirement of an effective visualization in general. The book (*Beautiful Visualization: Looking at Data through the Eyes of Experts*) of (9) was found and chosen to give readers an overview of different aspects of effective visualization. This book is not a scientific journal, but it covers several ideas from different authors.

4.1.1 Novelty

Novelty is an important aspect of an effective visualization but not always necessary. According to (9), Novelty is defined as "a fresh look of the data or a format that gives readers a spark of excitement and results in a new level of understanding." Even though familiar visualizations have their own benefits, they no longer have the ability to surprise or delight readers, meanwhile, novel visualizations can do just that. They can attract and engage readers. Novelty is a common quality that has been mentioned in many research (10; 9; 11). However, the effect of Novelty seems to be stronger to designers rather than the general public (10). The research shows that general audiences feel more connected to familiar visualizations while designers feel more engaged with novel visualizations. Thus, for engineers, novelty is not always their first concern when making a visualization for the general public.

4.1.2 Informativeness

Informativeness is an essential requirement of any visualization, beautiful or not. (9) defines an informative visualization as "providing access to information so that users may gain knowledge". Informativeness has two main particulars: the intended message and context of use. Firstly, the intended message refers to the message, goal, and knowledge that visualizations are trying to convey. The intended message of good visualizations should be simple, clear, and truthful (9; 12). It is also the standard to judge the success of visualizations. Secondly, the context of use refers to the situation where visualizations are used (9). Good visualizations should fit into either two contexts of use: to reveal the knowledge of designers or to aid designers to gain knowledge. For example, fault trees are often made by engineers to visualize the analytical result of the risk of models. So fault trees are the presentation of what researchers already know. On the other hand, an example of a visualization used to aid people to gain knowledge are mind maps. People often use mind maps to systemize their knowledge in order to remember it better.

4.1.3 Efficiency

Efficiency is the third important requirement of an effective visualization. According to (9), if visualizations are considered to be efficient, it means their intended message is delivered straightforwardly and without any distraction. The efficiency of a visualization depends on two factors: the amount of the information displayed (on the screen) at a time and the amount of irrelevant information that does not contribute to the main intended message. It is obvious that the more information displayed at a time, the longer time readers need to spend to process. Also, if the irrelevant information is displayed at the same time with relevant information, the irrelevant one starts being noisy, distracting the process of interpreting the useful information. For example, Figure 4.1 shows an example of an inefficient graph that has too many lines overlapping each other. As can be seen in Figure 4.1, if readers are trying to understand the information presented in the blue lines, the purple, red, and yellow lines are distracting the process.

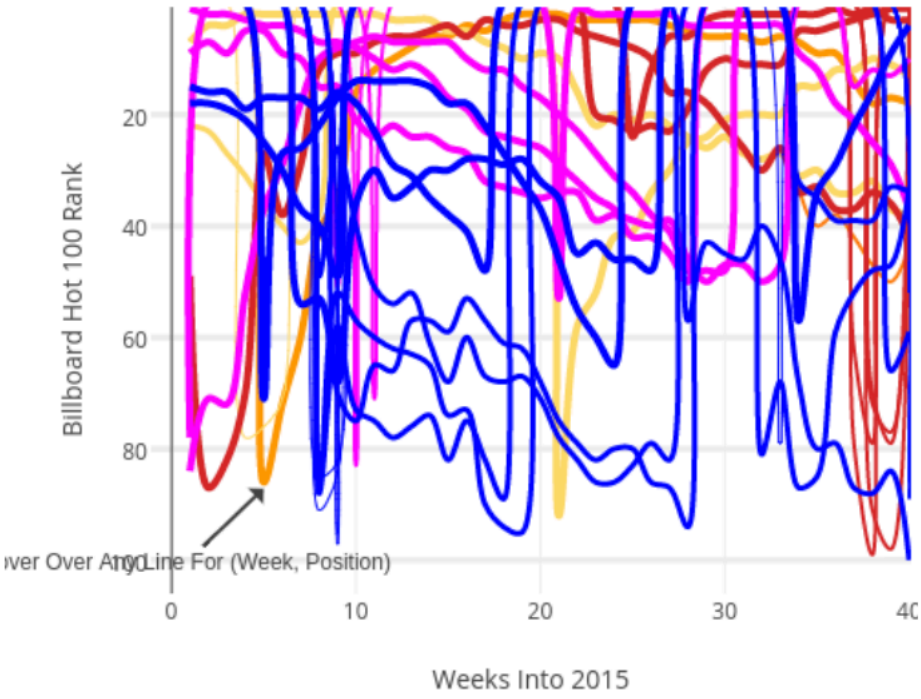


Figure 4.1: The inefficient graph which has several lines overlapping each other.

Efficiency also depends on the hierarchical organization of a visualization. According to (13) a hierarchy is defined as “a system that is structured in layers or levels that have asymmetric relations.” In the context of visualization, a hierarchy can be understood as a logical organization of data or information into different layers or levels. Node-link diagrams are a great example of hierarchical visualization (14; 15). Node-link diagrams have a structure of botanical trees that present the information in the leaf (node) and the hierarchical organiza-

tion of those leaves by the branches. Therefore, the efficiency of node-link diagrams depends highly on how clear the hierarchical structure is visually presented. According to (15), some visual elements such as proximity, color, and size are often used to make the hierarchical structure clearer. Hence, when evaluating the efficiency of node-link diagrams, we should not only focus on the amount of useful and distracting information but also on the visual clarity of the hierarchical structure of the diagram.

4.1.4 Aesthetics

For visualizations, Aesthetics brings a lot of benefits to user experience but it lacks guidelines. (11) believe an aesthetic visualization can improve task performance and user satisfaction. The interesting finding in this research is that users are more forgiving to the visualization which has system errors but is aesthetically pleasing. Similarly, (9; 16) thinks Aesthetics can improve the hierarchy and communicational purposes of visualizations. However, there seem to be no design materials or frameworks that guide designers to achieve Aesthetics. It might be because aesthetics are highly subjective. Also, aesthetic judgment is not based on a single visual element but a combination of several visual elements, so it is very hard to create a framework.

Even though there is a lack of guidelines to achieve aesthetic, the primary goal of aesthetics is supported by many researchers. Researchers agree that the goal of aesthetics is to increase the clarity of a visualization in order to aid understanding (10; 9; 16; 11). In the context of visualizations, visualizations do not need to be visually impressive. Instead, all visual elements should be used in order to present information in the most efficient and effective way (9). All of the visual effects that do not follow this goal should be eliminated. Hence, in the context of visualization, Aesthetics can be substituted by the quality of Clarity of visualization.

4.2 First Conclusion

The first conclusion is drawn based on the result of whether fault trees meet the four requirements of effective visualization: Novelty, Informativeness, Efficiency, and Aesthetics (Clarity)

Novelty

Fault trees do not totally need to be novel in order to have an effective information display. It is because fault trees still can be very effective when following the cognitive concept of node-link diagrams: introducing the information at nodes and using the edges to represent the relationship among those nodes. Hence, the concept of fault trees does not to be changed. Plus, the scope of this project does not focus on creating a new version of fault trees but to improve the effectiveness of the information display of faults trees.

Informativeness

Fault trees meet the requirement of Informativeness. First, it shows clearly the intended message which is to enhance the understanding of the failure system of the model. Readers also have not much trouble to realize this goal. Second, fault trees are in one of the two contexts of use: to reveal the knowledge/ finding of designers. After identifying and analyzing the risk of the model, engineers use fault trees to systemize the risk events and their relationships through the gates and edges. Hence, there seems to be not much improvement in this requirement.

Efficiency

Fault trees do not meet the requirement of efficiency completely. If the intention of readers is to have an overview of the entire failure system of the model, fault trees are efficient. However, if the intention of readers is to debug or focus on the problem sequence caused by one basic event, fault trees might not be super-efficient. When readers only need to see one branch of the tree, the fact that fault trees showing the entire diagram is distracting.

Furthermore, in terms of hierarchical organization, fault trees are efficient. Because fault trees have the structure of botanical trees, useful information is organized logically into different layers. However, the effectiveness of the hierarchical organization of fault trees depends highly on the visual element such as color, proximity, and size used in that fault tree. At this stage of the project, the knowledge is not enough in order to conclude whether the hierarchical organization is effective.

Aesthetics (Clarity)

There is not enough knowledge at this stage to conclude whether fault trees meet the requirement of Clairy because of two reasons. Firstly, there is no guideline or framework that guides us to evaluate the Clarity of fault trees. Secondly, Clarity depends on the combination of the use of visual elements. It is far too complex to give any conclusion without looking deeply into visual principles and the use of visual elements. Hence, it is necessary for the later stage of this project to properly study the visual principles that make a clear visualization.

4.3 Interpretation Process

The process of how human brains pick up and interpret visual information is discussed here in this chapter. During the interpretation process, visual principles and visual elements are carefully introduced in the context of how and when human brains interpret those elements. This chapter provides the missing knowledge required in the previous chapter in order to conclude the Efficiency and Clarity of fault trees.

The interpretation process discussed in this chapter is introduced in the book of (17): *Information Visualization: Perception for Design* and some other journal articles. According to (17), the process of interpretation has three main stages: Stage 1: Parallel processing to extract low-level properties of the visual scene, Stage 2: Pattern perception, Stage 3. Visual working memory. It is reasonable to say that the whole process of interpretation will be accelerated if each stage of the process is improved. To do so, it is important to understand what humans interpret in each stage and how it contributes to the whole process. In the following sub-sections, each stage of the process is described and some visual elements are explained how they are applied in visualizations.

4.3.1 Stage 1: Parallel processing to extract low-level properties of the visual scene

In the first stage, visual information is processed parallelly in order to extract the basic features of the surrounding. The lesson learned in this stage is that a good visualization first needs to be a clear and all-encompassing visualization. The process of interpretation starts when the light reaches human eyes, allowing them to be able to see. During this stage, billions of neurons in the eyes work parallelly, transforming the information into electrical signals and sending them to the primary visual cortex at the back of the brain. As a result, features from every part under the visual range are extracted into meaningful information for the brain such as different light contrast of objects and color(17). This process enables humans to detect an object out of its surroundings. When designing a visualization, designers should pay attention to the clarify of a visualization. A clear visualization is the one allowing viewers to detect the information effortlessly.

Luminance contrast

The clarity of a visualization depends on two dimensions of color: luminance contrast and color hue. First, luminance also has another name as color value, one of the three dimensions of color (18; 17). In theory, color has three dimensions: hue, saturation, and color value which

is considered to be the most important dimension (17). Luminance or color value is usually represented in a grayscale, the higher value the color has, the whiter the color is. So the color having the highest value possible is white and, in contrast, the color having the lowest value possible is black (17). (17) believes that human eyes can detect an object out of its surroundings just fine as long as they have different luminance values. The evidence for his belief is black and white movies. During the time people were able to make only black and white movies, viewers still can distinguish things effectively most of the time. It is because human eyes are very sensitive to luminance contrast.

In the context of making a visualization, luminance contrast is extremely important when choosing colors for an object and its background. As can be seen in Figure 4.2, the text can be seen so much better in the right picture compared to the text in the left picture. The reason simply is that the right picture has higher luminance contrast between the color of the text and its background, compared to the luminance contrast in the left picture. If designers make an advantage of luminance contrast, their visualization might be very clear for the viewers to see every little detail. However, in practice, people like to choose more colors such as red, yellow, and green rather than just different grays color or black and white. Their choice sometimes makes the visualization very clear, but sometimes it does not. It could be because they do not know even different colors such as red and yellow also has a different luminance range (17). For example, red has a very low luminance range. People can only see red as red if it is in its low luminance range. Otherwise, red turns into pink, if people push it out of its luminance range by increasing the luminance value.

Figure 4.2: The same texts are written in a background with different colors. The right picture shows the higher luminance contrast while the left picture shows the lower luminance contrast.

Color Hue

Humans eye also can detect an object out of its surroundings by its color hue. As mentioned earlier, the hue is also one of the three dimensions of color, determining the different wavelengths

of color in the light spectrum. Each color such as red, yellow, or green has its own wavelengths range (19). Human eyes can read the color hue quite effectively (17). For example, the two pictures in Figure 4.3 are taken from the book of (17): *Information Visualization, Perception for Design*, showing how human eyes can detect an object by its color. It might be very hard for human eyes to detect the cherries hidden in the surrounding leaves in the left picture. However, it is so much easier with the help of color in the right picture: the red cherries are so outstanding from the green leaves. The ability to distinguish things by their color of humans have been evolved for a long time (17). It is the reason why humans develop some perceptions of color. First, humans assume objects that have the same color are more likely to be the same type. Second, humans think different colors have different meanings. For example, humans can tell when bananas are ripe by looking at their color: green means unripe while yellow means ripe.

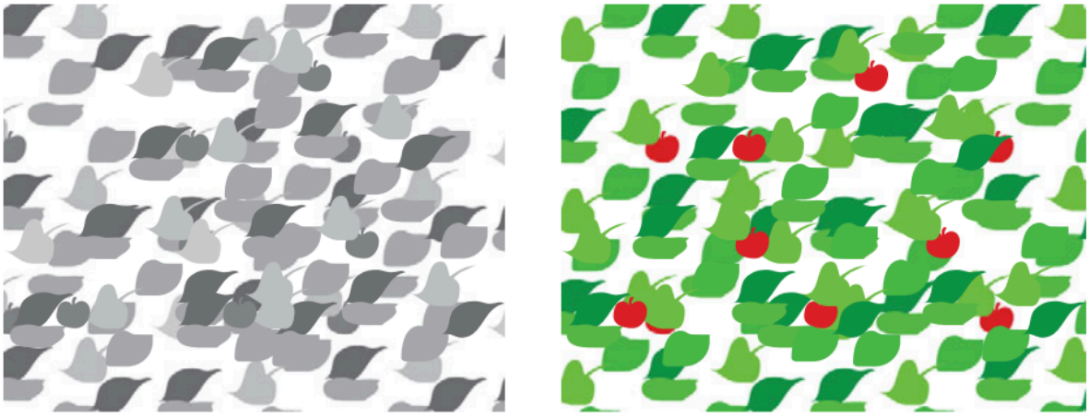


Figure 4.3: The cherries can be detected easily with the help of color.

Designers usually apply the color perceptions of humans in their visualization. First, because humans have a perception that different objects have different colors, colors are usually used to label and categorize (20; 17). So people often use colors to represent nominal data. Second, in visualization, each color also has its own meaning (17). For example, in the context of visualization, red means danger, green means safe, and yellow means warning. Hence, choosing the color wisely with correct meanings makes the visualization clear not only physically but also cognitively.

4.3.2 Stage 2: Pattern Perception

In the second stage, the human brain analyzes visual relationships among objects in order to find visual regions and simple patterns. The lesson learned in the second stage is that a visualization also needs to be organized and structured logically. After being able to see objects in the visual

field, human eyes land at a specific object and then start looking at its neighbors to identify the differences and similarities among them (17). The similarities represent that those objects are similar kind and create regions of visual such as regions of the same color and regions of the same texture. The differences imply that each object has a different meaning and when they are seen as a whole, they create simple patterns. Each pattern also represent specific relationship such as casual relationship (17). The requirement of a good visualization is that the visualization should ease the process of searching patterns and present clearly the meaning of patterns. The following paragraphs describe more detailed the process of how the human brain searches and perceives patterns

Eye Movements

The human eyes pick up an object to look at, in order to start the process of pattern searching. Naturally, eye movements focus on the visually strongest area, which is usually the different area compared to the rest (17). However, there are three factors that can influence that process (17). First, it is a priori salience, which can be understood that some patterns naturally excite the brain more than others. For example, the yellow color always attracts more attention than the green color. The second influencing factor is the top-down salience modification. It means that human intention also decides what they see first. For instance, if they wish to find a mostly vertical elongated symbol, the vertical orientation will gain enhanced sensitivity. Finally, it is the scene gist. It says that experience/ memory influences what human eyes land first. The brain is able to recognize quickly the type of scene that it has seen before, then it influences the way humans look at things based on the last experience. Hence, when designing a visualization, designers should be aware of the effect of prior salience and use them intentionally. Also, providing ahead hints that tell users what they should look at might be one way to ease the searching pattern process.

Pre-attentive Process

The human brain detects the similarities and differences among objects, which is called the preattentive process. It is also defined as the process of understanding visual distinctiveness (21; 17). It is best to explain how the preattentive process works by examples. In picture A of the Figure 4.4, counting the 3s is so much harder than counting them in picture B. The reason is that the visual element, colors in this case, is used to make the targets more outstanding than the background, which makes the searching process become so much easier. The human brain can detect eleven visually distinctive elements during the preattentive process: *orienta-*

tion, curve/straight, shape, length, size, color, light/dark, enclosure, convex/concave, motion, and addition (Figure 4.5)(21; 17; 22). In practice, designers sometimes combine two or more visual elements to increase the distinctiveness among objects (17). Organized and structured visualization should optimize the effect of the preattentive process in searching and should not cost much brain energy to find the target or starting point.

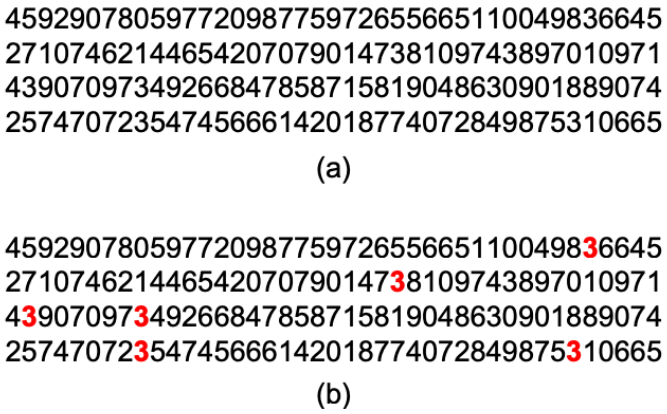


Figure 4.4: By using a different color, the 3s are much more outstanding, which makes the searching the 3s much easier in picture B compared to picture A.

Gestalt Laws

The human brain pays attention to group and structure the visualization by perceiving its patterns. One of the most famous and important theories on pattern perception is Gestalt Laws (17; 22). The Gestalt theory includes eight laws, however, this paragraph only discusses six laws that are relevant to this project: *Connectedness, Continuity, Proximity, Similarity, Common Field, and Symmetry*. In these six laws, except *Symmetry*, all five laws describe how the human brain perceptually groups objects together. According to Gestalt, the strongest way to group objects is to make a connection (*Connectedness*). The most common method to show the connection is by using lines or ribbons of color, which can be seen in Figure 4.6. Another law that supports the previous law is *Continuity*. This law says that connections are better shown by a smooth line rather than the one containing abrupt change. The law is demonstrated in Figure 4.7. The human brain also groups objects by spacial position (*Proximity*): objects that are closed together usually perceptually grouped together. Also, objects that are similar also usually grouped together (*Similarity*). Finally, the human brain tends to group objects if they are moving in the same direction (*Common Field*). There are many ways to present the directions such as arrows, change in the thickness of the lines, and the light fading. Different from all

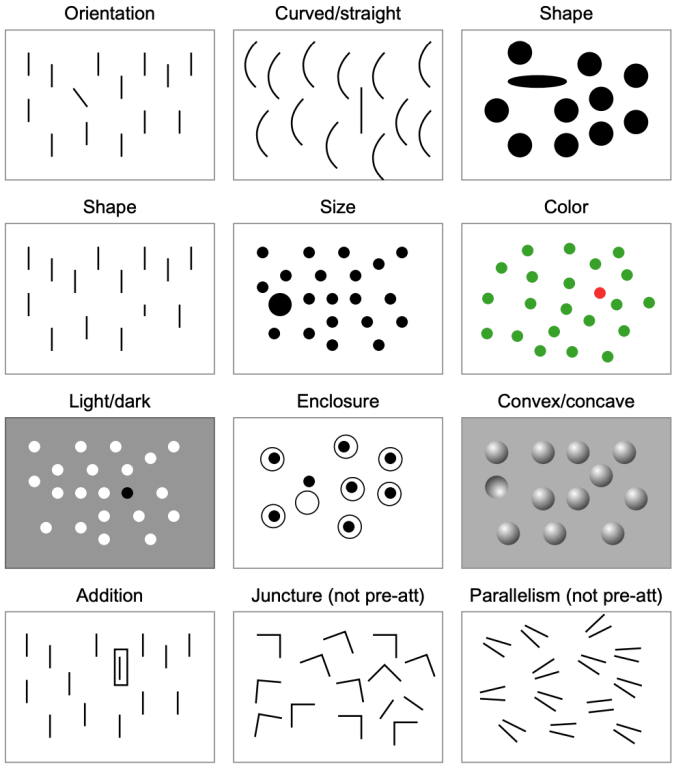


Figure 4.5: Ten visual elements that the human brain can process during the preattentive process.

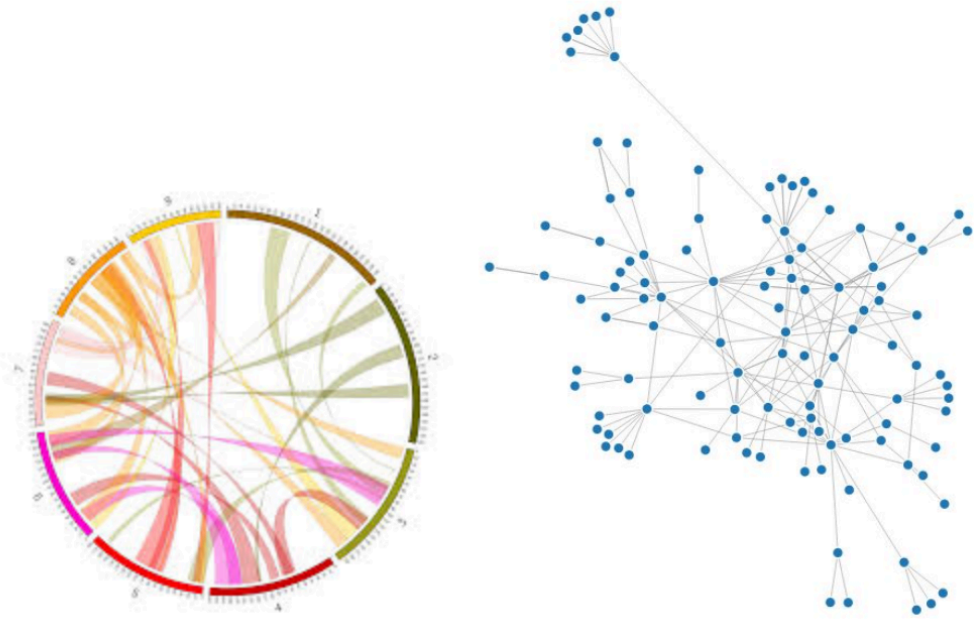


Figure 4.6: The most common method to show the connection is by using lines or ribbons of color.

five above laws, *Symmetry* focuses more on organizing principles: objects that are arranged symmetry are perceived more strongly as forming a visual whole.

The Visual Grammar of Node-Link Diagrams

Similar to patterns, the visual grammar of visualizations is also perceived in the second stage of interpretation. Because the scope of this project focuses on fault trees, the visual grammar of node-link diagrams is discussed. As can be understood by the name, node-link diagrams include two main elements: nodes, representing different kinds of entities, and links, representing the relationships between them (17). Nodes are often presented in the shape of an outline, box, circle or small symbol. Inside, there often are some texts explaining the meaning of the nodes. In some cases, designers also demonstrate extra information of nodes by using its shape, color, and size as secondary attribution, like in Figure 4.8. Furthermore, links are often presented by lines. Even though according to Gestalt Laws, lines are very powerful to connect objects together, lines used in node-link diagrams are also quite ambiguous (17). Without any extra visual cues, lines do not show clearly which type of relationships between nodes. So that is the reason why designers often replace lines by arrows or adding some extra text explanation. The effectiveness of using arrows instead of lines is proven in the study of (23).

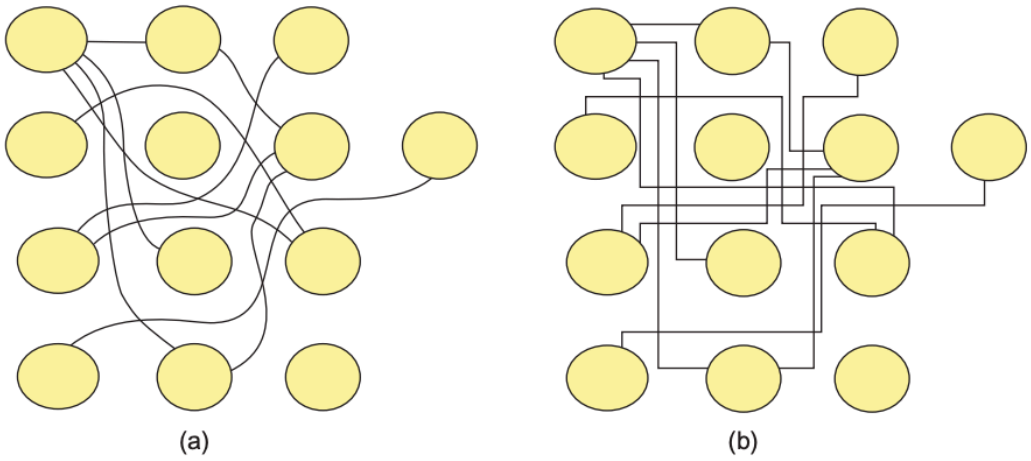


Figure 4.7: Connections are better shown by a smooth line rather than the one containing abrupt change.

Figure 4.8: Designers also demonstrate extra information of nodes by using its shape, color, and size as secondary attribution.

4.3.3 Stage 3: Visual Working Memory

The third stage of interpretation (Visual working memory) is the highest level of perception where the human brain spends active attention learning and memorizing visual objects. The lesson learned during this stage is that a good visualization should be remembered easily. After the second stage of pattern perception, visual objects are identified and briefly stored in a kind of memory called the visual working memory. The goal of this stage is to analyze these objects in order to remember them (17). However, due to the low capacity of the human visual working memory, humans can only hold three to four objects at a time with intense concentration. The process of learning these objects is done by a sequence of visual queries. For example, when viewers read a road map, the visual queries can be: where is the starting point, where is the ending point, how are they connected together. These visual queries can be answered through visual search strategies (17). Each kind of visualization often has one or two visual search strategies. Hence, if designers understand the visual search strategy of the visualization and facilitate it, designers can make their visualization easy-to-remember.

In the scope of this project, the visual search strategy of node-link diagrams and its limitations are described. The strategy has two main steps: the first step is finding the starting and ending nodes, the second step is following the line connecting these nodes together in the correct direction (17). These two steps can be repeated several times, depending on the number of nodes in between. In the case of tree diagrams, from the long-term experience, people often know where the starting point and ending point are. The starting point of tree diagrams is usually is the top node located at the top of the diagrams and the ending points often are the lowest point of the branch. However, as mentioned above the human visual working memory capacity is low, viewers do not easily process a path having more than six segments (seven nodes and six edges). If the number is higher than that, viewers often need extra help from verbal working memory, for example repeating their findings verbally. Furthermore, another limitation of the strategy is that viewers can only pay full attention on one path. Thus, when viewers move on to the next path, they start forgetting the visited one. Hence, in practice, when designing an interactive node-link diagram, designers should simplify the path as much as possible and provide a function/ ability to highlight the visited path.

4.4 Second Conclusion

The second conclusion shows the analytical result of the Efficiency and Aesthetics (Clarity) of the fault tree used in the study of (3)(Figure 4.9). Because the layouts of fault trees are relatively different from engineers and generated software, it is difficult to analyze these two requirements for the general fault trees. Hence, it is decided to use the fault tree of (3) as a use case in this project. Even though the reliability of this conclusion is limited in this use case, it can be a reference for the different layout of other fault trees

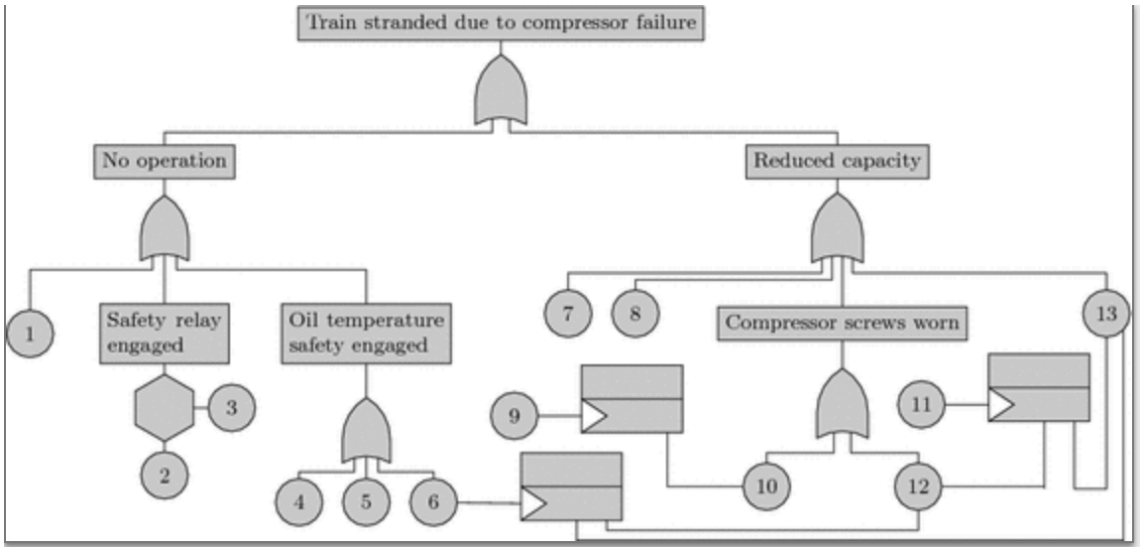


Figure 4.9: The fault tree used in the study of Ruijters et al (2006)indicates the major failure modes of the compressor.

Efficiency: Hierarchy

The hierarchy of fault trees relays on Color, Eye movement, Size, and Proximity. The hierarchy problem of the fault tree is analyzed below: Colors: there is no use of color in order to emphasize the different kinds or levels of information.

- **Eye movement:** There is no emphasis on the starting point of the visualization. It does not facilitate the reading process and the recognition of what information is more important.
- **Size:** there is no use of size in order to emphasize the different kinds or levels of information.
- **Proximity:** The branches are too close to each other which creates very little negative space. Therefore, distinction in space between branches is very weak and sometimes confusing.

Clarity

To examine in which part the fault tree is clear or unclear for non-experts to understand, a list of visual information processed in the three stages are described below:

- **Luminance Contrast:** As can be seen in Figure 4.9, the contrast of the white background, the grey symbols, and the black text is high enough. That means readers can read the text inside the grey symbols and distinguish them from the white background without any problem. However, the improvement can be put in the contrast of the black and grey. By increasing this contrast, readers can read the text clearly in the further distance.
- **Color:** All of the components of the fault tree such as gates, basic events, and intermediate events have the same color. It creates an unclear impression that they are different kinds. Plus, the chosen color also does not bring suitable meanings to a visualization of risk.
- **Eye movement:** As discussed in Stage 3 (Section 4.3.3), readers often know where the starting point of the diagram is. So it is lucky that the human eyes always land at the correct starting point.
- **The pre-attentive process:** readers can distinguish the differences among gates, basic events, and intermediate events. However, it can be improved by changing their color.
- **Gestalt Laws:** the fault tree follows correctly the two laws: *Connectedness* and *Similarity* . However, it does not follow *Continuity, Proximity, and Symmetry*. Figure 4.10 shows that the edges of the fault tree contain a lot of corners (abrupt changes).Figure 4.11 shows that there are some symbols that are not supposed to be grouped but are too close to each other. And Figure 4.12 shows that the fault tree is not organized symmetrically. Plus, the fault tree contains ambiguous lines instead of arrows or having explanation text.
- **Visual working memory:** As can be seen in Figure 4.13, the number of nodes between the starting point and the ending point at some branches is just equal to the limitation of easy-to-remember node-link diagrams (7 nodes).

4.4.1 Summary

The ineffective information display of fault trees is often due to the lack of efficiency and clarity. In terms of efficiency, fault trees usually show a great amount of information in one time which makes the act of understanding one branch from the whole tree of non-experts more difficult.

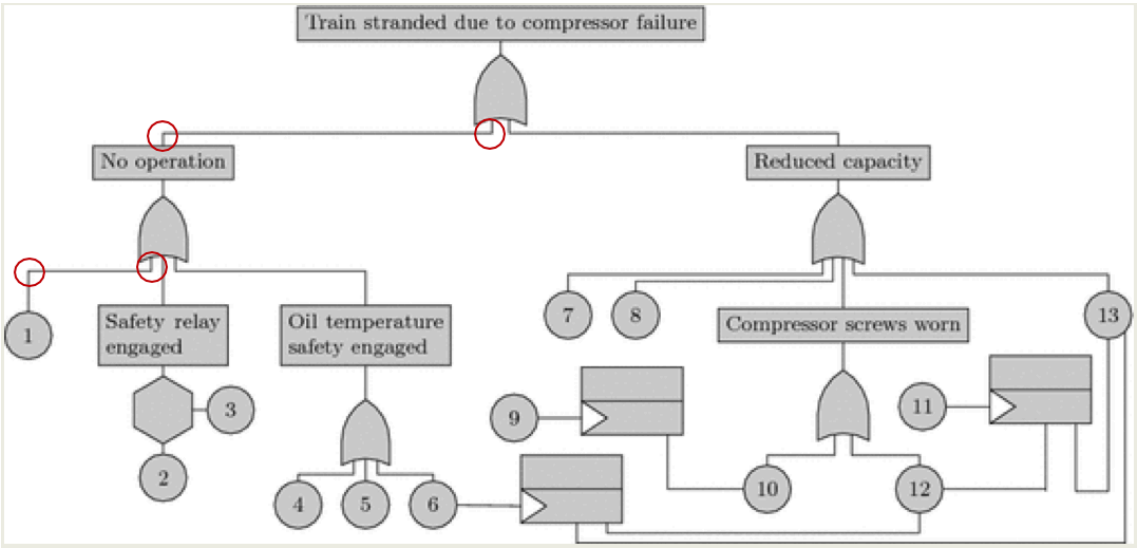


Figure 4.10: The edges of the fault tree contain a lot of corners (abrupt changes).

Furthermore, the conclusion drawn from the use case fault tree in the study of (3) shows the lack of efficiency might also because of ineffective hierarchical organization. In other words, the poor use of visual principles and elements make the hierarchy of fault tree poorly presented and sometimes confusing. In terms of clarity, the conclusion drawn also from the use case fault tree shows the mistake of using color (contract and hue), pre-attentive methods, and Gestalt laws, and the lack of eye movement guides make the fault tree visually unclear. To sum up, the lack of efficiency and clarity in the layout of fault trees due to the poor use of visual principles is the reason why non-experts have a hard time understanding fault trees.

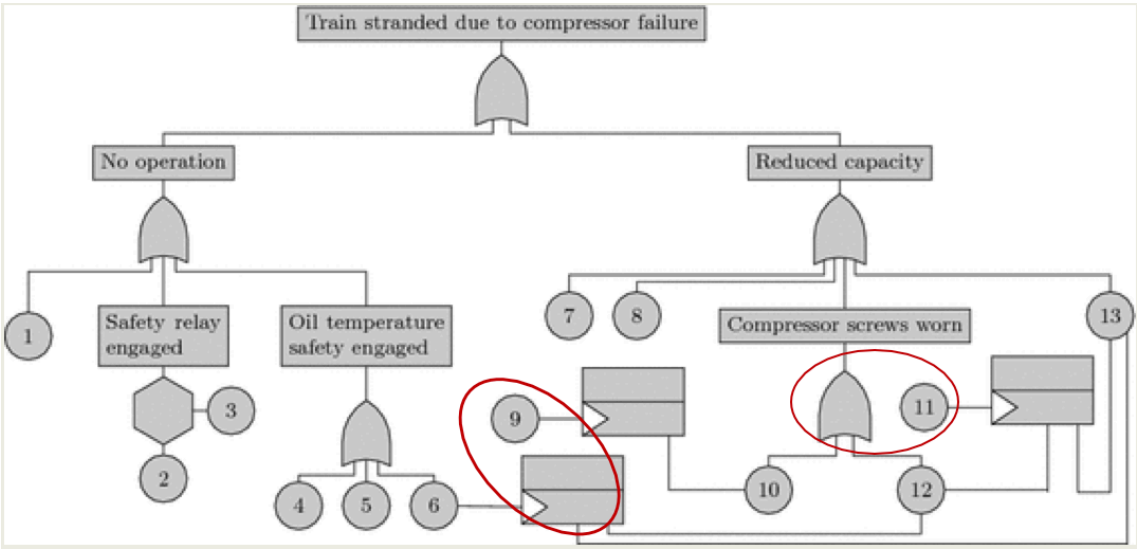


Figure 4.11: There are some misleading proximity problems in the fault tree.

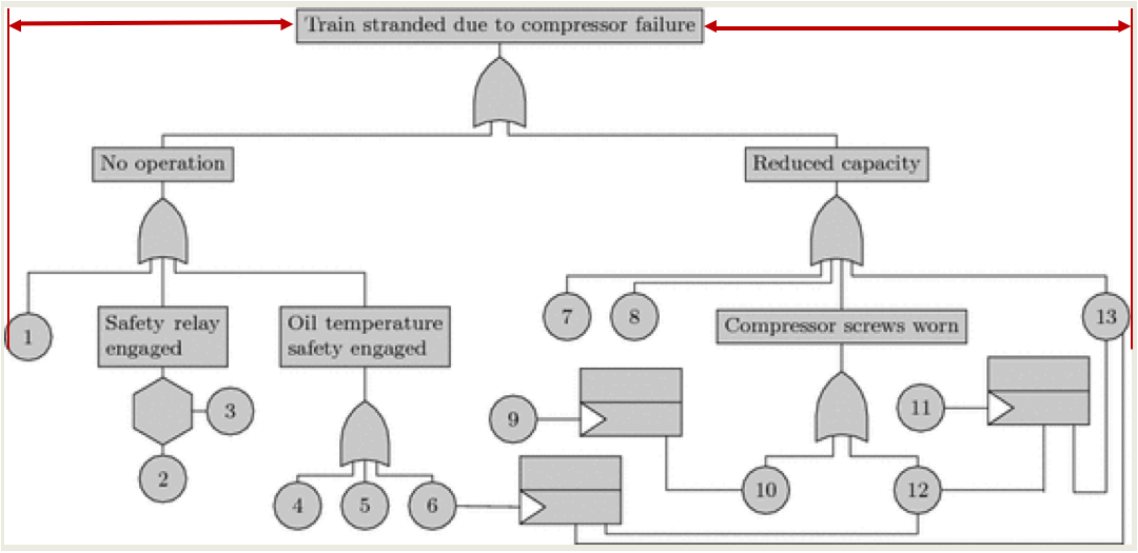


Figure 4.12: The fault tree is not organized symmetrically

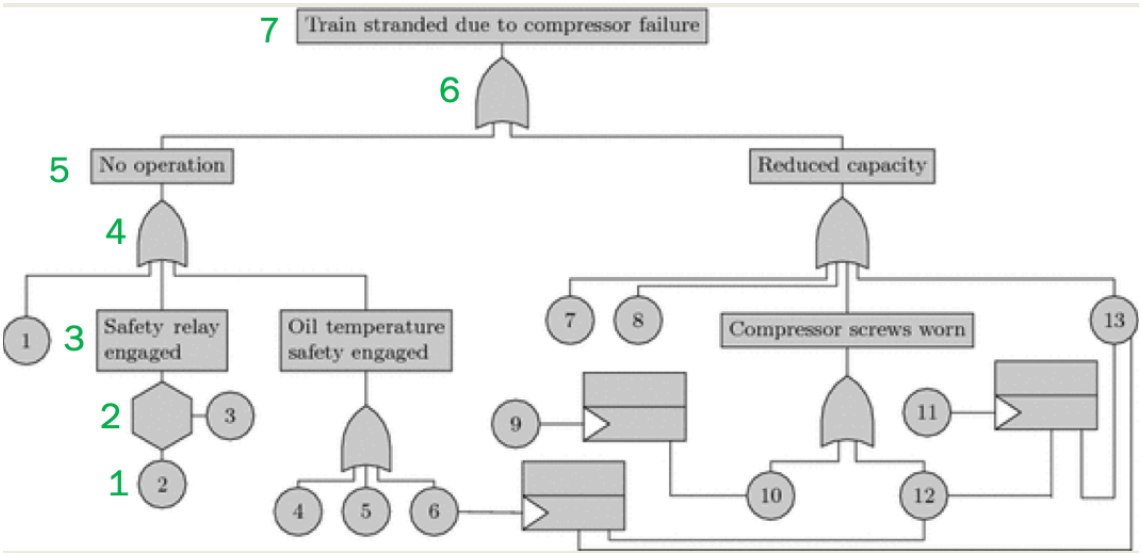


Figure 4.13: The number of nodes between the starting point and the ending point at some branches is just equal to the limitation of easy-to-remember node-link diagrams (7 nodes)

5 IDEATION

This chapter describes the process of ideation for the prototype of the final product. The previous chapter clearly points out that the ineffective information display of fault trees is mainly because of the visual design mistakes that make fault trees unclear and inefficient. Therefore, the prototype is designed in order to increase the efficiency and clarity of fault trees. The fault tree that is demonstrated in the prototype is the air compressor fault tree of (3)

This chapter has three sections: Static Visualization, User Interview, and Interactive Visualization. The first section indicates the process of applying the visual principles discussed in section 4.3 into the layout of the use case fault tree. The new layout then is tested with potential users. After the interview with users. The limitations of the new layout and the static layout of fault trees, in general, are found, which are the main problems that the interactive visualization will try to solve. After the interview, the expectations of users in the interactive visualization are also collected. They are the valuable design inputs for the interactive version of the fault tree. Finally, the chapter ends with the third section: Interactive Visualization. This section describes the research on the functional requirements of comprehensive software visualization tools and the use of motion in visualization. This research is the reference for the design of the interactive version of the fault tree which also is the prototype of the final product developed in this project.

5.1 Static Visualization

This section focuses on creating an efficient and clear statics layout for the use case fault tree in Figure 4.9. The static layout is the result of applying all the visual principles and elements that are discussed in the previous section: The interpretation process. This section has two subsections: Efficiency Realization and Clarity Realization. Efficiency Realization describes the application of visual principles in the original layout of the use case fault tree in order to achieve Efficiency. Similarly, Clarity Realization also describes the application of visual principles in order to achieve the Clarity of the fault tree. The static layout is the foundation for the interactive version which is discussed in the further chapter.

5.1.1 Efficiency Realization

As mentioned earlier, fault trees often have two efficiency problems which are too much useful information displayed at one time, and ineffective hierarchical organization. It is admitted that the static layout of fault trees could not reduce or adjust the amount of information to suit the need of readers automatically. However, it can improve the hierarchical organization by applying some visual principles, which will increase the readability of fault trees.

Color, Eye movement, Size, and *Proximity* are the visual principles and elements used in order to improve the hierarchy. Firstly, the different kinds of information such as gates, intermediate and basic events are coded in different greyscale colors. In Figure 5.1, the gates are coded in black, the intermediate events are coded in medium grey and the basic events are coded in a lighter grey. The reason why information is coded in greyscale colors is that the huge effect of color hue is wanted to be minimized in the early stage of design. The color hue will be applied in the last step of the process. Secondly, the eye movement of readers is guided to the starting point (the top event) by color. The top event is the only event in black, like in Figure 5.2. Making differences by color is one of the pre-attentive methods introduced in section 4.3, which is either not too settle or too obvious. Thirdly, in Figure 5.3, the size of the intermediate events is made to be uniform instead of being adjustive to the amount of text like in the original layout. The advantages of having a uniform size are that it makes the events look more similar and also easy to organize in the grid. Finally, Proximity is the most important factor creating a big change in the layout. The Gestalt law of proximity is applied: things that are related to each other should be placed closely. The result is demonstrated in Figure 5.4. All events are arranged symmetrically as the application of the Gestalt law of symmetry as well.

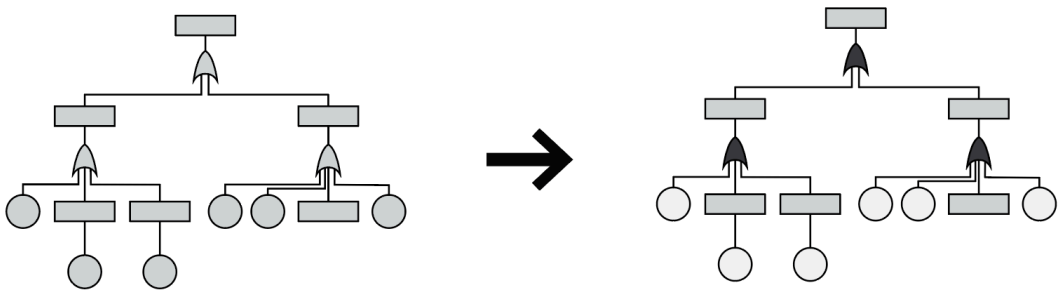


Figure 5.1: The the greyscale color change is applied into the simple version of the use case fault tree.

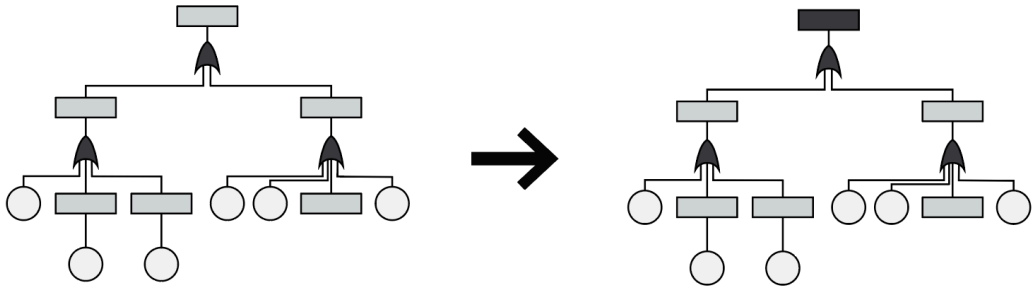


Figure 5.2: The color of the top event is changed in order to guide readers'eyes to the starting point.

5.1.2 Clarity Realization

The problem of clarity in the layout of fault trees occurs when engineers do not follow or use properly visual principles and visual elements. In order to increase clarity, the layout should follow the visual principles. While improving the efficiency of the layout, we already made the fault tree clearer for reading. This is the overlap between efficiency and clarity, which is the reason some of the visual principles are not mentioned in this section. In this section, some more visual principles and the result of (16) in aesthetics heuristic are applied.

Four more visual principles continue to be used in order to increase the clarity of the use case fault tree's layout. Firstly, as be shown in Figure 5.5, the luminance contrast color of the grey background and the text is more than a 3:1 ratio as suggested in the book of (9) Secondly, the Gestalt law of Continuity is applied, changing the sharp corner in the edges into smoother corners (Figure 5.6). Thirdly, the arrows are added in order to solve the ambiguousness of lines in node-link diagrams. The arrows imply the causal relationships between the below events and the top events, as can be seen in Figure 5.7.

The result of (16) in aesthetic heuristic gives some suggestions on how to keep node-link diagrams being clear and understandable. The result is shown in Table 5.1. There are three metrics in the table: node, edge, and overall layout, and for each metrics, authors suggested some heuristic methods to make diagrams easier to read and understand. This material acts as a great reference while designing a layout of fault trees.

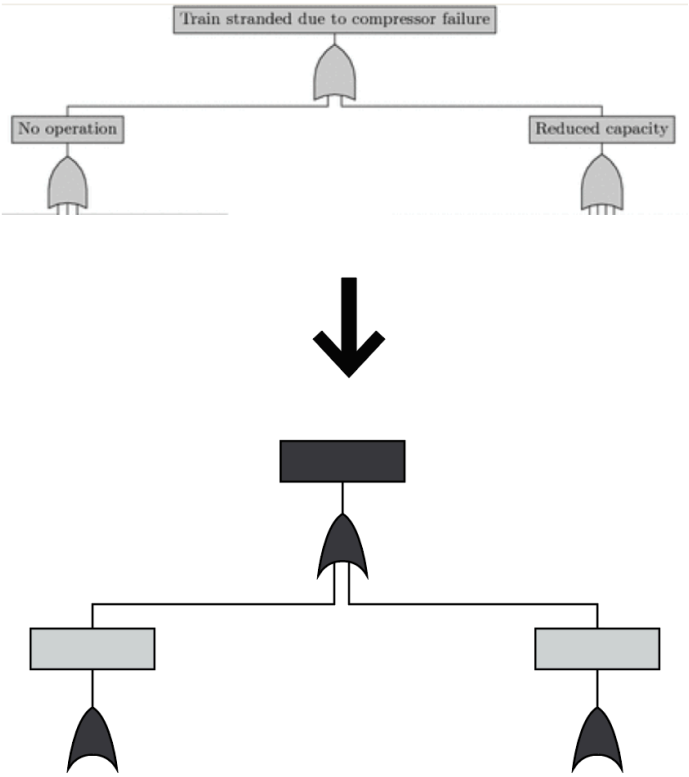


Figure 5.3: The sizes of the intermediate events are uniform in order to increase the similarity of the event and ease to organize.

5.1.3 The New Layout of the Fault Tree

Figure 5.8 shows the new layout of the fault tree after be applied all the visual principles. This new layout is used in the User Interview reported in the next section.

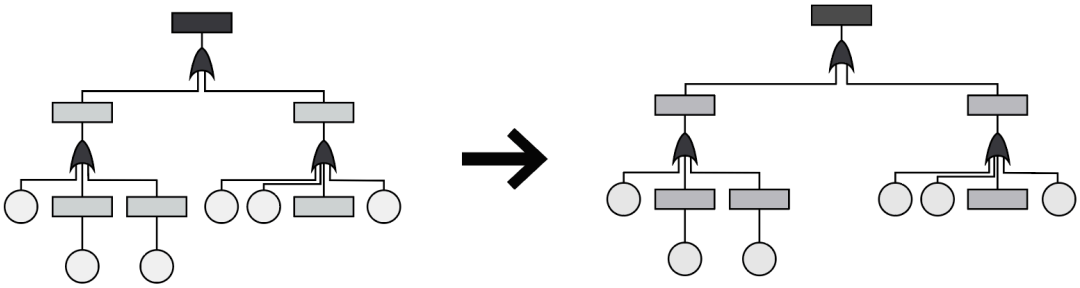


Figure 5.4: The Gestalt Law of proximity is applied into the simple version of the use case fault tree.

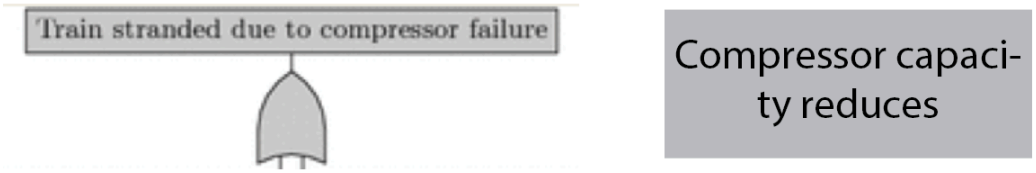


Figure 5.5: The grey colors between the original and the updated version are slightly different. The different color between the text and the background is more obvious.

5.2 User Interview

During the stage of Ideation, an interview with potential end-users was conducted. This exploratory interview plays an important role to evaluate our lo-fi prototype made in the previous chapter and to find the requirements (user and functional) of our final prototype. The findings in this interview are divided into two parts. The first part talking about the evaluation of the lo-fi prototype and new visual and clarity problems are reported here in this section. However, the second part talking about the user requirements and the expected features of our final prototype are reported in the next chapter.

Interview Goals

The interview has two goals: evaluate the lo-fi prototype and find requirements for the final prototype. Firstly, in order to evaluate the lo-fi prototype, the answers to the three question below should be found:

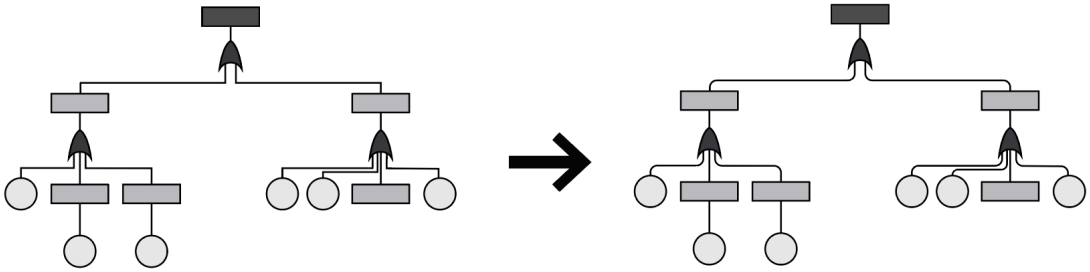


Figure 5.6: The Gestalt law of Continuity is applied, changing the sharp corner in the edges into smoother corners.

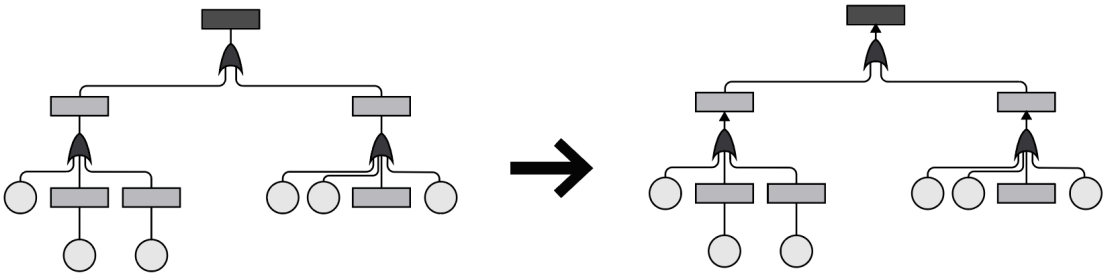


Figure 5.7: The arrows are added in order to solve the ambiguousness of lines in node-link diagrams.

1. What is the first impression of users when seeing the new layout?
2. How easily do users read the fault tree?
3. How easily do users interpret the fault tree?

Secondly, in order to find requirements for the final prototype, there two questions should be answered:

1. How do users picture the interactive fault trees look like?
2. What features do users want to have to make their reading experience easier?

Participants

Four participants were selected for the interview with the only requirement that they never worked with fault trees. Three out of four participants said that they never used or heard about

Metrics	Heuristic Methods
Node	Cluster similar nodes
	Distribute nodes evenly
	Keep nodes apart from edges
	Maximize node orthogonality
	Nodes should not overlap(except for nested nodes)
Edges	Minimize edge crossings
	Keep edge lengths uniform
	Minimize edge length (total and maximum)
	Minimize edge bends
	Keep edge bends uniform (angle/position)
	Maximize edge orthogonality
	Maximize minimum edge angles
Overall Layout	Maximize consistent flow direction
	Keep correct aspect ratio
	Minimize area
	Maximize convex faces
	Maximize global symmetry
	Maximize local symmetry

Table 5.1: The heuristic methods suggested by the study of (16) make node-link diagram easy to read and understand

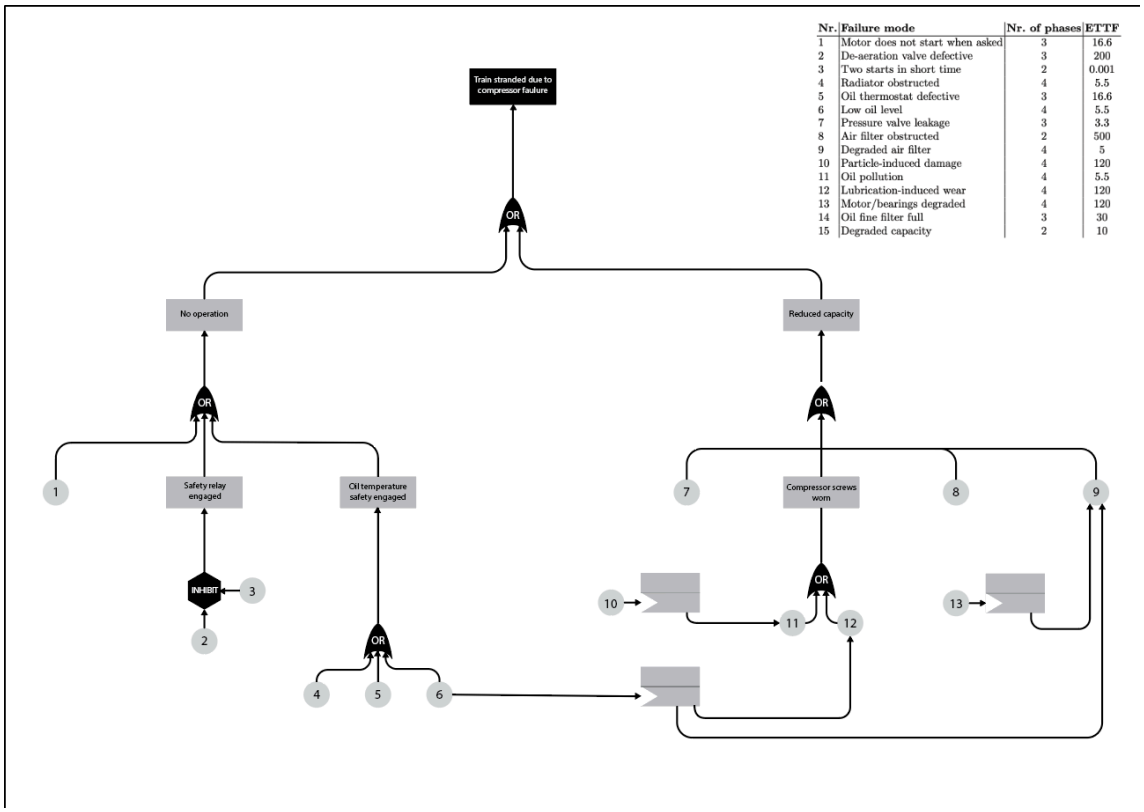
fault trees before, and one said she had heard about fault trees before, but never actively worked with them.

Setup

Since the interview was conducted online, there are only one laptop, the list of questions, and one phone to record needed. Each interview lasted around 20 minutes.

Interview Plan

The plan of the interview is described in this section. Before starting the interview, the interviewer introduces himself and explains the purpose of the interview. Next, the consent form is handed out to participants, and the important parts of the consent form are highlighted. In order to check on whether participants meet the requirement of the interview, participants will answer



Nr.	Failure mode	Nr. of phases	ETTF
1	Motor does not start when asked	3	16.6
2	De-aeration valve defective	3	200
3	Two starts in short time	2	0.001
4	Radiator obstructed	4	5.5
5	Oil thermostat defective	3	16.6
6	Low oil level	4	5.5
7	Pressure valve leakage	3	3.3
8	Air filter obstructed	2	500
9	Degraded air filter	4	5
10	Particle-induced damage	4	120
11	Oil pollution	4	5.5
12	Lubrication-induced wear	4	120
13	Motor/bearings degraded	4	120
14	Oil fine filter full	3	30
15	Degraded capacity	2	10

Figure 5.8: The new layout of the fault tree after be applied all the visual principles.

some questions about their background with fault trees. If they are not familiar with fault trees, the interview continues by showing them a very simple example of fault trees and a short introduction to the feature of fault trees. The interview starts with the first task: participants have 15 seconds to look at the layout of the fault tree design in Figure . When 15 seconds pass, participants answer questions in order that we find out their first impression of the layout. Some unscripted questions are encouraged to be asked so the interviewer can understand clearly what participants feel.

Then, in the next part of the interview, participants use the method called Thinking Aloud to finish the second given task. The task is to tell the interviewer their reading process while they are looking at the fault tree. Participants are required to give very specific information such as " Now I am looking at …" and " Then my eyes move to …". After that, there are some follow-up questions asked to find out their reading experience. The next part focuses on the fault tree interpretation of participants. In this section, interviewees need to answer some questions about interpretation. For example, they need to explain the relationship between two or three events connected with the gate. The interview should observe participants' reactions in order to know how confident and comfortable they give the answers.

Finally, the last part of the interview is designed in order to find the expectation of

participants on the interactive visualization of the fault tree. Participants suggest what features or any methods that would male their reading and interpretation experience become easier.

Interview Protocol

The full interview question list and their answers are reported in the Appendix 1

Analyzing Technique

The information collected from participants is recorded by video recording and interview notes. This information is then analyzed by a technique called Content Analysis (24). In particular, keywords and important hints from the answer of different participants are picked up and grouped together. Consequently, the differences and similarities between answers are identified.

Result

To guarantee the concrete of the report, the full answers of participants for each question are summarized, compared together, and reported in the Appendix 2. This sub-section only describes the result analysis of the lo-fi prototype in terms of Efficiency and Clarity and the new problem findings.

5.2.1 Lo-fi Prototype Evaluation

Efficiency

Half of the visual elements applied in the previous chapter got positive feedback from the interviewees while the rest is still causing some confusion. In the previous chapter, four visual elements (Color, Eye movement, Size, and Proximity) are applied to the layout in order to increase the efficiency of the fault tree. Eye movement and especially Proximity received good feedback from the interviewees. The large amount of white space created from Proximity helps interviewees easily jog around and divide the fault tree into different parts. However, some problems such as Color, Size, and amount of information still remain.

People would like to see more color introduced. The colors of different events and gates are coded in different gray-scale, as can be seen in Figure 5.8., which interviewees did not have a big problem with. However, they think more colors such as red and green could be applied as well to give them the signal of danger or safety. For some people, color is an important factor to attract attention and create a good impression.

Interviewees think the size of events and gates could be bigger. They were quite happy with the size uniform but some people had to zoom in to read the text. This action shows that the layout fails to give viewers a proper and easy reading experience.

The biggest remaining efficient problem is that there is still too much information displayed in the layout which made interviewees confused and overwhelmed. Even though some visual principles were applied, the layout still fails to prevent interviewees from having these feelings. It is a very serious problem because the goal of this project is to take away these feelings. So it is necessary to find the solution to this problem in further discussion.

Clarity

The interview acts as an evaluation of the use of three visual elements (Luminance contrast, Continuity, and Direction) and also identifies some new clarity problems. While two visual elements, Luminance contrast and Continuity, are well-applied, arrows at some edges cause little misunderstanding. For example, as can be seen in Figure , because event 3 is the conditional event of the INHIBIT gate, there should not be an arrow connecting this event with the gate. The edge connecting event 3 with the INHIBIT gate should be different from other edges simply because it shows a different relationship.

5.2.2 New Problem Findings

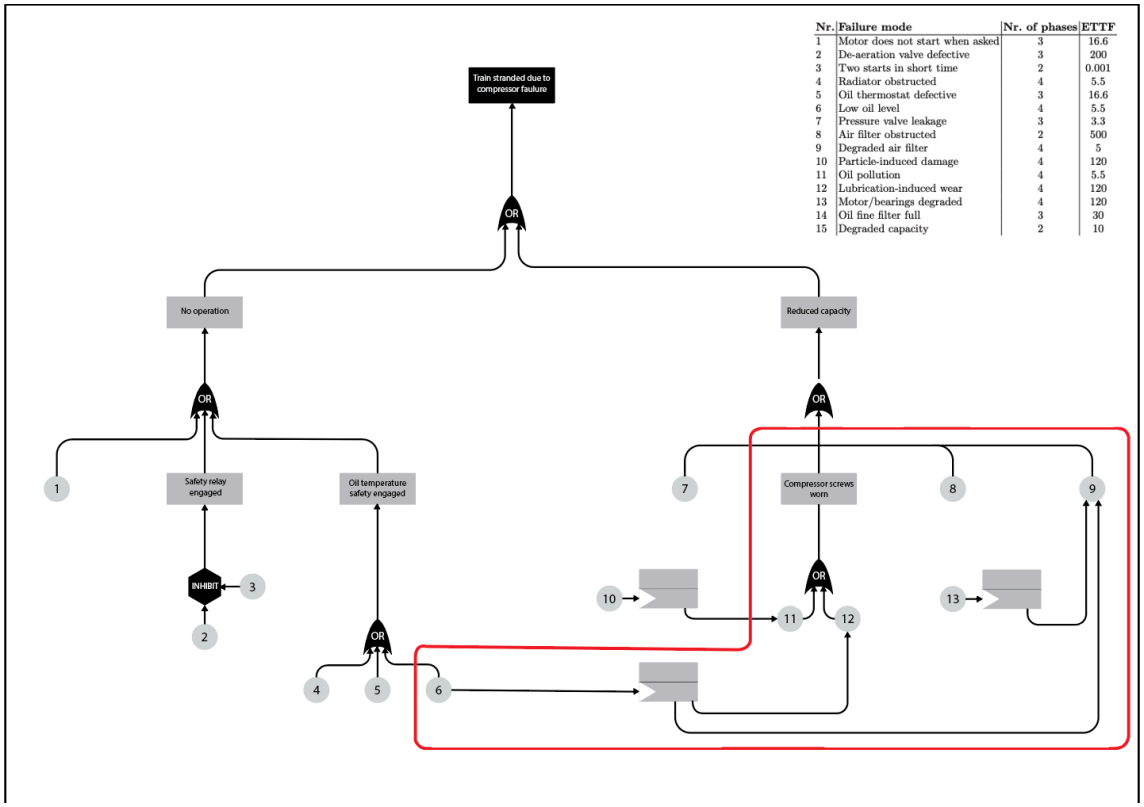
Visual problem: The Loop

The new visual problem that are found in the user interview is a loop which interrupts the reading flow of participants. a loop such as in Figure 5.9 breaks the natural reading flow of tree diagrams, causing a lot of confusion for viewers. The natural reading flow of tree diagrams is usually from the top going further to the bottom. People naturally pick the furthest left branch to read first and once that branch is finished, they start reading the branch on the right. However, if the branch connects from the left to the right, viewers would be very difficult to read further from there.

In particular, Figure 5.10 shows the order and direction of the reading flow among participants. It is interesting that all of them read the fault tree in the same order. One little note is that all participants read in the Western direction which means reading from the left to the right, and from the top to the bottom. Participants start reading the first three nodes at the top with the direction like in Figure 5.10. Then they discovered branch on the furthest left first. They finish the branch from the top to the bottom and then repeat with the next branch on the right.

One thing that is out of our expectation is that from node number 6, participants con-

tinue to read node numbers 12 and 9, which makes them all uncomfortable because the default three structure is broken and the unexpected loop was introduced. This loop interrupted the reading flow which can be seen by the hesitation of participants when they are trying to go forward. Finally, they stop the reading flow at node numbers 12 and 9, and start a new flow at “Reduce capacity”. The flow continues in the same order from the top to the bottom and from left to right.



Nr.	Failure mode	Nr. of phases	ETTF
1	Motor does not start when asked	3	16.6
2	De-aeration valve defective	3	200
3	Two starts in short time	2	0.001
4	Radiator obstructed	4	5.5
5	Oil thermostat defective	3	16.6
6	Low oil level	4	5.5
7	Pressure valve leakage	3	3.3
8	Air filter obstructed	2	500
9	Degraded air filter	4	5
10	Particle-induced damage	4	120
11	Oil pollution	4	5.5
12	Lubrication-induced wear	4	120
13	Motor/bearings degraded	4	120
14	Oil fine filter full	3	30
15	Degraded capacity	2	10

Figure 5.9: The unexpected loop makes viewers difficult to read.

Textual Clarity

Besides the evaluation of the visual elements, two new clarity problems related to texts are also addressed. First, without any explanation, viewers do not understand the meaning of gates, especially the one showing difficult relationships such as INHIBIT and FDEP gates. Second, the title of events sometimes use technical terms and are not descriptive enough for interviewees to understand. For example, the event named “No operation” is not very descriptive, it can be improved to be “ the air compressor does not operate”. The text of each event should have a complete meaning.

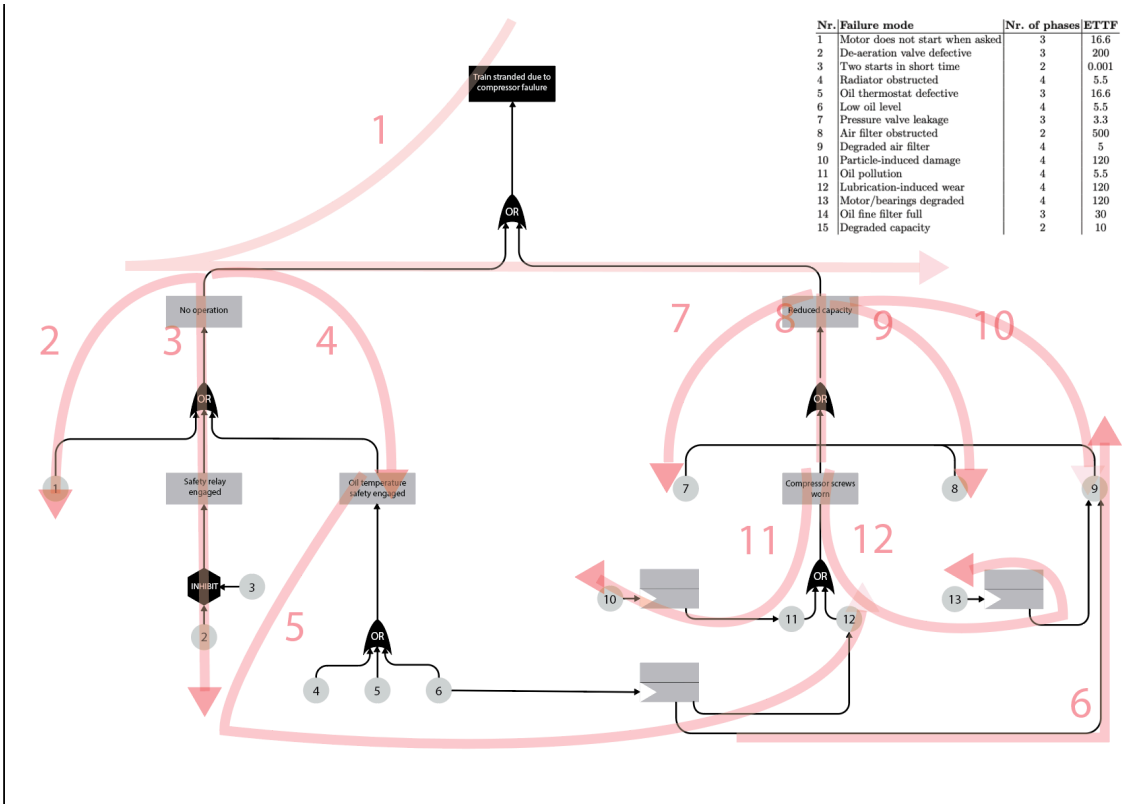


Figure 5.10: The reading flow of interviewees when they read the fault tree.

5.2.3 Improvements and further steps

After the interview, there are two further steps that are set. Firstly, visual elements such as Color, Size, etc should be adjusted based on the interviewees' feedback. In particular, the specific activities are listed here: More colors such as red and green will be applied in the next version of layouts. The size of the texts, gates, and events will be increased. The arrow and some edges will be changed in order to fit with the relationship between events. Trying to eliminate loops in the fault trees. Secondly, further research is conducted in order to find materials on how to solve the remaining problems through interactive visualizations. These remaining problems are the overwhelm caused by a large amount of information displayed on one screen, and the confusion caused by the lack of essential information and context. As can be seen in the previous sub-section, interviewees expect the interactive visualization to be able to solve these two problems which statics visualizations hardly can. Hence, in the next section, further research is made in order to find effective solutions for the two problems.

5.3 Interactive Features

In this section, research on how to use the advantages of interactivity to make fault trees more efficient and clearer is summarized. At the end of the previous chapter, it is concluded that interactive visualization should be able to fix the two problems that statics visualization hardly solves: manage to display a large amount of information on one screen without making users overwhelmed, and provide extra explanations on gates and events. So the knowledge gained in this section is used to find suitable features for our interactive fault tree visualization that help to solve the two mentioned problems.

This section has two sub-sections. The first one is Software Visualization which introduced the adjacent field with fault tree visualization. This sub-section shows the functional requirements of comprehensive software visualization tools which can be the reference to find the suitable features for our prototype. The second sub-section is Motion which introduces the benefits of using motions in displaying qualitative information. The knowledge learned from this sub-section can show us how to use motion to increase the efficiency and clarity of fault trees.

5.3.1 Software Visualization

The lack of related work to fault tree visualization leads to an exploration of the adjacent field: software visualization. Before designing an interactive fault tree visualization, it is necessary to review previous and related work in order to choose essential functions for the visualization application. Unfortunately, fault tree visualization seems to be a very new field, creating many difficulties to find the previous works. It is the reason why this project has to explore to some adjacent fields of fault tree visualization. After researching, it is believed that software visualization can be a good reference for designing fault tree visualization.

Software visualization and fault tree visualization share three similarities which make software visualization a good reference when designing fault tree visualization. Firstly, software visualization and fault tree visualization have the same goal of facilitating the understanding of viewers to complex model systems. Secondly, they both use visual effects such as graphic design, animation, and typography to visualize model systems. Lastly, because they are both in digital form, the ways users interact with the visualization are the same.

This section has three main parts. The first part describes the ultimate goal of software visualization: software comprehension. The second part describes the definition and the problem of current software visualization tools. Finally, the last part introduces nine functional requirements of comprehensive software visualization tools.

Software Comprehension

Even though the definitions of software comprehension vary among authors, they imply the same general idea. In each article talking about software comprehension, several different definitions are reviewed, but all refer to one similar meaning (25; 26). The most straightforward definition might be the one proposed by (27, p. 9): “software comprehension refers to the process of understanding and reasoning about software”. Similarly, the definition introduced by (28) has a similar meaning but more concrete: “the task of building mental models of the underlying software at various abstraction levels, ranging from models of the code itself to models of the underlying application domain”. Personally, the second definition is preferred because it is more descriptive than the first one. People can imagine, to some certain levels, the work of engineers when doing software comprehension.

Software comprehension is essential in software engineering activities, especially in development and maintenance tasks. According to (26), software comprehension is the core of all development, maintenance, reuses, migrates and reengineers software activities. He believes that understanding how the software works is the foundation that all software practitioners need to know. Additionally, (27) also believes that developers need to understand deeply various stages of the software development process such as source code, documentation, and execution log data, in order to perform their development and maintenance tasks effectively. Therefore, many researchers applied cognitive theories in software comprehension.

Two main cognitive models have been developed to aid in program understanding. Firstly, the top-down model proposes that developers understand software by first formulating the general hypothesis and then trying to prove or disprove it according to their further understanding. In detail, the initial hypothesis is first formed and refined into several subsidiary hypotheses which are evaluated later in a depth-first manner (29; 27). The verification of these hypotheses relies mainly on the absence or presence of the recognizable or familiar pattern in the code. Secondly, the bottom-up model is a totally reversed process. This approach starts with understanding the source code from small sections upwards to bigger sections (29; 27). (29) explains that code statements are read and grouped into higher-level abstractions. These abstractions continue to be aggregated further until the whole program is understood. While the top-down model is often used when the code or type of code is familiar, the bottom-down model is used more often in the situation when the knowledge of code is limited (29; 27). In several situations, programmers and engineers combine these two models together (26). Depending on their knowledge of the domain, they could switch between the two models.

To sum up, more broadly, software comprehension is the process of understanding big

complex model systems, which is very similar to the goal of fault trees: breaking down systems into smaller components and discovering the relationships of these components. The more software comprehension is understood, the clearer the similarity in purpose between software visualization and fault trees is revealed.

Software Visualization Tools

Software visualization (SoftVis) is understood as a tool to improve Software comprehension. Different from the diverse definitions of software comprehension, the definition of SoftVis introduced by (30) is trusted in many articles (29; 27; 31). (30) define software visualization as "the use of interactive computer graphics, typography, graphic design, animation, and cinematography to enhance the interface between the software engineer or the computer science student and their programs". Furthermore,(27) states that the goal SoftVis is to increase the effectiveness and efficiency of software comprehension. Similarly, (32) argues that SoftVis is used to provide insight and understanding and to simplify the existing software system.

However, many SoftVis tools do not seem to achieve the goal completely. Many researchers believe that there is a gap between the cognitive models in software comprehension (top-down and bottom-up models) and the way in which some existing SoftVis tools support these models (29; 27; 31). This gap can lead to some undesirable user experience (27). For example, in the research of (29), five most referenced SoftVis tools, including Jeliot, BlueJ, jGR ASP, X-ray, and JIVE, are reviewed under ten categories of visualization such as Structure, Code, and Methods call. The result drawn from this research is that none of the complete comprehensive software visualization is provided from these Softvis tools for professional software engineering. A similar result was concluded in the research of (33). In that research, a cognitive framework was designed based on the background of program comprehension theories and used to examine the cognitive models of program comprehension of SoftVis tools. The result addresses the issues of hierarchy among SoftVis tools. According to the examples, it can be concluded that designing a software visualization tool that supports software comprehension is not easy. Hence, identifying the requirements for a comprehensive software visualization tool is necessary

Functional Requirements of A Comprehensive Software Visualization Tools

In this section, nine requirements of a comprehensive software visualization tool are discussed. In which, three similar requirements (*View, Details on-demand, and Filter and Search*) and six different requirements (*Code proximity, Automatic layout, Undo/History, Select, Re-arrangement,*

and Comparison) are summarised from three researcher groups: (34; 35; 36).

Views: The first similar requirement proposed by all three research groups is Views. Views is the ability to provide different visual displays of the target history (34). Researchers agree that by providing consistent and integrate views, users gain insight into how different parts of a software evolve and change (34; 36). Also, by providing several views options, users can find a view that fits their goal the best (34). In the research of (34), multiple views can be shown in four different methods: based on different granularity levels of a system, abstraction levels, dependencies and relationships, and different types of visualization. Furthermore, in the different research of (35), many functions supporting Views such as Brows, Save, Navigation, View current focus, are considered to be "useful, but not essential."

Details on-demand: The second similar requirement is Details on-demand. (34) define this requirement as the ability to see more detailed information when needed. This requirement is defined similarly in the research of (36) but in a different name: Abstraction. According to (36), the abstraction mechanism is super useful when the results are too complex. It distributes the meaning of the results into various layers, allowing users to absorb results gradually and deeper if it is needed. There are several ways to display detailed information on demand. (34) suggest some popular user interaction techniques such as Zooming, Expand, and Mouse cursor hovers. For example, in some software visualization tools like *AniMatri*, using the Zooming-out function, users can change the scale of data overview.

Filter and Search: The final similar requirement is Filter and Search. In the research of (36) filter and search function are two separate requirements but in the research of (34), these two functions are combined into one requirement. However, both two researches agree on the definition and the importance of the two functions. Both of the two functions are used to control the data displayed under a specific condition (34). While filter allows users to see the data that satisfies a certain condition, search allows users to see the data that has similar text strings to what users type into the searching bar (34). Furthermore, Filter is an important function because it might reduce the dis-orientation effects by limiting the amount of data and analyzed areas that users are working on (34). Search is also considered to be one of the most useful functions, allowing users to find target data much faster (35).

On the other hand, there are six requirements that are suggested separately from the three research articles. (34) propose three requirements:

- **Select:** Select is the ability to select data sets in order to do some further actions with them such as facilitate, follow up, display the changes of large data over time, and identify locations of related data.

- **Re-arrange:** Re-arrange is the ability to change representation methods of software entities in new orders or to group them in different perspectives.
- **Comparison:** Comparison is the ability to compare the differences between two data sets.

Besides, the research of (35), and (36) seems to have an agreement on three different requirements:

- **Code Proximity:** Code proximity refers to the ability to provide easy and fast access to the original source code.
- **Automatic Layout:** Automatic Layout is the ability to show the result in the most suitable layout such as scales and highlight.
- **Undo/ History:** Undo/ History is the ability to revert to or show the previous states.

5.3.2 Features to increase Efficiency and Clarity:

First, the efficiency problem of showing too much information in one display can be solved by the two features: *View and Filter and Search*. Even though showing the different views of changes in data throughout the time is extremely useful for fault trees displaying quantitative information, in the scope of this project focusing on the qualitative information, we need to adjust this feature slightly. One of the possibilities is to provide views which are suitable for the different purpose of reading. For example, the interactive fault tree could give viewers an option of seeing the whole fault tree which suitable for having an overview or summary of the tree. Also, another option such as showing parts of fault trees could be useful as well for viewers wanting to look deeply into one part of the tree. The *View* feature can be even more advanced if it combines with Filter and Search feature which allows viewers to see the part of the tree that they are interested in. The *Filter* feature can also filter out all the irrelevant information on the screen that would distract viewers from focusing on what they are interested in. Consequently, viewers might read and understand the information easier and more efficiently. Furthermore, a useful feature that could be added is an *Automatic layout* which will choose the most suitable size and scale for viewers.

The second problem (lacking extra explanatory information) can be solved by the two features: *Details on-demand and Select*. Similar to software visualization, it is hardly possible for fault trees to display all the essential explanation information on one tree. So the feature allowing viewers to be more information on-demand can be very useful. They can provide only

the information that viewers needed and under the viewers' decisions. To be able to choose where and how to see the information, *Select* is the must-have feature. In the context of the digital world, *Select* is usually done by mouse operation such as click, double-click, and hover, or keyboard shortcuts. One of the opposite but indispensable features with *Select* is *Undo*. These two features will guarantee the easy and smooth usability experience of users.

5.4 Motion

5.4.1 Motion Application in Visualizations

Motion is a very potential visual element that is widely applied in many different kinds of visualization. This paragraph shows two common uses of motion in the visualization domain. Firstly, motion is often used to represent ecological and cognitive events: ecological events are the changes in layout or formation of objects; cognitive events are the behavior and motivational events of objects. Furthermore, motion can imply causality (37; 38). For example, Figure 5.11 shows when the blue ball moves towards the red ball and after two balls hit each other, the red one starts moving. People often understand that because of the hit between two balls, the red one moves, which is the causal relation. Secondly, motion is an effective coding mechanism (38). Designers often use motion to code orienting response, the changes over time, grouping elements, and attract attention. Moreover, unlike other visual elements, people can track movements and even predict the future movement of objects.

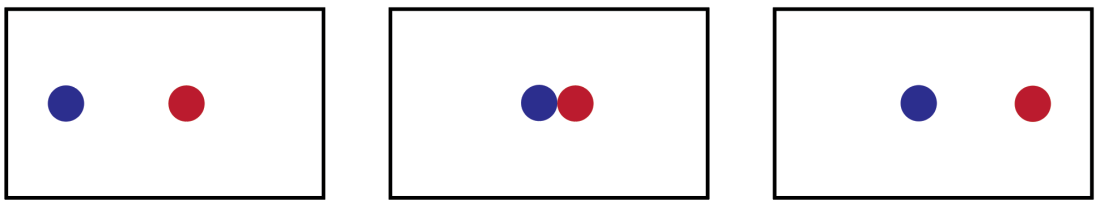


Figure 5.11: When the blue ball moves towards the red ball and after two balls hit each other, the red one starts moving.

Because motion is a great potential visual element, it is used widely in computer-based programs. There are eight common functions that often involve motion. These functions are collected in the study of three groups of researchers:

- *Awareness* is to provide to users the information which is not related to the current task that users are performing. The examples are alarms, messaging and peripheral awareness tools.
- *Transition* is the process of smooth movement between screens or states. For example, the common transition function are zooming and changing desktops in Mac
- *Functional Description* is the use of animated objects to describe the state of a process. An example can be seen often is the printer icon scrolling paper through when a print job is being executed.

- *Emphasis* is the use of motion to attract the attention of users to a specific place. An example is the blinking vertical bar that highlights where users are in the text while typing.
- *Expression* often use character-based animation or motion to provoke users' emotion or senses. An example can be an animated emoji.
- *Representation of Change* is often related to time-based behavior and how objects transform in a process in forms and sizes. For example, an animated pie chart shows the changes in different proportions over time.
- *Direct Visualisation* maps motion attribute such as speed, direction, and orientation to data variables. An example can be an animated line chart that has the line moving up and down based on the values of the data.
- *Association* uses motion to present the relationships of the group of objects. An example is the presentation of the union of the flock of birds.

5.4.2 Applying Motion in the Final Prototype

The use of motions in the Emphasis function seems to be one of the possible solutions for our efficiency problem. Because human eyes can be directed easily by motions, we could decide which piece of information viewers should see first based on its hierarchy. For example, the top event can have a shaking animation to grab viewers' attention. Another example is that our product can show the top event first and then show the bellowed events gradually one by one. In doing so, the large amount of information can be divided into smaller pieces and shown in the order that engineers design.

Since motion is a richly interpretive element, explanatory information can be represented by motions. For example, the Functional description function can be useful to present the causal sequence of events. For instance, an animated edge with an arrow starting from the basic event to the intermediate events and so on can show the causal relationship even clearer than the statics arrows. Another function that can be used is Expression. An example application can be that an animated smiley face placed above the top event will change into a sad face when one of the failures triggered. Furthermore, similar to the suggestion of an interviewee in the interview, explanatory information can be shown by a short animated video. For example, to demonstrate the process of one broken tire leading to a stopped car, an animation showing this process can be displayed.

6 SPECIFICATIONS

User and functional requirements of our prototype and a detailed prototype description are introduced in this chapter.

6.1 User requirements

The user requirements of an efficient and clear interactive fault trees are set based on the knowledge gained from the two types of interview. The first type is called User Interview which is reported carefully on section 5.2. From that interview, the requirements of the static layout and expectations of the interactive features from users are found. The second interview is called Expert Interview in which a fault tree expert was interviewed to find out the needs of project managers, one of the very common audiences of fault trees, when they look at fault trees.

6.1.1 User Interview

As mentioned in the previous chapter, the second half of the interview result is reported here. This section describes two big findings: different reading methods of users and .... The first findings show how users read information of fault trees which is very useful to design and specify the features of the final prototype later on. The second finding shows a summary of expectation of the final prototype on participant's perspective.

Different Reading Methods of Users

During the interview, the reading process among participants varies slightly, which show two different reading methods of users. When reading fault trees, some people have a quick overview before going to detail. They divided fault trees into different parts such as the top, middle, and bottom parts before reading. Some of them even compared the branches to find the logic of the tree. For example, all branches stop with the basic events represented by the circles. On the other hand, some people choose to read fault trees immediately in detail without having an overview. They explained that the overview does not help them to understand the tree better.

They prefer to read the text in the events to find the logic. They believed that if the text does not show the logic then that fault tree is not well-designed.

User expectation

The interview gave us a valuable opportunity to brainstorm with potential end-users on two subjects: their expectation on the interactive visualization and suitable features which they think would help them understand fault trees better. Firstly, most people agree that the interactive fault tree visualization should be very intuitive and easy to use. Their definition of an easy-to-use visualization can be summarized as the application allows users to easily navigate throughout it and should not introduce something surprising or unexpected. Users should instantly know what they have to do with the application. This information is very powerful for setting the usability requirements for our prototype. Secondly, interviewees also show their wished features which would make their experience with fault trees become much easier. There are five features that interviewees suggested:

1. A feature that allows users to see the extra explanation of events or gate on their demand, or more advanced, whenever they need.
2. Animations or images could be used to explain the complicated concept.
3. A feature that highlights only the related events.
4. A feature that can divide the qualitative information of fault trees into different layers such as hierarchy levels.
5. A filter feature that can filter out the irrelevant event or information.

Through these features, it can be seen that the expectation of interviewees on what problems the interactive visualization should be able to solve. In particular, through features 1,2, and 3, they want the interactive visualization to be able to inform them some more explanatory information and helps them navigate the tree better. Also, through features 4 and 5, they hope the interactive visualization to be able to solve the efficiency problem of having too much information in one display. These opinions are a valuable input for identifying the goal of our interactive fault tree visualization.

6.1.2 Expert Interview

An expert interview was conducted with the purpose of finding the user requirement from the managers' perspective. The expert interviewed is an engineer who has several years of working

with fault trees. It is taken into account that the interviewee is not a manager so her perspective could vary from actual managers. However, with all of her years working with fault trees, she should be qualified enough to know the needs of managers and the difficulties of making fault trees for them.

Results

The interview with the expert did not give us many new user requirements compared to the interview with non-experts, but it did give us some remarks about how managers work with fault trees. Firstly, the biggest difference between managers and non-experts is they already have ideas of what the fault tree is about. They have enough background knowledge about the project which supports them to understand fault trees better. They have a stronger need for going deep and further in details rather than seeing the broad and shallow overview. Hence, fault trees are often broken down into smaller pieces because of this reason. Secondly, managers often pay extra attention to the quantitative information which showing the biggest cause of the problem. In many situations, probability values should be presented also in fault trees. Besides that, managers also have a need to see fault trees in both overview and detail modes. They often ask for more information and explanation of one node or the relationship between that node and its children.

6.1.3 The needs of users

This sub-section summarized the needs of interviewees concluded from the interview results reported in both this section and the previous chapter and the expert interview. These needs focus on how users use fault trees and what the layouts should look like.

- The interactive fault trees must satisfy the two basic types of reading: reading overall (full fault tree) and reading in detail (some branches of the tree).
- The layout of the fault tree should have a default tree structure and avoid loops. The default tree structure starts at the top root and nodes connected further down. Nodes should not have links connecting from the left to the right to create loops.
- The interactive fault trees should guide viewers' eyes to read the tree more efficiently.
- The layout should have enough white space to help viewers to navigate throughout the tree easily.
- Nodes, texts, and gates should be big and clear enough to read.

- The interactive fault tree should manage and divide the tree into different layers, supporting efficient reading.
- The contrast of colors should be high, supporting the visibility of texts and backgrounds.
- Arrows should be used to demonstrate causal relationships.
- Edges should be straight lines or smooth lines with no sharp corners.
- Different relationships between nodes should be displayed in different types of edges.
- Complicated gates such as INHIBIT and FDEP gates should have extra explanations by text.
- Event titles should be descriptive and have a complete meaning.

6.2 Functional Requirements

The knowledge gained from the Interactive Visualization section and the User Expectation is the main contributors to come up with the featureal requirements of our interactive fault tree. The featureal requirement are listed below:

- The interactive fault tree should be able to provide two display modes which are suitable for the two kinds of reading: overview and read in detail.
- The qualitative information should be divided and presented based on hierarchy level.
- The fault tree should allow users to choose nodes and read more information of them.
- The fault tree should be able to filter out the irrelevant parts of the tree that users are not interested in.

6.3 prototype description

Our prototype is in the form of a web application, showing the interactive fault tree. The prototype can be analysed as two parts: layout and interactive features. The layout of the fault tree is designed carefully based on several visual theories to achieve clarity and efficiency. In particular, the layout should follow the following requirements:

- The layout of the fault tree should have a default tree structure and avoid loops.

- The layout should use preattentive visual properties such as color, size, or shape to guide viewers' eyes to read the tree more efficiently.
- White space should be used wisely in the layoutto help viewers to navigate throughout the tree easily.
- Nodes, texts, and gates should be big enough to read.
- They layout show follow the Gestalts laws:
  - **Connectedness**: nodes should be always connected together by edges
  - **Similarity:** there should be a uniform size between intermediate events and basic events
  - **Continuity**: Edges should avoid sharp corners.
  - **Proximity**: Nodes that are related to each other should be placed closer.
  - **Symmetry**: The layout should have a general look of symmetry.
- The contrast of colors should be high, supporting the visibility of texts and backgrounds.
- Color should be introduced based on clarity and the meaning of colors.
- The starting point should be emphasized.
- Arrows should be used to demonstrate causal relationships.
- Different relationships between nodes should be displayed in different types of edges.
- Complicated gates such as INHIBIT and FDEP gates should have extra explanations by text.
- Event titles should be descriptive and have a complete meaning.

The second part of the prototype is interactive features which are designed to support the reading process of users. Consequently, users can understand and work with fault trees easier. The list of interactive feature is described below:

View: The View feature allows users to switch between two display modes. The first mode is called Overview which shows the whole fault tree at once. On the other hand, the second display mode is Detail in which only the three events at the top of the fault tree are displayed. Then users are able to expand the tree further in the order that they want. These two does are designed to meet the different kinds of reading of interviewees. Also, in the Detail

mode, the information is broken down into different layers so it is easier for users to process the information.

Animation display: The animation display is a feature in the Overview mode. When fault trees appear in the Overview mode, the top event will appear first and then the animated edges moving downwards from the top event to the bellowed events. When the edge reaches the location of bellowed events, the bellowed events appear. The animation will go on until the last event appear. There are two benefits of doing so: firstly, the information appears gradually on the screen, preventing the overwhelmed feeling; secondly, the animation also can act as a reading flow guide for users.

Expansion button: Expansion button allows users to expand or collapse children of a node. Each intermediate event has one expansion button. It plays an important role in the Detail mode helping users to expand the tree as their desired order. The expansion buttons also give users the freedom to display only the nodes that they want.

Emphasis: the Emphasis feature is executed by a mouse click. If users click on a certain node, the chain of its parent and its direct children are highlighted in red and the rest of the tree is faded away. This way makes sure users can spend the most attention on the node that they are interested in. Plus, the sequence of failure is also highlighted, helping users to understand the sequence easier.

Comment/ Explanation: Comments or Explanations will appear every time users hover to any event. The comment explains clearly the relationship of the node to its parent. By doing so, users can choose what information they want to see more or what they do not understand.

Filter and Search (Optional): the interactive fault tree should have some filter options or a search bar. It would be very useful for complicated fault trees. Depending on the project, filter options can be varied. So users should be able to customize their own filter options. For example, some options can be based on different fields responsible for failures such as electrical engineering or mechanical engineering. Furthermore, a search bar should allow users to search wanted events by its key number or name. This information normally should be unique.

Explanatory Animation (Optional): In some cases, texts are not always the best presentation to introduce an explanation. Showing explanations by a short animated video is one of the alternatives. The animation is very useful to introduce a new concept or with general public who are not familiar with the topic of the fault tree.

7 REALIZATION

This chapter describes in detail the working process of the final prototype realization including both the initial plan and what changed afterward as well as the result of the realization. The realization can be divided into two parts: static layout and interactive features. The first part focuses on how to build up the static layout of fault trees which optimizes the effectiveness of visual principles and user feedback in order to increase the efficiency and clarity of fault trees. The second part introduces the Javascript library called GoJS, the main library used in this project, and the process of coding with GoJS.

7.0.1 Static layout

Initial plan

The plan for designing the fault tree layout has three steps. Step 1 is to design the layout in Adobe Illustrator. Then the layout are reviewed by experts. After having the reviewed layout, the last step is to transfer it into a web version.

Reality working experience

The actual working experience is relatively similar to the initial plan, expect the little change in the use of the Javascript library at the end. The first step of the plan was finished quite smoothly without running into any big problems. Also, because of some prior experience s gained from experience with lo-fi prototypes, the layouts were created in a very short amount of time. These layouts were then reviewed by some experts. There was much valuable feedback received such as: “ Dashed lines should not be used to describe the causal relationship because it might be confused with the default dashed lines used in FDEP gates“ and “The position of the gate should be placed closer to intermediate events because it is the default look of fault trees.” Also thanks to the feedback from users gained from the interview, there is no big confusion found in the layouts anymore. Finally, after that, the final layout was created, as can be seen in Figure …

After having the final layout, the next step is to draw them online. The first chosen JavaScript library is PIXI.JS. IT is a HTML5 creation engine, specifying for creating digital content. Some of the first efforts can be seen in Figure … However, after a short amount of time, it was realized that PIXI.JS is a very basic library which would take quite a long time to achieve what we were aiming for. Because of the restricted time, it is necessary to find another library that supports diagrams making. Then GoJS was found and chosen to be the main library used in this project. GoJS is a JavaScript library that specifies for building interactive diagrams and graphs. After discovering the new library, it took around a week to finish the online layout of the fault tree.

7.0.2 Interactive Features

Initial plan

The plan of coding interactive functions of the final product is quite simple: Try to program one function at a time and then integrate them together.

Reality working experience

Since the limited knowledge of programming, the actual working experience is quite difficult. The general coding experience can be described shortly as finding the example code or API with GoJS, adjust them as the wanted functions, and then integrate different functions together. However, some functions are built differently. For example, the Emphasis function is built by combining two functions Highlight and Hind. The Highlight function changes the stroke of the events and the edge to red while the Hind function reduces the opacity of the irrelevant events to 10The biggest challenge during the Realization stage is the working experience with animation. Since GoJs does not strongly support animation and every animation has to be built based on Math, It is super difficult to integrate it with the rest of the code.

7.0.3 Result

As described earlier, the final prototype has two parts: static layout and interactive features. Figure 7.1 shows the new layout of a provided fault tree. This layout is designed based on the visual requirements listed in section 6.1.3 in order to make an efficient and clear fault tree. Figure 7.2, 7.3, 7.4, 7.5, and 7.6 show how the prototype interface changes when users use the interactive features. In particular, Figure 7.2 shows the interface when the different view modes are chosen. Next, Figure 7.3 shows the interface when the Animation feature is shown. Figure

7.4 shows the interface when the Emphasis feature is used. Figure 7.5 shows the interface when the Explanation feature is used. Finally, Figure 7.6 shows the before and after interface when the Expansion Button is used.

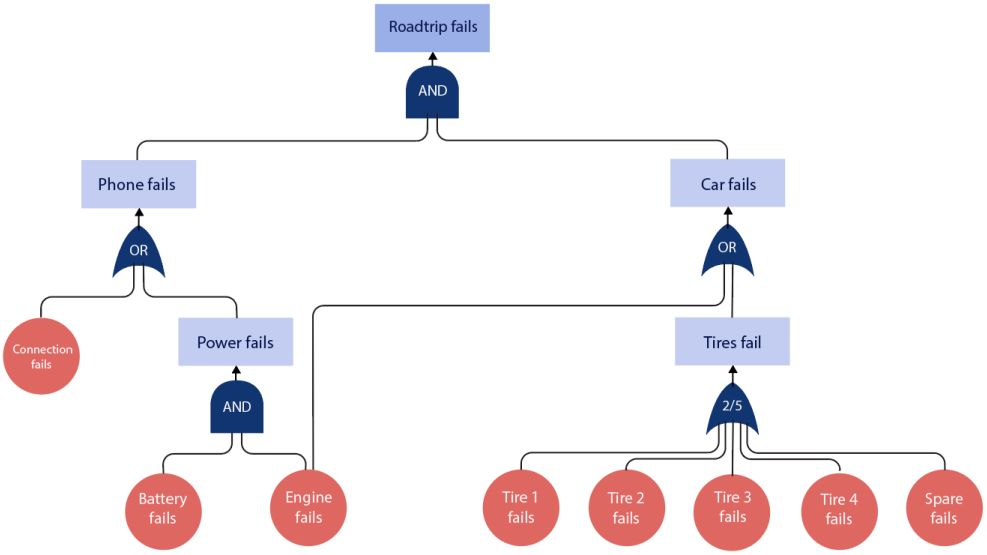


Figure 7.1: The static layout of the final prototype.

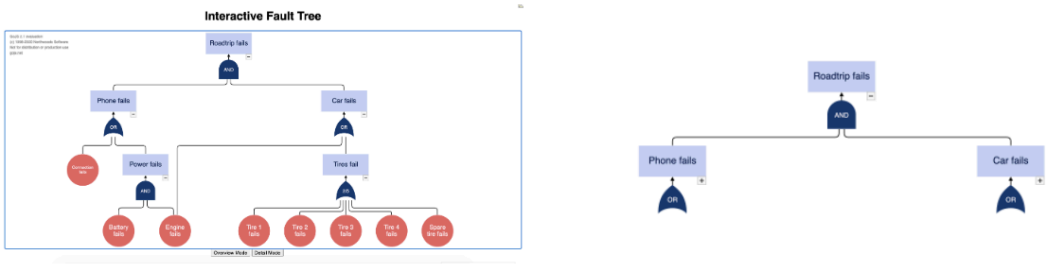


Figure 7.2: The left picture is the interface when the Overview mode is clicked and The right picture is the interface when the Detail mode is clicked

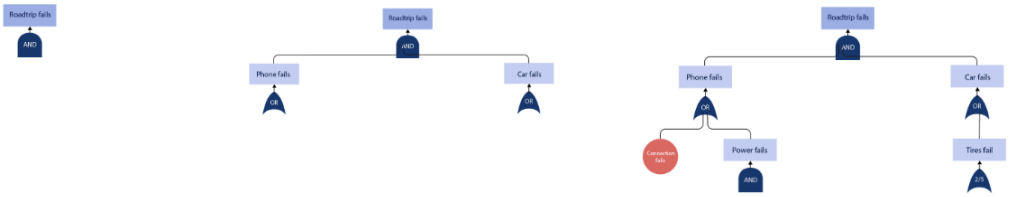


Figure 7.3: The interface is changing when the animation plays. Events gradually appear from the top to the bottom.

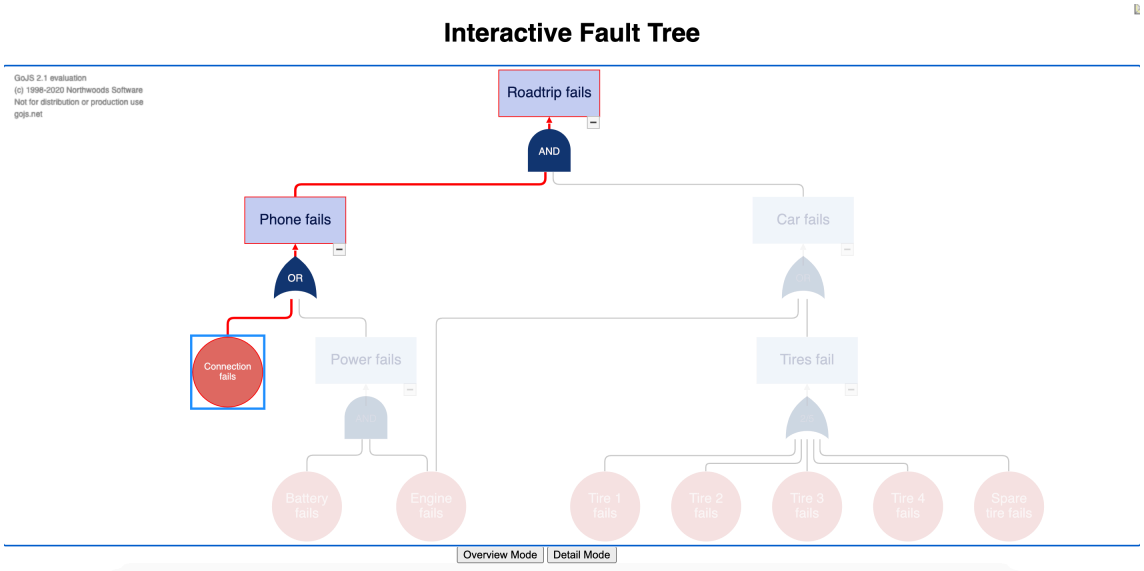


Figure 7.4: The interface when a ”Connection fails” event is selected.

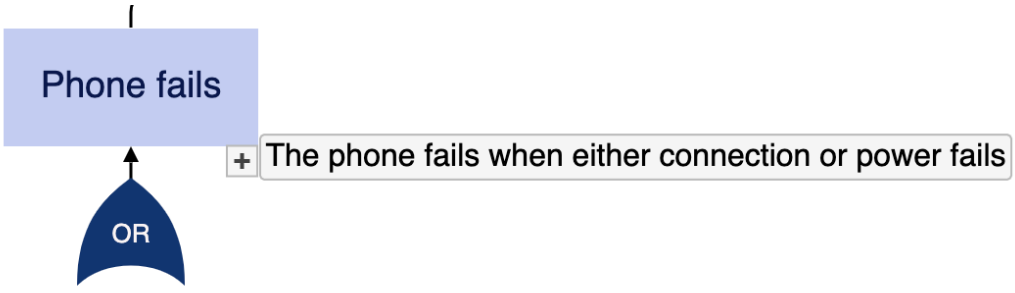


Figure 7.5: The explanation of the ”Phone fails” event shows up.

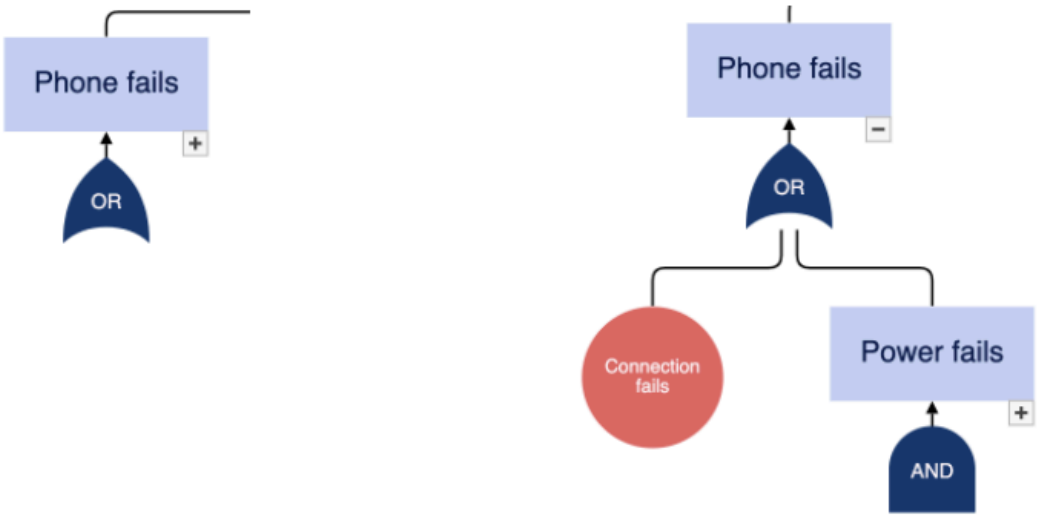


Figure 7.6: The left picture shows the ”Phone fails” event before click the Expansion Button, and the right picture shows the ”Phone fails” event after click the Expansion Button

8 EVALUATION

This chapter describes the evaluation on the final prototype that is made in the Realisation stage. There are two sections in this chapter. Firstly, the similarities and differences of the prototype described in the Specification stage and made in the Realisation stage are shown. This part is made by the self-evaluation of the reporter. Secondly, the user evaluation of the final prototype is collected from an evaluation interview. The interview conducted with seven participants who are non-experts with fault trees. Participants evaluate the final product in three aspects: user interface, effectiveness, and user satisfaction. The results reported in this chapter are valuable and the main inputs for the Discussion and Conclusion chapter.

8.1 Similarities and Differences between Realization and Specification stages

There are some similarities between the prototype made in the Realization stage and the plan described in the Specification. Firstly, the form of the prototype remains the same to be a web application. Secondly, the created layout meets all the visual requirements listed in the Specification and is transferred successfully online. Finally, there are three interactive features that are implemented completely: View, Emphasis, and Explanation. The three features are implemented exactly compared to the plan and receive no error.

On the other hand, there are some differences between the prototype and the plan in Specification, which mostly lay on the interactive features of the prototype. There are two features that are not implemented in the Realization: Filter and Search, and Explanatory Animation. The reasons for this absence are due to the limited realization time and the limited skills of programming. Also, there is one feature that is not implemented completely: Animation.

The Animation feature is completed partially compared to the plan. As mention in the Specification, the animation is planned to have all nodes that are gradually displayed on the screen and the animated edges that move from the gate to the top middle point of the children nodes. However, after the Realization, the animated edges fail to be implemented and only the gradual appearance of nodes is implemented successfully. It turned out that the animation of the edge is much more complicated than the expectation. The first intention is to create lines

that move along with the designed path but the plan did not work as expected.

8.2 Evaluation interview

The evaluation interview was conducted with seven potential end-users to evaluate the final prototype in three different aspects: user interface, effectiveness, and user satisfaction. The result of this interview will conclude the effectiveness of the final prototype to answer the research question. In this chapter, the results of this interview are divided into three parts: how the final prototype meet the user expectation of users found in the first user interview, the effectiveness of the final product on helping end-users understand fault trees easier, and the satisfaction of users on the interactive features.

Goal

The goal of the evaluation interview is to find the answers to these three questions:

1. How effective is the final prototype on increasing efficiency and clarity of fault trees?
2. To what extent, the interactive features are easy-to-use and intuitive?
3. To what extent, users are satisfied with the final prototype?

Interview plan

The evaluation interview was conducted with seven participants. The interview can be divided into two parts. In part 1, participants are asked to use the method Think Aloud to read two different layouts of the same fault tree as in Figure 8.1. Then some follow-up questions are asked to find out the effectiveness of the static layout. In the second part, first users answer some questions before using the interactive prototype to find out the expectation of users when seeing the product interface. Then they have two minutes to explore the interactive prototype themselves without any instruction and answer some follow-up questions. Their answer will give inputs in the usability, effectiveness, and user satisfaction of interactive features.

Analyzing Technique

Similar to the first user interview, the information collected from participants is recorded by video recording and interview notes. This information is then analyzed by a technique called Content Analysis (24). In particular, keywords and important hints from the answer of different

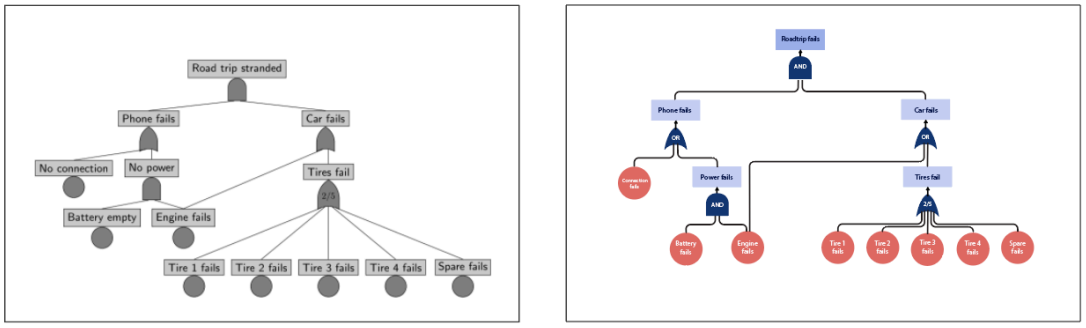


Figure 8.1: The left picture shows the old layout of the fault tree and the right picture shows the new layout of the fault tree

participants are picked up and grouped together. Consequently, the differences and similarities between answers are identified.

Result

8.2.1 The effectiveness of the final prototype

Static layout

The new layout shows many big improvements in the clarity of the fault tree. All seven participants feel the new layout of the fault tree, as can be seen in Figure 8.1, is much easier to read compared to the old layout. The reasons for this difference are mainly because of the way the events are organized and their colors. The positions of the events are placed horizontally equal which helps participants to easily divide events into different layers. The proximity is also applied to make the events that are irrelevant look even more separated and the events that are relevant look closer together. The difference in color also helps participants to navigate through the fault tree faster and visually easier. Furthermore, all participants could interpret the fault tree correctly without any prior instruction. They can recognize which events are the cause and which events are the reason, mainly because the texts are so clear and descriptive and the line with an arrow implying the relationship.

The new layout also shows some improvements in the efficiency of the fault tree. All seven participants can distinguish the different types of nodes such as intermediate events, gate, and basic events. Two visual elements contributing the most are colors and shape. Different from the old layout, the diversity of colors makes the different kinds of events and gate more distinguishing and visually clear. Additionally, the clear texts also create the logic flow helping participants read the fault tree easier.

Interactive Feature

The effectiveness of the five interactive features is reported in Table 8.1. During the interview, seven participants were asked whether they receive any benefit or drawback from the five features. In general, they receive the biggest benefit from the Expansion button and Emphasis feature and do not receive any benefit from the Animation feature. Participants also pointed out some drawbacks of the View and Animation feature. The number in front of each line is the number of participants giving that benefit or drawback.

Features	Benefits	Drawback
View	(4/7) participants feel beneficial to have 2 different view modes which are suitable for 2 different reading purposes: overview and reading events one by one.	(5/7) participants feel there are not many differences between the Overview mode and Detail mode. They expected to see more information of events in the Detail mode.
Emphasis	(7/7) participants agree that this feature improves their focus on the branch that they are interested in.	No drawback received
Animation	(2/7) participants feel that the animation guides their eye to read from the top to the bottom.	(5/7) participants do not receive any benefit from the animation. (3/7) They get lost after one or two events appear. Many events showing up at the same time are super confusing for them.
Explanation	(5/7) participant believe this feature will be more beneficial in **other more complicated fault trees** with complicated gates	No drawback received.
Expansion Button	(7/7) participants receive benefits from being able to expand or collapse the fault tree as their wish.	No drawback received.

Table 8.1: The interview result of the effectiveness of interactive features

8.2.2 User Interface

A part of the evaluation interview was a usability test. It is because, as mentioned in section 6.1.1, the expectation of users on the interactive prototype is that the prototype needs to be easy-to-use and intuitive. In another word, the prototype needs to have high usability. So the usability test was conducted to evaluate how well the interactive prototype meets this expectation of users.

The result of the usability test shows that the interactive prototype generally has a poor user interface. This is a big disadvantage of the interactive prototype when the goal of the project is to make an easy-to-understand fault tree but the interactive features are not easy-to-use. This opens for the improvements in further work. The full result of the usability test is listed below:

- **Web interface:** The initial interface of the prototype has three elements: the big title “Interactive fault tree” and two buttons named “Overview mode” and “ Detail mode”, as can be seen in Figure8.3. All seven participants agree that there are not any excessive elements. However, 5/7 participants think that a short introduction of the product is missing. Since this kind of prototype is new for them, they would like to have a short instruction within one sentence, such as “ Please click to one of the buttons below to see the interactive fault tree.” Additionally, 4/7 participants pointed out that the position of the title and the buttons should be symmetrical and placed exactly at the center of the screen.

- **View:** 6/7 participants expected to see the fault tree appear in the center of the screen and below the two buttons instead of above the buttons like in Figure 8.3. On the good side, all seven participants found these buttons are clickable and expect to see two different kinds of fault trees after clicking the buttons. 6/7 participants think that this feature is relatively new for them and they do not have any negative surprise about this feature.

- **Explanation:** 3/7 participants could not find this feature by themselves. And 3/4 participants who found this feature believe that they can only see this feature when they hover over the expansion buttons instead of the whole event as it is designed. Also because of this mistake, this feature is a bit out of their expectation. Since the explanation is about the relationship between two nodes through the gate, it is more logical for the explanation to appear when users hover over the gate, instead of the whole node like now. Furthermore, 4/7 participants think the explanation appears too slowly. They think this information is very important so they expected to see it right the way after hovering.

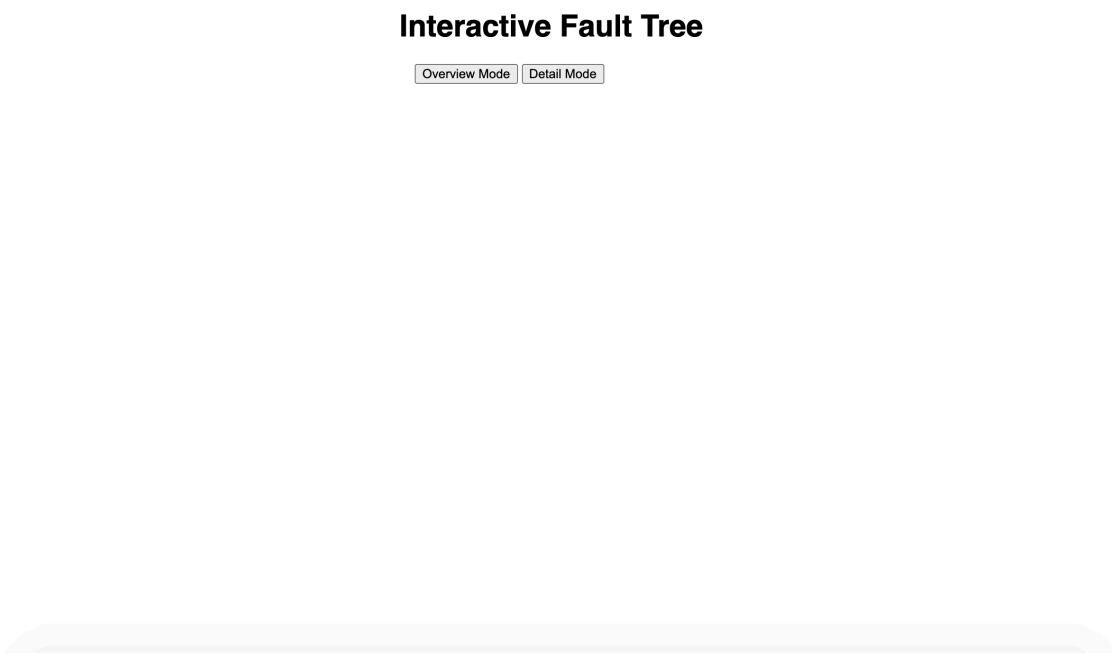


Figure 8.2: The initial web interface includes one big title and two buttons

- **Animation:** All seven participants recognized there is a short animation in the beginning when they click the Overview mode button. 6/7 participants think that they already saw this kind of animation before, especially during presentation and presentation talking about a process. They also do not feel displaying items with the animation is something out of their expectation.

- **Emphasis:** More than half of participants (4/7) could not find this feature on their own. 2/3 people who found this feature, they found it unintentionally. This means that they do not expect that the events are clickable and they will receive any feature if they click them. 4/7 participants have not seen this kind of feature before so it is relatively new and it does not create any un-expectation for users.

  There are two big interface mistakes with this feature that are found by participants. The first mistake is when participants clicked the “Engine fails” event, the route leading to the “Car fails” was not highlighted as it should be. The mistake can be seen in Figure 8.4. The second mistake is the wrong fault propagation with the AND gate. For example, the left picture in Figure 8.5 shows if participants click on the “Connection fails” node, all its parent and grandparent are coded in red. This implies that if the connection fails, the phone fails and the road trips also fail. However, it is not true because the road trip only fails when both the phone and the car fail. So the correct interface should look like the right picture in Figure 8.5. The road trip event should not be colored in red.

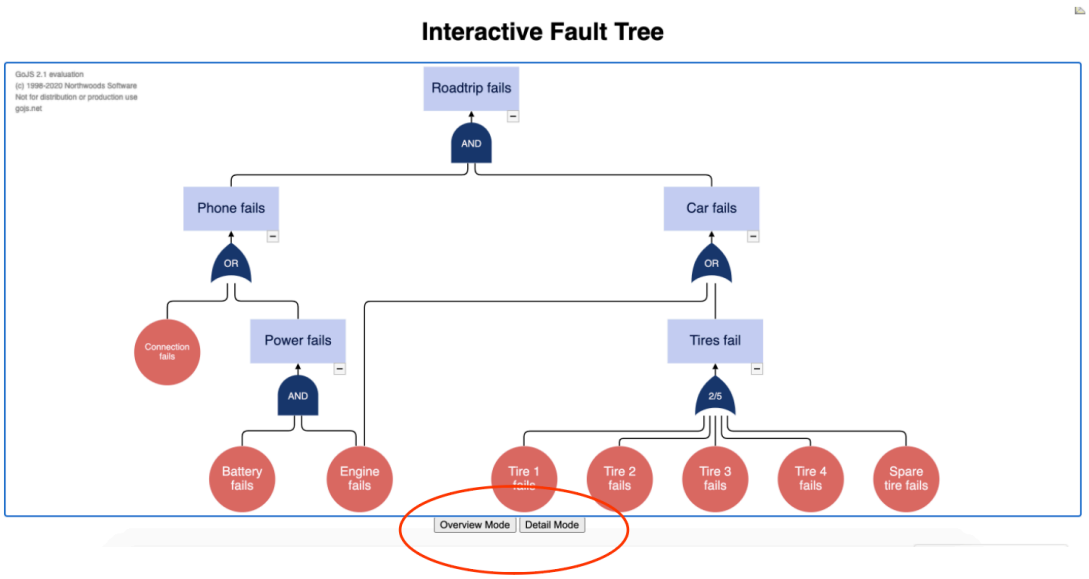


Figure 8.3: Participants prefer to see the fault tree below the buttons, not above the buttons

- **Expansion Button:** The expansion button is the only feature that has a good interface when all people found this feature and have very good feedback on it.

8.2.3 User satisfaction

The third result collected from the evaluation interview is user satisfaction. Our interest focuses on knowing which interactive feature brings the most and least satisfaction to users. Consequently, the direction for future improvement is identified. To pursue our interest, we used a model called the Kano model. This section describes briefly how the Kano model works and how our interest is answered by this model.

Kano model

The aim of the Kano model is to connect the features of a product with user satisfaction and identify four different types of features that influence user satisfaction (39). Figure 8.6 shows these four types of features:

- **Must-have:** The must-have feature is the most basic feature of a product. It implies that users will not be satisfied with the product if there is no feature like it. However, this kind of feature does not gain additional user satisfaction beyond a neutral level.
- **One-dimensional (Performance):** This kind of feature is connected directly with the satisfaction of users. That means the more these features are introduced, the higher the

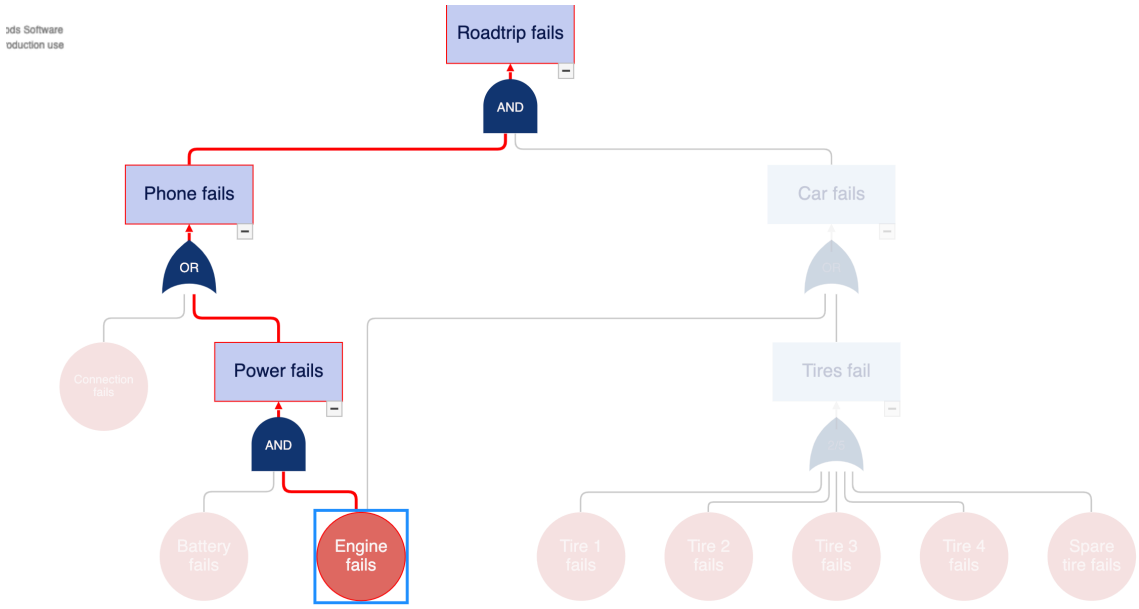


Figure 8.4: Mistake: When participants clicked the “Engine fails” event, the route leading to the “Car fails” was not highlighted as it should be

satisfaction users have for this product. This kind of feature should have high priority while designing a product.

- **Attractive:** This kind of feature indicates a feature that gains more user satisfaction when it is functional but does not reduce user satisfaction if it is less functional. This kind of feature is neither explicitly expressed nor expected by the users.
- **Indifferent:** This kind of feature does not affect the level of user satisfaction at all even though it can be either very functional or not very functional.

The biggest benefit of the Kano model is to easily convert the qualitative answer of users into quantitative data and from there, researchers can understand which type of feature in their product have a greater influence on user satisfaction(). This is also the reason why Kano model is chosen to be used in this project.

Expected Result

When designing the five interactive feature, our aimed type for each feature are: View, Expansion Button, and Animation are attractive features, Emphasis is a one-dimensional feature, and Explanation is a must-have feature. The result of this evaluation will show the comparison between what users think and what we are expected.

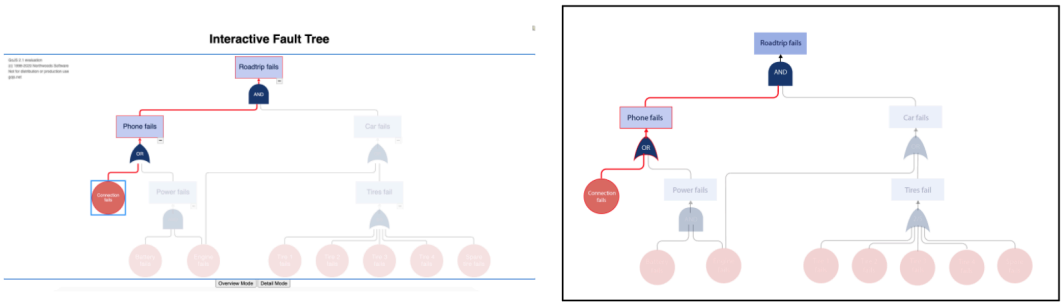


Figure 8.5: The AND gate mistake is shown in the left picture when the ”Road trip fail” event is highlighted in red, the correct interface is shown in the right picture.

Methodology

The inputs for the Kano model are collected from the pair questions in the questionnaire, during the Evaluation interview. In particular, participants have to answer a pair of questions look like this:

- How do you feel if there is a feature like it?
- How did you feel if there was no feature like it?

There are five possible answers to each question:

- I like it.
- I expect it.
- I am neutral about it.
- I can tolerate it.
- I dislike it.

These answers of participants then are analyzed in two steps. In step 1, in order to classify the type of the feature, each pair of answers of participants will be mapped into the table in Figure 8.7. For example, if a participant likes the existence of the Emphasis feature and dislikes when the Emphasis feature does not exist, then this feature will be classified into the Attractive type. There are two new symbols in the table which are Reverse(R) and Questionable (Q). When the result is Reverse type, the result of functional and dis-functional will be reversed. When the result is Questionable (Q), the result does not have any meaning. After analyzing

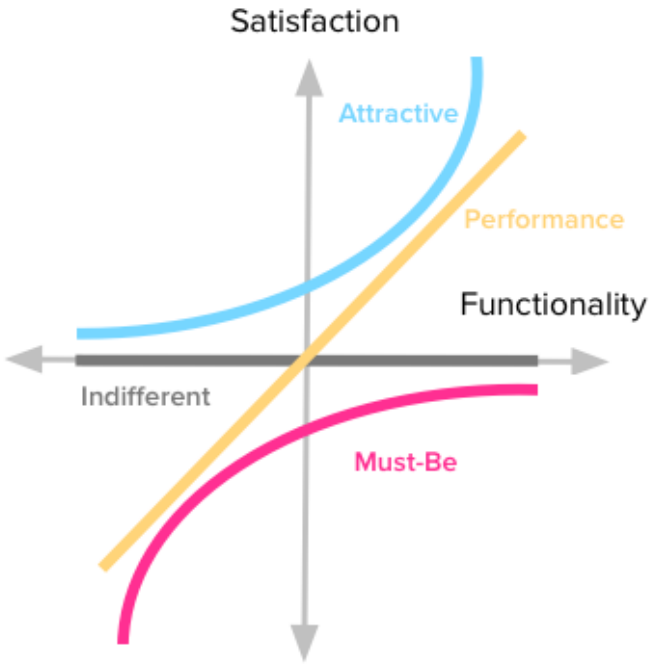


Figure 8.6: Kano model graph shows four types of features:Must-have, One-dimensional (Performance), Attractive, Indifferent.

all answers of all participants, the number of types will be counted and the type that has the highest number of participants will be the type of the feature.

In step 2, in order to understand the influence of the features on user satisfaction deeper, the result of step 1 will be calculated as in Figure 8.8. The satisfaction coefficient describes the level of satisfaction of users if the feature exists, ranging from 0 to 1. The dissatisfaction coefficient describes the level of dissatisfaction of users if the feature does not exist, ranging from 0 to -1.

Result

The results of the Kano model are shown in two tables and one figure. The table 8.2 shows the result of step 1, concluding the type of each feature. The table 8.3 and Figure 8.9 show the satisfaction and dissatisfaction coefficient of features.

According to Figure 8.9,only the Expansion button feature receives the expected result which is Attractive while the rest receives lower results than expected. Even though the Emphasis function also contributes strongly to user satisfaction, it is not truly functional that makes users dislike if the feature did not exist. The View and Animation feature do not make users satisfied enough to be an attractive feature. Finally, the Explanation feature is not functional

Functional (feature present)		Dysfunctional (feature absent)				
		Like it	Expect it	Don't Care	Live With	Dislike
	Like it	Q	A	A	A	P
	Expect it	R	I	I	I	M
	Don't Care	R	I	I	I	M
	Live With	R	I	I	I	M
	Dislike	R	R	R	R	Q

Figure 8.7: Kano evaluation table includes attractive (A), performance (P), Must-have (M), Indifferent (I), Reverse (R), Questionable (Q)

$$\text{Enhanced Satisfaction Coefficients} = \frac{A+O}{A+O+M+I}$$

$$\text{Reduced Dissatisfaction Coefficients} = \frac{O+M}{A+O+M+I}$$

Figure 8.8: Two formulas calculate satisfaction and dissatisfaction coefficients

enough to be a must-have feature.

Features	Indifferent	Attractive	Must-have	Performance	**Type**
View	4	0	3	0	**I**
Explanation	6	0	1	0	**I**
Animation	4	2	1	0	**I**
Emphasis	1	4	0	2	**A**
Expansion Button	0	4	0	3	**A**

Table 8.2: The table shows the type of the five features

Features	Types	Satisfaction Co.	Dissatisfaction Co.
View	I	0.2	-0.1
Explanation	I	0	-0.1
Animation	I	0.4	-0.2
Emphasis	A	0.85	-0.3
Expansion Button	A	1	-0.4

Table 8.3: The table shows the types, satisfaction coefficient, and dissatisfaction coefficient of features

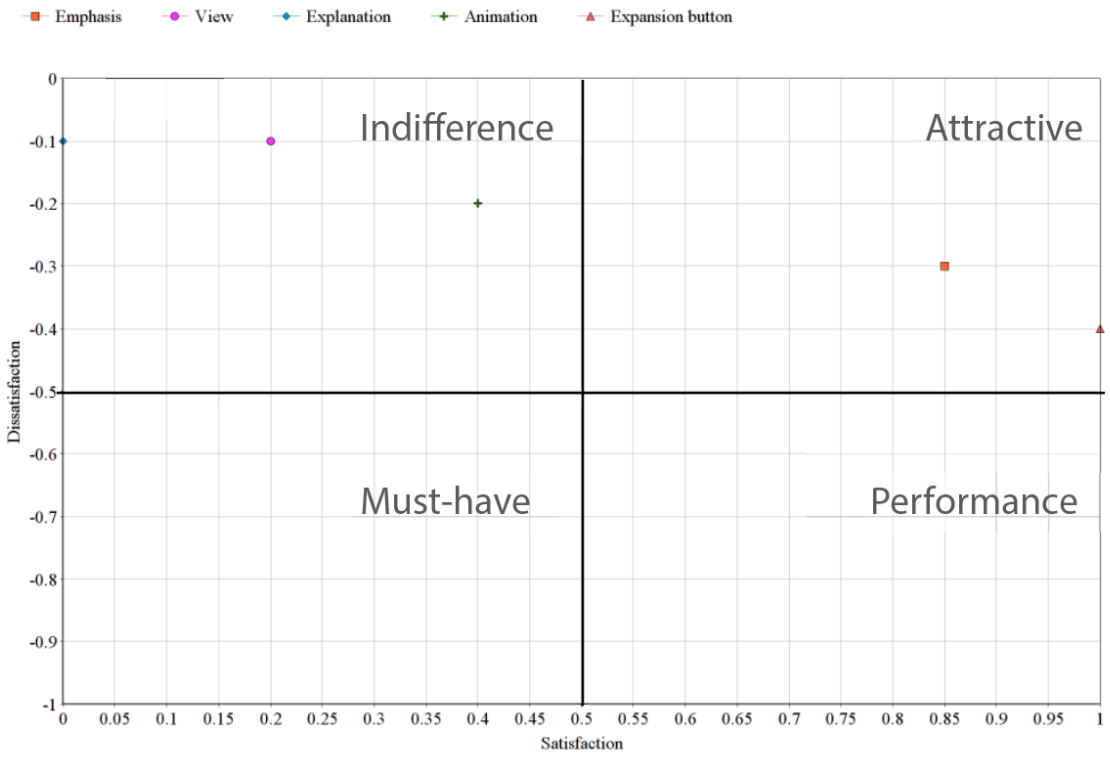


Figure 8.9: Satisfaction-dissatisfaction diagram

9 DISCUSSION

9.1 Limitations

One of the shortcomings of the study is that it only involved a limited amount of user research with experts. This could lead to several problems. Firstly, a limited understanding of how experts design fault trees causes a chance that experts could not apply the result of the study to their work. Secondly, the study did not compare how experts and non-experts read fault trees. Thus, we do not know whether experts read fault trees differently from non-experts. This difference could be part of the reason why fault trees designed by experts are difficult to understand for non-experts.

Another shortcoming of the study comes from the evaluation method. Firstly, the study only found that the final prototype helps non-experts to distinguish different events more efficiently but did not find if it increases the speed non-experts reach their desired information faster. Secondly, since the final prototype was built on a relatively simple fault tree, the effectiveness of the interactive feature could be different when applied to more complicated fault trees.

9.2 Further work

There are three directions that are recommended for future work. First, more evaluation tests could be conducted. For example, an evaluation of whether the final prototype increases the speed of finding the desired information among users can be made. This evaluation will contribute valuable evidence on the effectiveness of the final product to increase the efficiency of fault trees. Furthermore, it is recommended to apply the results of the study on different fault trees to evaluate its applicability in different cases. Another example is the evaluation with experts to check the consistency and accuracy of the final product. By doing so, it is guaranteed that the final prototype will introduce a correct method of reading fault trees to non-experts.

Second, future work could be done to improve the user interface of the implemented interactive features in this study. To be more specific, users expected to see more differences between the Overview mode and Detail mode. One way to do so is to include more information

about events on the Detail mode. Furthermore, the Animation feature could be improved by showing each event one by one in a correct reading flow and at a higher speed. The animation should be simple, exciting, and have clear benefits. Next, the interface error of the Emphasis feature should be fixed and the nodes should look more clickable. Finally, the Explanation feature should appear faster and only appear when users hover over the gates.

Finally, the two optional features which are Filter/Search, and Explanatory Animation could be implemented in the future work. Both of these features show potential effects on bigger and more complex fault trees. For example, Filter/Search could be very useful for fault trees having a large number of basic events. Moreover, the Explanatory Animation feature could be useful when new concepts or topics are introduced to non-experts. The effect of this feature can be even bigger if the topics are complex and hard to imagine such as machine structure or complex infrastructure.

10 CONCLUSION

In many technical fields, fault trees are often used to visualize the risk of models. However, they are often difficult-to-understand for non-experts. This study was conducted in order to find the root of this problem and find solutions that are presented by an interactive prototype. The value of this study is mainly for experts who want to make non-expert friendly fault trees and for organizations that want to create their own fault trees guidelines.

Fault trees are often difficult-to-understand for two reasons. Firstly, fault trees are not always efficient at helping users to read each branch of the tree. Normally, users read fault trees in two ways: glancing at the entire tree to have an overview and reading every event in each branch one by one. Fault trees are only efficient in the first way but they are often too big, complicated, and distracting in the second way. Furthermore, the layout of fault trees does not always show the proper visual hierarchy. The fault tree in Figure .. can be an example of this. Secondly, the clarity of fault trees is also often missing. The study found that not only visual clarity but also textual clarity affect how easy non-experts understand fault trees. The visual clarity of fault trees is often created by visual elements and visual principles such as color, shape, proximity while the textual clarity of fault trees is created by the clarity of event names and the gates' meanings.

The solutions for these two problems were researched in this study and presented by the interactive prototype in Figure …. The solutions are also the answer to the research question:" What is the most effective display method that makes fault trees easier-to-understand for non-experts?"

Firstly, to make fault trees easier-to-understand, the efficiency of fault trees needs to be increased. In this study, the efficiency of fault trees is increased by creating a new layout that meets the requirements of visual hierarchy and developing four interactive features. In particular, the new layout shows each type of event and gates with different colors, which increases their distinguishment. Furthermore, four interactive features developed to increase efficiency are View, Emphasis, Animation, and Expansion Button. The View feature allows users to have two different view modes of fault trees: Overview mode and Detail mode. The Emphasis fea-

ture allows users to highlight the selected events and their parent chain and hide the rest of the tree. The animation is one small feature played in Overview mode, showing the whole fault tree from the top to the bottom. Finally, the Expansion Button allows users to expand or collapse the children of any intermediate events.

Secondly, to make fault trees easier-to-understand, both visual and textual clarity of fault trees needs to be increased. The layout designed in this study is an example of showing how to use visual elements and principles to increase the clarity of fault trees. Some of the main elements can be listed as color, shape, and proximity. Furthermore, to improve the textual clarity, an interactive feature called Explanation has developed. The Explanation feature allows users to see the explanation of the relationship between certain events and their children.

At the end of the study, an evaluation interview was conducted to see how effective the prototype is in making fault trees easier to understand for non-experts. The result of the evaluation can be summarized as that users receive some significant benefits from the new layout and some interactive features. With the new layout, users can read the fault tree in the correct order and much easier. Also, they can interpret the fault tree accurately without initial instruction. The two features that show the greatest benefits and gain a lot of user satisfaction are Expansion Button and Emphasis. While the Expansion Button feature allows users to only look at the events that they are interested in, the Emphasis feature allows them to focus on the selected branch better. On the other hand, users hardly receive any benefit from the Animation and Explanation feature. The evaluation also shows that the user interface of all five features also needs to be improved.

REFERENCES

[1] M. J. Eppler and M. Aeschimann, “A systematic framework for risk visualization in risk management and communication,” *Risk Management*, vol. 11, no. 2, pp. 67–89, 2009.

[2] A. Alaszewski, “Risk communication: identifying the importance of social context,” *Health Risk Society*, vol. 7, no. 2, pp. 101–105, 2005.

[3] E. Ruijters, D. Guck, P. Drolenga, M. Peters, and M. Stoelinga, “Maintenance analysis and optimization via statistical model checking,” in *International Conference on Quantitative Evaluation of Systems*. Springer, 2016, pp. 331–347.

[4] E. Ruijters and M. Stoelinga, “Fault tree analysis: A survey of the state-of-the-art in modeling, analysis and tools,” *Computer science review*, vol. 15, pp. 29–62, 2015.

[5] L. Waghmode and R. B. Patil, “An overview of fault tree analysis (fta) method for reliability analysis,” *Journal of Engineering Research and Studies*, vol. 4, pp. 6–8, 2013.

[6] R. B. Patil, D. A. Mhamane, P. B. Kothavale, and B. Kothavale, “Fault tree analysis: a case study from machine tool industry,” *An International Conference on Tribology*, 2018.

[7] Y. Xu, F. Gao, W. Chen, Z. Liu, and P. Gu, *Nuclear Power Plants: Innovative Technologies for Instrumentation and Control Systems*. Springer, 2017.

[8] D. Vernon-Bido, A. J. Collins, and J. A. Sokolowski, “Effective visualization in modeling & simulation.” in *Simulation Series*, vol. 47, 2015, pp. 33–40.

[9] J. Steele and N. Iliinsky, *Beautiful visualization: Looking at data through the eyes of experts*. O’Reilly Media, Inc., 2010.

[10] A. Quispel, A. Maes, and J. Schilperoord, “Aesthetics and clarity in information visualization: The designer’s perspective,” in *Arts*, vol. 7, no. 4. Multidisciplinary Digital Publishing Institute, 2018, p. 72.

[11] A. V. Moere and H. Purchase, “On the role of design in information visualization,” *Information Visualization*, vol. 10, no. 4, pp. 356–371, 2011.

[12] M. Kemal, “Great visualization essentials -what makes a good data visualization?” 04 2019.

[13] J. Wu, “Hierarchy theory: an overview,” in *Linking ecology and ethics for a changing world*. Springer, 2013, pp. 281–301.

[14] E. Kleiberg, H. van De Wetering, and J. J. Van Wijk, “Botanical visualization of huge hierarchies,” in *Symposium on Information Visualization*. IEEE, 2001, pp. 87–94.

[15] N. H. Müller, B. Liebold, D. Pietschmann, P. Ohler, and P. Rosenthal, “Hierarchy visualization designs and their impact on perception and problem solving strategies,” in *Proceedings of the International Conference on Advances in Computer-Human Interactions*. IEEE, 2017, pp. 93–101.

[16] C. Bennett, J. Ryall, L. Spalteholz, and A. Gooch, “The aesthetics of graph visualization.” *Computational aesthetics*, vol. 2007, pp. 57–64, 2007.

[17] C. Ware, *Information visualization: perception for design*. Morgan Kaufmann, 2019.

[18] E. Wegman and Y. Said, “Color theory and design,” *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 3, no. 2, pp. 104–117, 2011.

[19] S. Bhowmick, “The rgb rendering of visible wavelength lights,” 01 2017.

[20] S. Bianco, F. Gasparini, and R. Schettini, “Color coding for data visualization,” in *Encyclopedia of Information Science and Technology, Third Edition*. IGI Global, 2015, pp. 1682–1691.

[21] A. Albustin, S. Bacheitner, A. Djerdjizi, and B. Hollerit, “Pre-attentive processing,” *Course material. Graz University of Technology*, 2010.

[22] R. Hunt and S. HF700, “Preattentive processing.”

[23] D. Holten and J. J. Van Wijk, “A user study on visualizing directed edges in graphs,” in *Proceedings of the SIGCHI conference on human factors in computing systems*, 2009, pp. 2299–2308.

[24] J. Lazar, J. H. Feng, and H. Hochheiser, *Research methods in human-computer interaction*. Morgan Kaufmann, 2017.

[25] K. M. Nur and H. Sarwar, “Software visualization tools for software comprehension,” *Conference: The 4th International Conference on Software, Knowledge, Information Management and Applications*, p. 185, 2010.

[26] M. P. O'brien, "Software comprehension–a review & research direction," *Department of Computer Science & Information Systems University of Limerick, Ireland, Technical Report*, 2003.

[27] M. Sensalire, "A framework to evaluate the effectiveness of software visualization tools in maintenance activities," Ph.D. dissertation, Makerere University, Uganda, 2009.

[28] H. A. Müller, K. Wong, and S. R. Tilley, "Understanding software systems using reverse engineering technology," in *Object-Oriented Technology for Database and Software Systems*. World Scientific, 1995, pp. 240–252.

[29] M.-A. Storey, "Theories, methods and tools in program comprehension: Past, present and future," in *13th International Workshop on Program Comprehension*. IEEE, 2005, pp. 181–191.

[30] B. A. Price, R. M. Baecker, and I. S. Small, "A principled taxonomy of software visualization," *Journal of Visual Languages & Computing*, vol. 4, no. 3, pp. 211–266, 1993.

[31] L. Merino and O. Nierstrasz, "The medium of visualization for software comprehension," Ph.D. dissertation, Universität Bern, 2018.

[32] C. Knight, *Visualisation for program comprehension: information and issues*. Visualisation Research Group,Centre for Software Maintenance, Department of Computer Science, University of Durham, 1998.

[33] M.-A. Storey, F. D. Fracchia, and H. A. Müller, "Cognitive design elements to support the construction of a mental model during software exploration," *Journal of Systems and Software*, vol. 44, no. 3, pp. 171–185, 1999.

[34] H. Bani-Salameh and A. Ahmad, "Software evolution visualization tools functional requirements: a comprehensive understanding," *11th International Conference on Software Engineering Advances*, p. 209, 2016.

[35] S. Bassil and R. K. Keller, "Software visualization tools: Survey and analysis," in *Proceedings 9th International Workshop on Program Comprehension.* IEEE, 2001, pp. 7–17.

[36] H. M. Kienle and H. A. Muller, "Requirements of software visualization tools: A literature survey," in *2007 4th IEEE International Workshop on Visualizing Software for Understanding and Analysis*. IEEE, 2007, pp. 2–9.

[37] J. Gibson, "The ecological approach to visual perception." 1979.

[38] L. R. Bartram, “Enhancing information visualization with motion,” Ph.D. dissertation, Simon Fraser University, 2001.

[39] P. Qiting, N. Uno, and Y. Kubota, “Kano model analysis of customer needs and satisfaction at the shanghai disneyland,” 2011.

APPENDIXES

.1 APPENDIX 1: User Interview Protocol

User Interview

Introduction: Hello, my name is Chris; I will be conducting the interview today. This interview will be part of my graduation project in order to have more insight into how you as users think about the tree diagram and your expectations about my product. In the professional interview, I need to give you a consent form to protect your privacy. But ofc we are friends, so I hope that we can skip that part. But I need to inform you that this interview will be recorded for research purposes only. Are you okay with that? In this interview, you just need to answer my questions and tell me what you are thinking. The expected time of this interview is around 20 mins.

Interview: *Part 1: Explain the fault tree and classify interviewees:*

- Have you ever heard or used a fault tree?
  - If yes, how often do you use it?
- On the scale from 1 to 10, How confident are you to work with fault trees?
  - If no: Okay then I will explain for you a bit about the fault tree. Show the picture, This is the fault tree, it is very similar to a tree diagram. It is used to visualize the risk of a model. So for example in the picture, the road trip is stranded because the phone fails and the car fails.

But dont worry about if you do not understand the fault tree now, you will have a little time later on to figure it out yourself. And that is the main goal of the interview.

Part 2: Get to know the first impression of users when they first see the prototype)

Instruction: Now I will give you a different fault tree and you will first have 15 seconds to look at it and then answer some questions

Question 1: What is your general feeling after seeing the fault tree? Question 2: Can you tell me what you remember now about the fault tree? Question 3: On the scale from 1 to 10, how motivated are you to keep looking at or working with the fault tree?

- Can you elaborate on your answer? Why cant it be higher?

Part 3: Get to know how users read the fault tree

Instruction: Thank you so much for your answer, now we are moving to the next part of the interview. In this part, you will have time to look at the fault tree again and I would like you to tell me what you are seeing. This method is called Thinking Aloud. Are you familiar with this method? If yes, okay good, then I do not have to explain every detail about the method. However, I still need to show you an example of how the method is done. Because we want to make sure all the participants will have the same instruction. Do you mind to see the example again?

- If no, Okay dont worry, I will explain with you right now, what is the Thinking Aloud and how people do it. Do you have any questions about the technique?

Task: Using the same method, can you walk me through the sequence of what you are seeing in the fault tree?

Question 4: How did you find the experience of reading the fault tree? Question 5: In your opinion, on the scale from 1 to 10, how easy to read the fault tree? Why the score cant be higher? Question 6: Where is the most difficult part for you when reading the fault tree? Question 7: What factor did influence your reading?

Part 4: Get to know how users interpret the fault tree Instruction: using the same method, Thinking Aloud, now give users tasks

Task: Question 8: Can you explain the relationship between basic event 1 and No operation? (2 events through 1 gate) Question 9: Now can you explain to me the path from 2 to top event? (1 branch from the top event to one basic event)

Question 10: How did you find the experience of reading the fault tree? Question 11: In your opinion, on the scale from 1 to 10, how easy to interpret the fault tree? Why cant the score be higher? Question 12: When interpreting the fault tree, did you encounter any difficulty? If yes, what are they? Question 13: What factor did distract you when interpreting the fault tree? Question 14: Did you use extra help to interpret the fault tree? Like the voice, finger, or mouse?

Part 5: Identify the user's expectation about the web application Instruction: Explain the idea of the concept of the project

Question 15: What experience do you expect to have with the interactive fault tree?

Question 16: What function should the application have to help you process the information?

.2 APPENDIX 2: User Interview Results

Question 1: What is your general feeling after seeing the fault tree?

Answer: All four participants did not show a very positive impression of the fault tree. Three out of four participants gave a relatively similar answer: they were confused and overwhelmed with the amount of information. However, they gave different explanations for this confusion. The first participant had a difficult time understanding the meaning of the gates while the second one had trouble with the small size of the text and boxes. The third participant was nervous because he is overwhelmed with the information within a short period of time. Only the fourth participants showed a neuter impression but make a remark that this fault tree can be improved a lot. For example, she pointed out the location of the table could be at the bottom of the tree so people can look at them easier.

Question 2: Can you tell me what you remember now about the fault tree?

Answer: The second question was asked so that we can find out how much information was actually stored in the participants'memory after 15 seconds. Four participants gave the same answers which are that they could remember the table showing some names of failure events and the first three events at the top of the tree. Some of them even could remember what is written in the first three events but some of them could not. This is a very interesting and important finding.

Question 3: On the scale from 1 to 10, how motivated are you to keep looking at or working with the fault tree?

Answers: The score collected were quite low: from 3 to 7. Besides all of the flaws mentioned in the first questions, they explained that the topic of the fault tree does not intrigue them and the layout is not appealing enough to attract them to look at. Two participants say further that if they were not demanded to look at the tree, they probably would not choose to look at it out of their free will.

Question 4: How did you find the experience of reading the fault tree?

Most of the participants do not feel comfortable to read this fault tree and very confused to read the fault tree after entering the loop. One person even gave up after seeing the loop.

Question 5: In your opinion, on the scale from 1 to 10, how easy to read the fault tree?

Answer: Most of the participants give a very high score: 8. But one participant shows his strictness by giving a 3. He said he had a big problem to find the way out of the loop to continue, so that is such a big flaw.

Question 6 7: Where is the most difficult part for you when reading the fault

tree? And What factor did influence your reading?

Answer: The most difficult part for all participants is the gate symbol. Participants showed an unpleasantness when they did not understand the INHIBIT and FDEP gate. Even they do not need to know the relationship between the two events related through the gate, the lack of understanding of the meaning of the two gates breaks their logic flow. The reading flow is also affected. Besides that, some people also complain about the size of the text and the disconnection between the fault tree and the table.

Question 8 9: Can you explain the relationship between basic event 1 and No operation? (2 events through 1 gate) and Can you explain to me the path from 2 to top event? (1 branch from the top event to one basic event)

Answer: Most participants could interpret correctly the relationship between event 1 and "No operation" but no one could answer the second question. The reason is that they do not understand the meaning of the INHIBIT gate with that given information. One person made a guess that the INHIBIT gate maybe is an AND gate because there are two arrows going towards the gate. This mistake shows how important the visual elements can be in order to imply the meaning of the object.

Question 10: How did you find the experience of reading the fault tree?

Answer: Most of the participants felt question 8 is not as challenging as question 9. They did not have any clue about how to answer this question and emphasized that they need an extra explanation to understand it. The visual elements did not give them any hint either.

Question 11: In your opinion, on the scale from 1 to 10, how easy to interpret the fault tree?

Answer: It is quite straightforward that they scored the interpretation experience of question 8 is much higher than question 9. The score of question 8 is around 9 and another is around 2.

Question 12: When interpreting the fault tree, did you encounter any difficulty? If yes, what are they?

Answer: When interpreting the fault tree, all four had a problem with the meaning of the gate and the descriptiveness of the text. Besides the gate's problem, some people had trouble understanding fully the event. They said the texts were not clear enough for them to guess the relationship between these events.

Question 13: What factor did distract you when interpreting the fault tree?

Answer: All four people said no.

Question 14: Did you use extra help to interpret the fault tree? Like the voice,

finger, or mouse?

Answer: Two out of four people felt the tree is not so complicated that they need to use extra help such as voice or finger to interpret the fault tree. However, the other two use their fingers to track where they are at the fault tree while encountering the loop.

Question 15: What experience do you expect to have with the interactive fault tree?

Answer: Most people had the same idea that they want to have an easy-to-use application. They all emphasized the clarity of the product: they should be able to easily navigate through the product and everything should be intuitive. They can instantly know what they have to do when they see a product.

Question 16: What tool do you need to read the fault tree better? And to interpret the fault tree better?

Answer: There were five functions that were suggested by participants: Hover to an event or link to see more information. Showing animation to explain the concept. Highlight the related events Divide the information of fault trees into different layers. Filter out the irrelevant event or information.

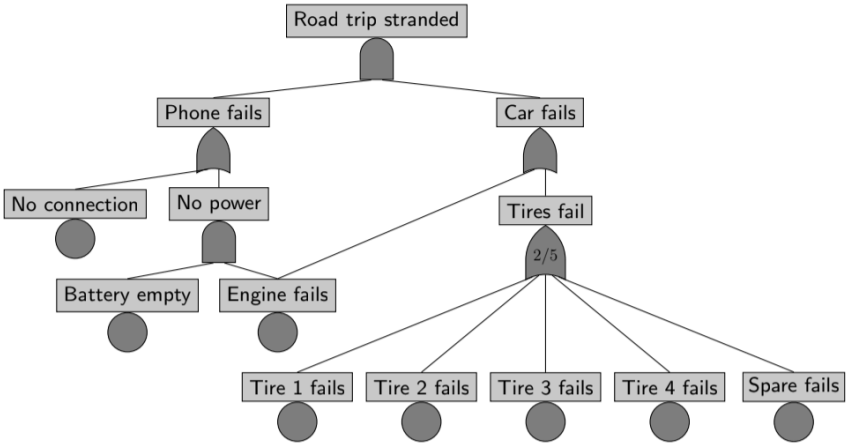


Figure 1: Old layout

.3 APPENDIX 3: Evaluation Question

You will have time to look at a different fault tree and I would like you to tell me what you are seeing. This method is called Thinking Aloud. Are you familiar with this method?

- If yes, okay good, then I do not have to explain every detail about the method. However, I still need to show you an example of how the method is done. Because we want to make sure all the participants will have the same instruction. Do you mind to see the example again?
- If no, Okay dont worry, I will explain with you right now, what is the Thinking Aloud and how people do it. Do you have any questions about the technique?

Part 1: Reading flow:

Task 1: Can you walk me through the sequence of what you are seeing in this old layout?

Question 1: On the scale from 1 to 10, how easy do you find about the reading flow that you just described to me?

Task 2: now I will give you a different fault tree. This fault tree I already add some different elements. So can you walk me through the sequence of what you are seeing in the new fault tree for me to see the effect?

Question: On the scale from 1 to 10, how easy do you find about the reading flow that you just described to me?

Question 2: Why there is a difference between the scores? Can you explain?

Part 2: Visual effect:

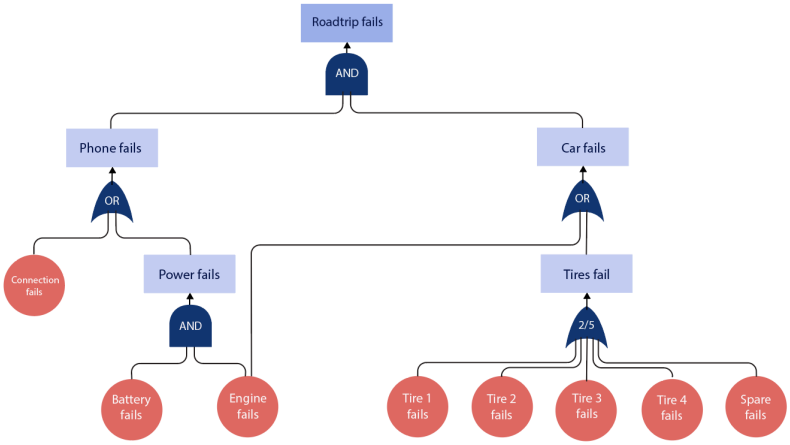


Figure 2: new layout

In the normal node-link diagram, there are two main elements: node and link. Question 3: Can you tell me how many different kinds of nodes do you think there are in the fault tree? Question 4: How can you distinguish them? Question 5: Next part, when you see the node Road trip fails and Phone fails, which one do you think is the cause and which one is the result? Question 6: Is there any visual element that tells you about this cause and effect relationship? Question 7: How clear do you think the text is compared to the background?

Part 3: Web interface:

Now I will show you the interactive version of this fault tree. And we will step by step discover it together okay? Show them the website without the fault tree

Question 8: What are you seeing on the screen? Question 9: What item do you think is excessive? Question 10: What item do you think is missing here? Question 11: What do you expect to see if you click the button? Question 12: Where the fault tree do you expect to see on the screen?

Part 4: Function evaluation:

Give them 1,2 minutes to discover the fault tree themselves.

Question 13: How many functions do you think the interactive fault tree has and what they are?

- If they discover all, pass
- If they do not, show them.

Comment function:

- Question 14: How visible/easy to find the Comment function is for you, on the scale from

1 to 10?

- Question 15: How can you improve that score?
- Question 16: How fast can you finish this function? (Do you think the comment should show faster?)
- Question 17: To what extent, do you describe this function is familiar to you?
- Question 18: What is the effect of this function on you?
- Question 19: How do you feel when there is a function like that?
- Question 20: How do you feel if there is no function like that?
- Question 21: If you were able to change it, what would you change?

Animation:

- Question 23: Do you think the animation is too slow or too fast for you?
- Question 24: To what extent, do you describe this function is familiar to you?
- Question 25: What is the effect of this function on you?
- Question 26: How do you feel when there is a function like that?
- Question 27: How do you feel if there is no function like that?
- Question 28: If you were able to design the animation in your own style, how would it be?

Emphasis function:

- Question 29: How visible/easy to find this function is for you, on the scale from 1 to 10? (How can you improve that score?)
- Question 30: To what extent, do you describe this function is familiar to you?
- Question 31: What is the effect of this function on you?
- Question 32: How do you feel when there is a function like that?
- Question 33: How do you feel if there is no function like that?
- Question 34: How do you interpret this branch?
- Question 35: If you were able to change it, what would you change?

Button function:

- Question 36: How visible/easy to find the button is for you, on the scale from 1 to 10? (How can you improve that score?)
- Question 37: To what extent, do you describe this function is familiar to you?
- Question 38: What is the effect of this function on you?
- Question 39: How do you feel when there is a function like that?
- Question 40: How do you feel if there is no function like that?
- Question 41: If you were able to change it, what would you change?