

Institute for Visualization and Interactive Systems

University of Stuttgart  
Universitätsstraße 38  
D-70569 Stuttgart

Master Thesis No. 10250005

# Visual Correlation Analytics of Event-based Error Reports for Advanced Manufacturing

Iqbal Nazir

**Course of Study:** M.Sc. in Information Technology

**Examiner:** Prof. Dr. Thomas Ertl

**Supervisor:** Dipl.-Inf. Dominik Herr

**Commenced:** January 18, 2017

**Completed:** July 06, 2017

**CR-Classification:** H.5.2, H.2.8



## Abstract

With the growing digitalization and automation in the manufacturing domain, an increasing amount of process data and error reports become available. To minimize the number of errors and maximize the efficiency of the production line, it is important to analyze the generated error reports and find solutions that can reduce future errors. However, not all errors have the equal importance, as some errors may be the result of previously occurred errors. Therefore, it is important for domain experts to be able to find out the correlations among the errors. A visual analytics approach may help visualize and understand how the errors relate to each other. The goal of this thesis is to develop a concept that helps the analysts understand the cause-effect relations of reported errors. The concept of this thesis is based on Markov model, which helps to find that relations from a production line data. At first, to understand the source of the errors and position of that source in the production line, data overview visualizations like treemap, block diagram are initiated. Then, for the purpose of detailed analysis for any source of the errors and to show the correlations among them, visualizations like heatmaps, rooted trees, and bar charts are used. The adapted visual analytics approaches show that it is possible to visualize the cause-effect correlations among the errors with an appropriate model and thus may help reduce the number of errors in the production line.



# Contents

1	Introduction	15
2	Theoretical Fundamentals	17
2.1	Stochastic Process . . . . .	17
2.1.1	Stochastic Process and Deterministic Process . . . . .	17
2.2	Markov Process and Markov Chain . . . . .	18
2.2.1	Markov Process Definition . . . . .	19
2.2.2	Markov Chain Definition . . . . .	19
2.3	Transition Probabilities . . . . .	20
2.3.1	First-Order Transition Probability . . . . .	21
2.3.2	$m$ -Order Transition Probability . . . . .	21
2.3.3	Transition Matrix and Transition Diagram . . . . .	22
2.4	Information Visualization . . . . .	22
2.4.1	Visualization Definition . . . . .	22
2.4.2	Information Visualization Definition . . . . .	24
2.4.3	Visualization Variables . . . . .	24
2.4.4	Interactive Visualization . . . . .	25
2.5	Hierarchical and Comparative Data Visualizations . . . . .	25
2.5.1	Tree Visualization . . . . .	26
2.5.2	Treemap . . . . .	27
2.5.3	Treemap Generation Algorithm . . . . .	28
2.5.4	Matrix Diagram . . . . .	30
2.5.5	Heatmap . . . . .	31
2.5.6	Color Theory and Mapping . . . . .	33
2.6	Visual Analytics . . . . .	34
3	Related Works	35
4	Concept	41
4.1	Data Analysis . . . . .	41
4.1.1	Structure and Hierarchy of Data . . . . .	41
4.1.2	Preliminary Filtering of Data . . . . .	42

4.2	Design and Architecture Concept . . . . .	43
4.2.1	Calculation and Preparation of Data . . . . .	43
4.3	Visualization . . . . .	44
4.3.1	Data Overview Visualization . . . . .	44
4.3.2	Visualization for Pairwise Analysis . . . . .	47
4.3.3	Visualization for In-depth Analysis . . . . .	49
5	Implementation . . . . .	53
5.1	Architecture and Models . . . . .	53
5.1.1	Adaption of Frameworks . . . . .	53
5.1.2	Database Fields and View . . . . .	54
5.1.3	Core Models . . . . .	55
5.2	Implementation of Visualizations . . . . .	56
5.3	Implementation of Data Overview Visualizations . . . . .	57
5.3.1	Electronic Circuit Block Diagram . . . . .	57
5.3.2	Treemap . . . . .	57
5.4	Pairwise Analysis Visualizations . . . . .	61
5.4.1	Generation of Transition Matrix . . . . .	61
5.4.2	Analysis with Heatmap for First Order Transition Matrix . . . . .	61
5.4.3	Analysis with Heatmap for Second Order Transition Matrix . . . . .	62
5.5	In-depth Analysis Visualizations . . . . .	63
5.5.1	Generation of Fault-Code Tree . . . . .	63
5.5.2	Analysis with Fault-Code Tree . . . . .	63
5.5.3	Analysis with Multi-Circuit Fault-Code Tree . . . . .	64
5.5.4	Generation of Bar Charts from Fault-code Tree . . . . .	65
5.5.5	Analysis with Fault-Code Duration Bar Chart . . . . .	65
5.5.6	Analysis with Inter Fault-Code Delay Bar Chart . . . . .	66
6	Evaluation and Discussion . . . . .	69
6.1	General Information . . . . .	69
6.1.1	Reading the User Manual . . . . .	69
6.1.2	Performing a List of Tasks . . . . .	70
6.1.3	Answering the Questionnaire . . . . .	70
6.2	Evaluation for Data Overview Visualizations . . . . .	70
6.2.1	ELC Block Diagram . . . . .	70
6.2.2	Treemap . . . . .	71
6.3	Evaluation for Correlation Analysis Visualizations . . . . .	71
6.3.1	Heatmaps . . . . .	72
6.3.2	Fault-Code Tree . . . . .	73
6.3.3	Multi-Circuit Fault-Code Tree . . . . .	74
6.3.4	Fault-Code Duration Bar Chart . . . . .	76

6.3.5	Inter Fault-Code Delay Bar Chart . . . . .	76
6.3.6	Use Case Scenario . . . . .	78
6.4	Result and Discussion . . . . .	81
6.4.1	Discussion for Data Overview Visualizations . . . . .	81
6.4.2	Discussion for Pairwise Analysis Visualizations . . . . .	81
6.4.3	Discussion for In-depth Analysis Visualizations . . . . .	82
7	Summary and Future Work . . . . .	83
7.1	Summary . . . . .	83
7.2	Future Work . . . . .	84
	Appendices . . . . .	85
A	Class Diagrams . . . . .	87
A.1	Back-end Models . . . . .	87
B	Evaluation Appendices . . . . .	89
B.1	How-to Used for the Evaluation . . . . .	89
B.2	Task List . . . . .	99
B.3	Evaluation Questionnaire . . . . .	102
	Bibliography . . . . .	115





# List of Figures

2.1	Jumping frog example of Markov Chain by Howard . . . . .	20
2.2	Example of transition diagram . . . . .	23
2.3	Information visualization pipeline . . . . .	24
2.4	Different visual variables . . . . .	25
2.5	Example of a simple depth-3 tree . . . . .	26
2.6	Example of treemap generated from a tree diagram . . . . .	27
2.7	Slice and dice algorithm for treemap . . . . .	28
2.8	Subdivision problem of the slice-and-dice algorithm, which may create many thin and elongated rectangles. Figure is taken from Bruls [BHV00].	29
2.9	Squarified Treemap procedure . . . . .	30
2.10	Example of a matrix diagram . . . . .	31
2.12	Heatmap for 1st and 2nd order transition matrix . . . . .	32
2.11	Heatmap representation of transition matrix . . . . .	32
2.13	5-class color scheme by Brewer . . . . .	33
2.14	Color theory . . . . .	33
2.15	Visual analytics model by Keim . . . . .	34
3.1	Event-based data discovery by Cook and Wolf . . . . .	35
3.2	Screenshot of <i>Outflow</i> visualization tool by Wongsuphasawat and Gotz .	36
3.3	Screenshot of <i>Frequence</i> visualization tool by Perer and Wang . . . . .	37
3.4	A use-case example of Plaisant's <i>Lifelines</i> . . . . .	38
3.5	Screenshot of <i>Viztree</i> visualization by Lin et al. . . . .	38
3.6	A use-case example of <i>Lifelines</i> by Burch et al. . . . .	39
4.1	Production line hierarchy . . . . .	42
4.2	Core idea of the thesis . . . . .	44
4.3	Concept of electronic circuit block diagram . . . . .	45
4.4	Concept of electronic circuit block diagram with bar chart . . . . .	46
4.5	Concept of treemap . . . . .	47
4.6	Concept of heatmaps . . . . .	48
4.7	Concept of tree visualization . . . . .	49
4.8	Concept with bar chart . . . . .	50
4.9	Concept with stacked bar chart . . . . .	51

5.1	Web application architecture . . . . .	54
5.2	Screenshot of the database <i>view</i> . . . . .	55
5.3	Home page of the prototype . . . . .	56
5.4	All visualizations in one figure . . . . .	58
5.5	The screenshots of implemented ELC block diagrams . . . . .	59
5.6	Screenshot of the production line treemap at depth-0 . . . . .	60
5.7	Context-menu and navigation bar of treemap . . . . .	60
5.8	Station and Fault-code view of treemap . . . . .	61
5.9	Hetmaps for first and second order transition matrices . . . . .	62
5.10	Implemented fault-code (FC) tree . . . . .	64
5.11	Multi-circuit fault-code tree . . . . .	65
5.12	Fault-code duration bar chart . . . . .	66
5.13	Inter fault-code delay bar chart . . . . .	67
6.1	Evaluation of ELC Block Diagram with Circles . . . . .	71
6.2	Evaluation of ELC Bar Chart . . . . .	71
6.3	Analysis of a fault-code . . . . .	72
6.4	Analysis of a fault-code pattern from heatmap-2 . . . . .	72
6.5	Analysis of a fault-code in STN-2 and ELC-4543 from FC Tree . . . . .	73
6.6	Analysis of a fault-code pattern in multi-circuit FC Tree . . . . .	75
6.7	Analysis of a fault-code pattern with FC duration bar chart . . . . .	76
6.8	Analysis of a fault-code pattern with inter FC delay bar chart . . . . .	77
6.9	Use case analysis with overview visualizations . . . . .	78
6.10	Use case analysis with heatmpas . . . . .	79
6.11	Use case analysis with FC tree and bar charts . . . . .	79
6.12	Use case analysis with multi-circuit FC tree and bar charts . . . . .	80
A.1	Class diagram of production line elements. . . . .	87
A.2	Class diagram of three core models . . . . .	88

# Listings

- 2.1 Slice and Dice Algorithm . . . . . 28
- 2.2 Squarified Treemap Algorithm . . . . . 29



# Abbreviations

<b>CSS</b>	Cascading Style Sheets
<b>DB</b>	Database
<b>D3</b>	Data-Driven Documents
<b>ELC</b>	Electronic Circuit
<b>FC</b>	Fault-Code
<b>HTML</b>	HyperText Markup Language
<b>JS</b>	JavaScript
<b>MVC</b>	Model View Controller
<b>OS</b>	Operating System
<b>SQL</b>	Structured Query Language
<b>STN</b>	Station
<b>SVG</b>	Scalable Vector Graphics



# 1 Introduction

To cope with the ever improving competitors, there have been growing digitalization and automation in the domain of manufacturing. To smoothen the operation and to improve the analysis process of the production line, all the errors that occur are stored in the database. This growing number of error reports are important to study the actual causes of the errors and find effective solutions, which could minimize similar future problems. However, all the error reports do not have the equal importance, as some error reports may be the result of previously reported errors or there could be an error, which could have resulted because of broken machine or malfunction of some part of a machine. Hence, it is important for domain experts to understand, how the reported errors are correlated to each other. Finding a suitable model and implementing a visual analytics approach to that model may help to understand these relations and thus may reduce the number of errors in a significant amount.

The objective of this thesis is to develop a concept to visually analyze the correlation of errors across a production line. In the first step, a Markov model is built according to the likelihood of the correlation of the reported errors. The resulting likelihood of correlations is then visualized with the help of a web application to provide an analyst with a few means to find and interactively inspect the correlations of errors.

The web application is implemented with three major types of visualizations. The data overview visualizations help visualize the overview and location of the errors. These visualizations also help filter and sort the errors based on their sources in the production line. The pairwise analysis visualizations help find one-to-one correlations, while in-depth analysis visualizations show the cause-effect correlations among the errors.

The evaluation of the developed concept was done along with some tasks related to error analysis. A use case along with discussion and result is provided after the evaluation, which demonstrates the functionality of the implemented web application. After the thesis work, some future works are also identified which are discussed at the last chapter of this paper.





## 2 Theoretical Fundamentals

This chapter discusses some of the key topics that are necessary to build a conceptual foundation for the evolution of this thesis. This thesis work is largely developed with the concept of Markov model. This chapter summarizes the theories related to the Markov model and fundamental concepts of information visualization.

### 2.1 Stochastic Process

Markov chains and Markov processes are stochastic processes. To better understand Markov process, the stochastic process should be defined first.

According to J.L. Doob, any process in nature whose evolution can be analyzed successfully in terms of probabilistic rules can be termed as stochastic process [Doo42].

According to Cinlar,

A stochastic process with state space  $E$  is a collection  $\{X_t; t \in T\}$  of random variables  $X$ , defined on the same probability space and taking values in  $E$ . The set  $T$  is called its parameter set. If  $T$  is countable, especially if  $T = \mathbf{N} = \{0, 1, \dots\}$ , the process is said to be *discrete parameter* process. Otherwise, if  $T$  is not countable, the process is said to have a *continuous parameter* [Cin13, p. 7].

#### 2.1.1 Stochastic Process and Deterministic Process

Stochastic models can be defined as the opposite of deterministic models. The difference between a deterministic process and a stochastic process is that in a deterministic process, for a given starting point of a process the complete trajectory or path of travel of that process is determinable by a set of equations whereas trajectory of a stochastic process can only be predicted in terms of probability. equations that describe exactly how the system evolves over time. The evolution of the processes of stochastic models are at least

partially random and for several runs of a process, the paths are also random, which are often termed as realizations of the process.

### 2.2 Markov Process and Markov Chain

This section shortly describes the Markov process and Markov chain theory. According to Howard, "The Markov process is a probabilistic model useful in analyzing complex systems" [How12, p. 1–5]. He also expressed the importance of having the concepts of state and state transition to understand Markov process. In his book "Dynamic Probabilistic Systems", he explained the concept of the state and state transition as follows:

#### Definition of State

Howard in his book "Dynamic Probabilistic Systems" said that usually the present situation in a physical system can be described by specifying the values of a number of variables that describe the system [How12, p. 1–5]. That is, the variables of a system, which describe that system are called state variables. According to Howard, temperature, pressure, and volume are the state variables for a chemical process; in the same way, variables like position, mass, and velocity are state variables for the spacecraft. These crucial variables, which define a system are termed as "state" variable. To clarify specify a system's state, the value of each state variable for that system has to be determined first. In Figure 2.1, each pad occupied by the from is a state.

#### Definition of Transition

Transition is the process or a period of changing from one state or condition to another.

According to Howard, transition happens when a system passes from one state to another state as the time goes on and hence shows dynamic behavior [How12, p. 1–5]. For example, the change in the chemical system can be caused by applying heat or by increasing volume etc. On the other hand, a spacecraft might experience a change of state if it burns fuel or even when it coasts under the earth's gravitational force. According to R.A. Howard, such changes of state are termed as *state transitions* or more specifically, *transitions*. The most specific *state transition* model would allow states described by continuous variables and transitions that could occur at any time. But for the sake of simplicity, one could consider systems that may occupy only a finite number of states.

### 2.2.1 Markov Process Definition

A process that can stay in more than one state, can switch positions or make transitions among those states is called Markov process. In other words, the future state depends on the most recent past state(s).

There can be multiple order of Markov processes, which is further discussed in Section 2.3.

#### First Order Markov Process

A first order Markov process depends only on the most recent previous state.

#### m-Order Markov Process

An m-order Markov process depends on the most recent previous m states. For  $m = 2$ , the processes depend on two preceding ones. This is called second order Markov process in that case.

### 2.2.2 Markov Chain Definition

A Markov chain is a statistical model of a system that moves sequentially from one state to another. A Markov process is said to be a Markov chain if the state space is discrete and finite.

According to Levinson [LRS83],

A probabilistic function of a (hidden) Markov chain is a stochastic process generated by two interrelated mechanisms, an underlying Markov chain having a finite number of states, and a set of random functions, one of which is associated with each state. At discrete instants of time, the process is assumed to be in some state and an observation is generated by the random function corresponding to the current state. The underlying Markov chain then changes states according to its transition probability matrix.

If there are  $r$  number of states  $s$ , then the set of states  $S$  can be expressed as  $S = \{s_1, s_2, \dots, s_r\}$ . The process begins transitioning in one of these states and moves to another state. One single move is called one single step. If the chain is currently in state  $s_i$ , then it can sequentially change its position to state  $s_j$  at the next step with a probability denoted by  $p_{ij}$ . The probabilities  $p_{ij}$  are termed as *transition probabilities*,



**Figure 2.1:** Markov chain explained by Howard with a picture of a frog jumping through lily pond, taken from Howard [How12].

which is described in Section 2.3. This probability from a present state to a future state depends only on the current state (2.2.1). Howard provides the description of a Markov chain with a picture 2.1 as a frog jumping on a set of lily pads [How12, p. 1–5]. The frog starts on one of the pads and then jumps from lily pad to lily pad with the appropriate transition probabilities. A Markov chain can be described by a transition probability matrix

### 2.3 Transition Probabilities

Transition probability is the probability of transition from one state to another. The one-step transition probability is the probability of transitioning from one state to another in a single step. If the transition probabilities are not dependent on time index  $n$ , the Markov chain is said to be time homogeneous. There can be multiple orders of transition probabilities as explained below:

### 2.3.1 First-Order Transition Probability

If the transition probability is  $P$ , then the first-order transition probability matrix,  $p_{ij}$ , can be expressed as

$$p_{ij} = Pr\{X_n = j | X_{n-1} = i\}$$

In Figure 2.2, the weight of each edge is the associated probability, which is shown next to it. For example, the first order transition probability from state 2 to state 1,  $p_{21} = \{p_n = 2 | p_{n-1} = 1\} = 0.5$  and in the same way, the loop from state 1 to state 1 represents the probability  $p_{11} = p_n = 1 | p_n = 1$ , which is equal to 0.4 and so on.

### 2.3.2 $m$ -Order Transition Probability

The  $m$ -order transition probability  $p_{ij}^{(m)}$  is the probability of transition from state  $i$  to state  $j$  in  $m$  steps.

$$p_{ij}^{(m)} = Pr\{X_{n+m} = j | X_n = i\}$$

In this case, the  $m$ -order transition matrix is simply denoted as  $P^{(m)}$  whose elements are the  $m$ -order transition probabilities  $p_{ij}^{(m)}$ .

$m$ -order transition probability can also be determined from transition matrix as well as from transition diagram. In Figure 2.2, if  $m = 2$ , then 2<sup>nd</sup>-order transition probability from state 2 to state 1 can be calculated from the following formula:

$$p_{12}^{(2)} = p_1 + p_2$$

where:  $p_1 = 1 \times 0.5 = 0.5$  is the probability of edge sequence 3,2,1 and  $p_2 = 0.4 \times 0.4 = 0.16$  is the probability of edge sequence 1,1,1.

If  $i$  is the current state of a process and  $j$  is the state, which is going to be occupied by that process in the next transition, then  $p_{ij}$  is the transition probability and this always satisfies the following two conditions [How12, p. 4],

$$0 \leq p_{ij} \leq 1$$

$$0 \leq i, j \leq N$$

where  $N$  is a finite number.

### 2.3.3 Transition Matrix and Transition Diagram

According to Kachapova, a homogeneous finite Markov chain can be determined by its initial state distribution and its transition matrix and can be graphically represented with the help of a transition diagram. Both the transition matrix and transition diagram are equivalent to each other. The transition diagram of a Markov chain is a single weighted directed graph. If the transition probability  $p_{ij} > 0$ , then each vertex of this graph represents a state 2.2 of the Markov chain and there is a directed edge from vertex  $j$  to vertex  $i$ . This edge has the weight or probability of  $p_{ij}$  [Kac13]. There is no edge for  $p_{ij} = 0$ . Figure 2.2 is an example of transition diagram, whose corresponding transition matrix is explained with the following example.

For example, if a Markov chain has states 6 states namely: 1, 2, 3, 4, 5, 6 and their first order transition probabilities are as shown in Figure 2.2, then the corresponding transition matrix can be presented as the following [Kac13]:

$$S = \begin{bmatrix} 0.4 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0.3 & 0.5 & 0 & 0 & 0 & 0 \\ 0.3 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (2.1)$$

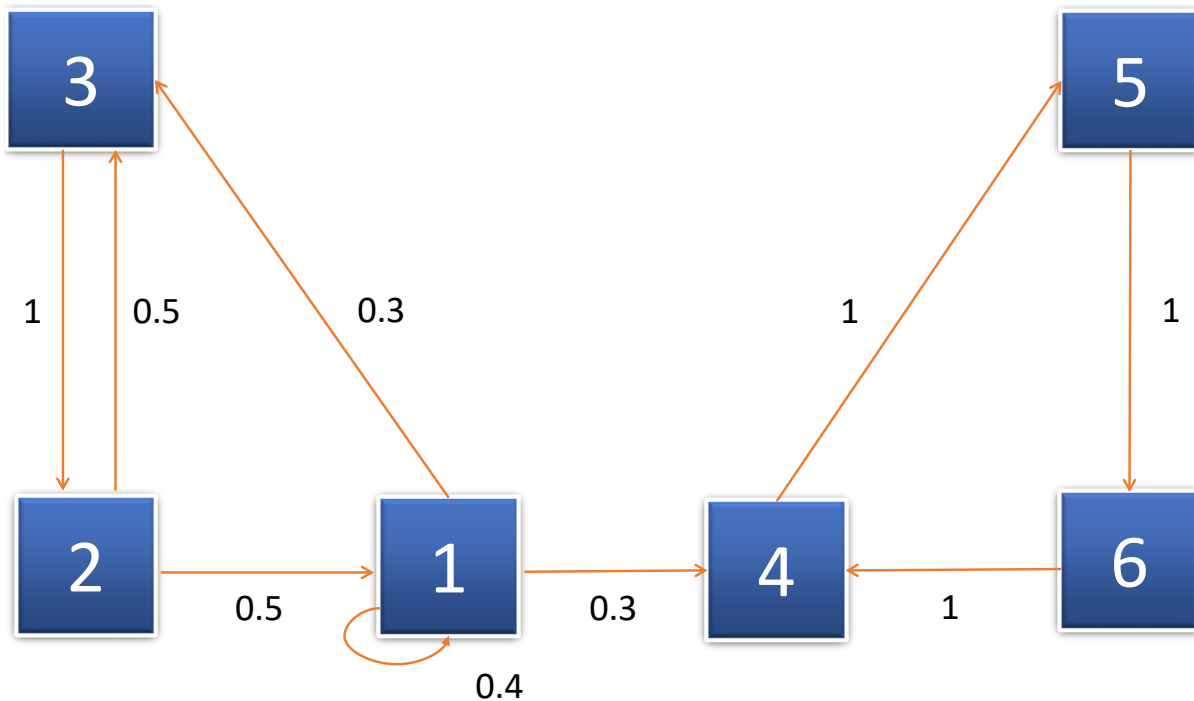
In Figure 2.2, the weight of each edge is the corresponding probability, which is shown next to it. For example, the transition probability from state 2 to state 1,  $p_{21} = \{ p_n = 2 \mid p_{n-1} = 1 \} = 0.5$  and in the same way, the loop from state 1 to state 1 represents the probability  $p_{11} = p_n = 1 \mid p_n = 1$ , which is equal to 0.4 and so on.

## 2.4 Information Visualization

This section summarizes the definitions of some basic terms related to information visualization.

### 2.4.1 Visualization Definition

Visualization is the graphical representation technique of data or abstraction for creating some kind of mental vision or picture of that data or abstraction, which is not visible or present to sight. Visualizations are the process of making abstract data visible to the



**Figure 2.2:** The transition diagram of the Markov chain explained in Subsection 2.3.3, where the squares with the numbers 1, 2, 3, 4, 5 and 6 are the states, and values by the arrow lines are the respective transition probabilities. This figure is adapted from Kachapova [Kac13].

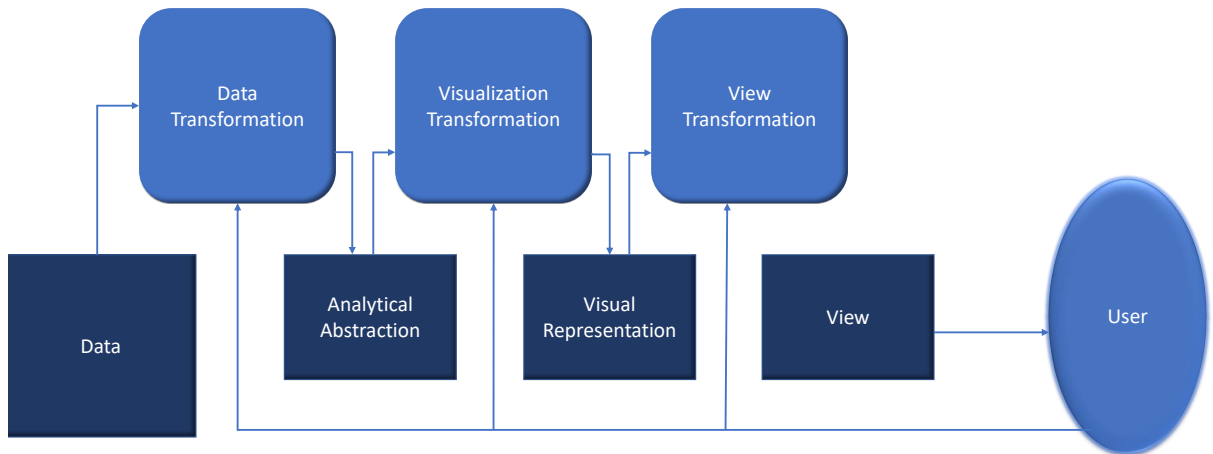
mind or imagination internally and taking a decision on the basis of that imagination externally. Visualization helps people understand both spatial and non-spatial data. According to Munzner, augmenting human capabilities in different situations where, for a computer, it is not adequately well defined to handle the problem with some algorithm is the goal of visualization [Mun14].

The major sub-fields of visualization are:

- Information Visualization
- Scientific Visualization
- Visual Analytics

According to Ferreira and Levkowitz,

"Both scientific visualization and information visualization create graphical models and visual representations from data that support direct user interaction for exploring and acquiring insight into useful information embedded in the underlying data [DL03]."



**Figure 2.3:** Information visualization pipeline proposed by Card [CMS99].

### 2.4.2 Information Visualization Definition

Information visualization is the technique that helps visually represent information. In the world of ever increasing information volumes, information visualization quickens understanding and inner meaning of data by amplifying cognitive performance [CARD08]. According to Manovich, it is sometimes difficult to come up with an acceptable common definition of information visualization. He described information visualization provisionally as the mapping between discrete data and a visual representation [Man10].

According to Stuart Card,

"Information visualization is a set of technologies that use visual computing to amplify human cognition with abstract information [CARD08]."

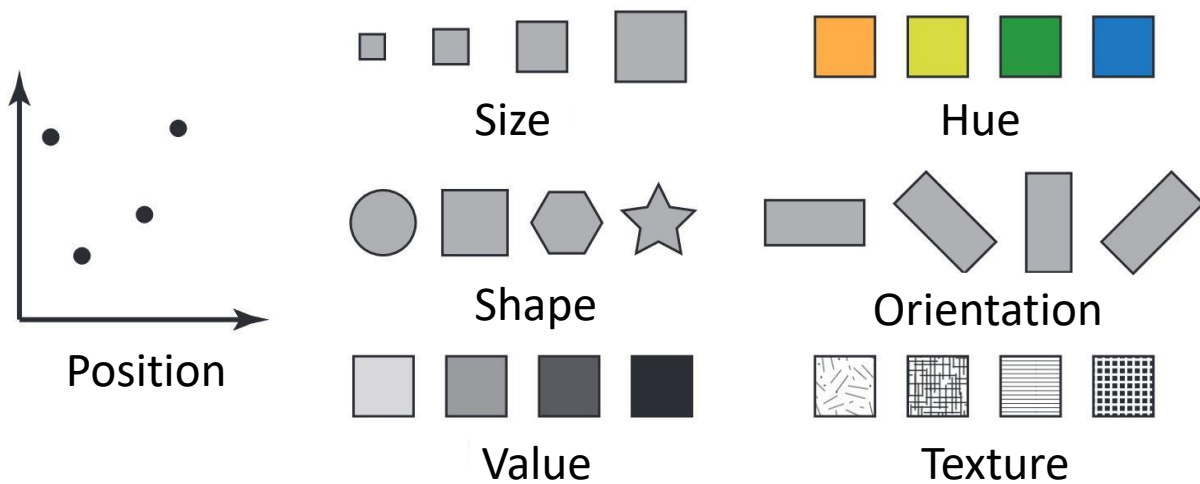
Card described different stages of information visualization with a block diagram 2.3. According to Card, after the initial transformation, data becomes ready for analytical abstraction, which is again transformed for visual representation [CMS99].

### 2.4.3 Visualization Variables

Every picture tells a story. —Rod Stewart, 1971

There exist a few parameters and variables, which need to be considered to design information visualization. As shown in Figure 2.4, proposed by Bertin [Ber83], these variables are namely: position, size, shape, value, hue, orientation, and texture.





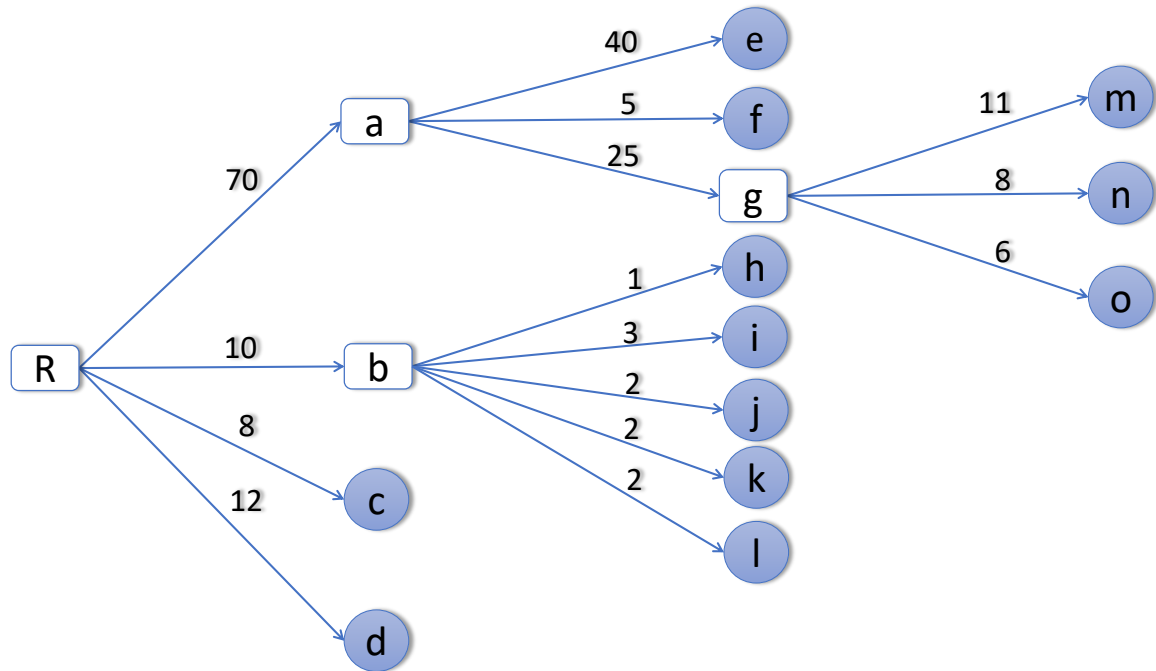
**Figure 2.4:** Different visual variables like position, size, shape, value, hue, orientation, texture etc. that are used in information visualization and visual analytics, taken from Zhao [Zha15].

#### 2.4.4 Interactive Visualization

Interactive visualization is one of the major fields in the visualization world that enables the analysis of data via the manipulation of different kinds of chart images, with the visual variables described in Subsection 2.4.3 and motion of visual objects representing various aspects of the dataset being analyzed [gartner20]. Interactivity plays a very important role to create visualization tools that handle complex and large datasets. Interactive visualization minimizes the limitations of both user and display in the scenario where it becomes very difficult to show everything at once [Mun14]. This technique provides a good number of visualization options, which extend the ability of conventional bar, line, pie charts, including tree and heatmaps, geographic maps, scatter plots and other sophisticated visuals. Users are able to explore and analyze the data by playing around and interacting with a visual representation of it by these tools. Figure 2.3 shows the role of user interaction in information visualization as described by Card.

## 2.5 Hierarchical and Comparative Data Visualizations

This section briefly outlines the descriptions of some visualization techniques used to visualize the hierarchical data structure. This section also discusses some visualization techniques that help visualize the correlations and dependencies of data.

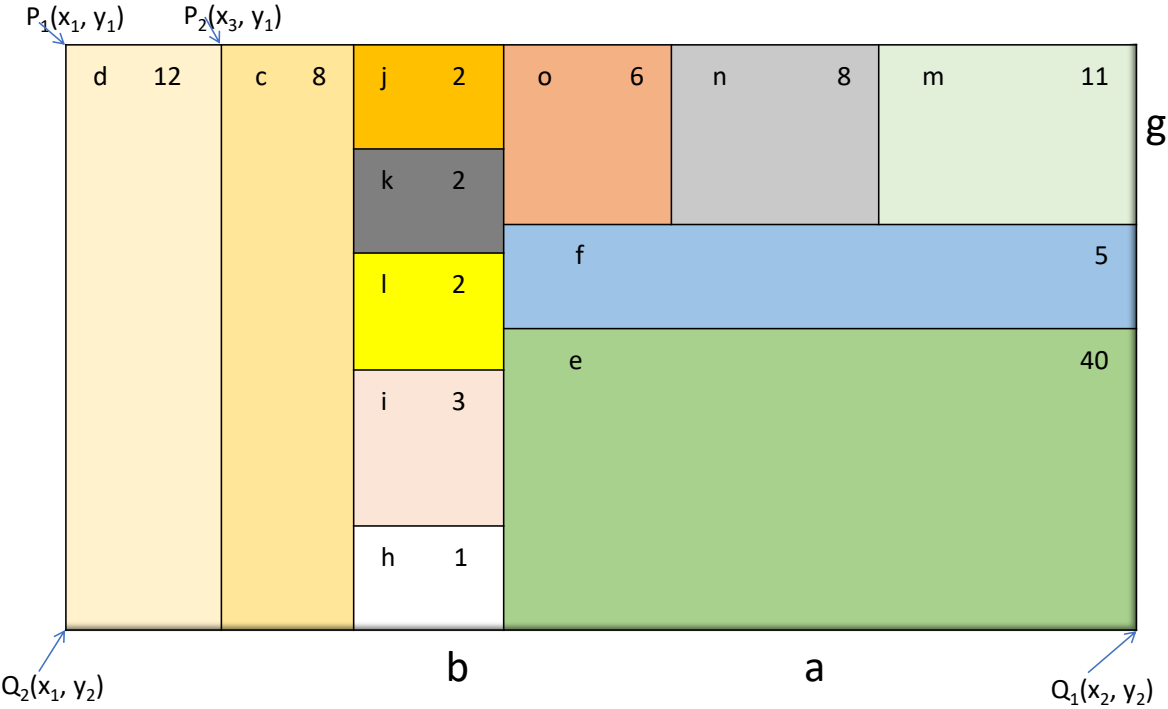


**Figure 2.5:** Traditional depth-3 tree structure where lower case letters indicating the leaf nodes and numbers indicating the weight of each leaf node, circles are end-level leaf nodes, while rectangles are the roots of corresponding sub-trees. The image is adapted from Shneiderman [Shn92].

### 2.5.1 Tree Visualization

Tree visualization is one of the most common and well-known techniques to represent hierarchical data. This visualization technique can also be used for non-hierarchical data elements to represent additional relations between different data items [Hol06]. Trees can be visually represented in different ways like node-link, stacking, nesting etc. Again, node-link representation of the rooted trees can have multiple layouts like top-down, left-to-right etc. The left-to-right layout is shown in Figure 2.5.

The advantage of rooted tree visualization is that if an appropriate layout is chosen, the hierarchical structure becomes obvious and the links can also be easily followed. But the problem with this visualization is that it is not very space-efficient for the large datasets and there are scalability problems too.



**Figure 2.6:** Treemap generated from the tree diagram of Figure 2.5, where each rectangle represents a leaf node. Some rectangles have sub-rectangles representing their corresponding leaf nodes. Nodes of the tree are shown by letters, while associated weights are represented by numbers. This image is modified from Shneiderman [Shn92].

2.5.2 Treemap

Treemap is a visualization technique that is used to represent hierarchical data and the contribution of the corresponding data element as shown in Figure 2.6. The treemaps can either be static or dynamic. In the static treemaps, data are usually rendered as static pictures in acceptable formats. In dynamic or interactive treemaps, data are displayed in applications that provide the facility of user interaction. For example, a user might be able to zoom into the deeper elements of the overall hierarchy [SDC04]. Sometimes, color and annotation can also be used to provide with extra information about the nodes and the children nodes.

### 2.5.3 Treemap Generation Algorithm

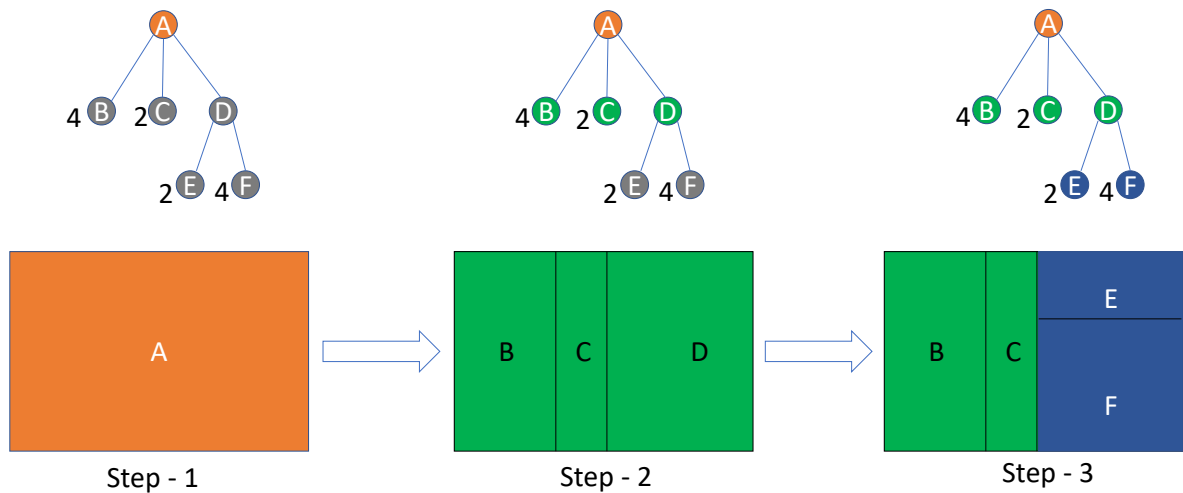
#### Slice and Dice Algorithm

Shneiderman used *slice-and-dice* algorithm to generate treemap. This procedure is defined in Algorithm 2.1 and explained in Figure 2.7. The disadvantage of this algorithm is that it may create many thin and elongated rectangles. It also suffers from poor aspect ratio problem. Thus the labeling of child nodes and interaction with them becomes difficult as shown in Figure 2.8.

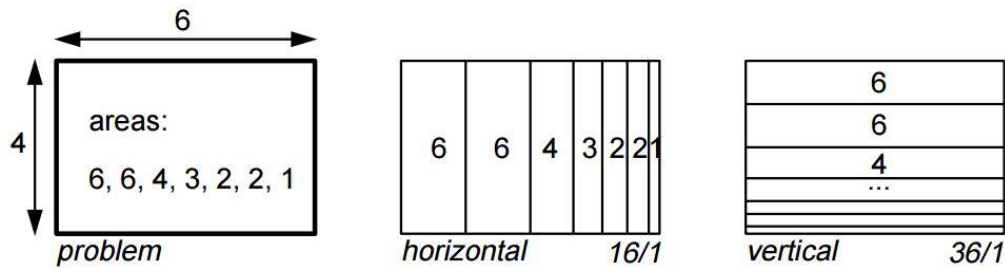
```

Treemap(root, P[0..1], Q[0..1], axis, color)
  Paint_rectangle(P, Q, color) -- paint full area
  width := Q[axis] - P[axis] -- compute location of next slice
  for i := 1 to num_children do
    Q[axis] := P[axis] + (Size(child[i])/Size(root))*width
    Treemap(child[i], P, Q, 1 - axis, color) -- recur on each
      slice, flipping axes
  P[axis] := Q[axis];
endfor
    
```

**Listing 2.1:** Slice and Dice Algorithm



**Figure 2.7:** Slice and dice algorithm to generate treemap is explained with this figure. In step-1, an area is defined for the whole tree. In step-2, the root node is subdivided horizontally or vertically according to the weight of child nodes. In step-3, Direction of subdivision is flipped for each subtree and new nodes are placed in. This step is repeated until all the subtrees are done.



**Figure 2.8:** Subdivision problem of the slice-and-dice algorithm, which may create many thin and elongated rectangles. Figure is taken from Bruls [BHV00].

### Squarified Treemaps

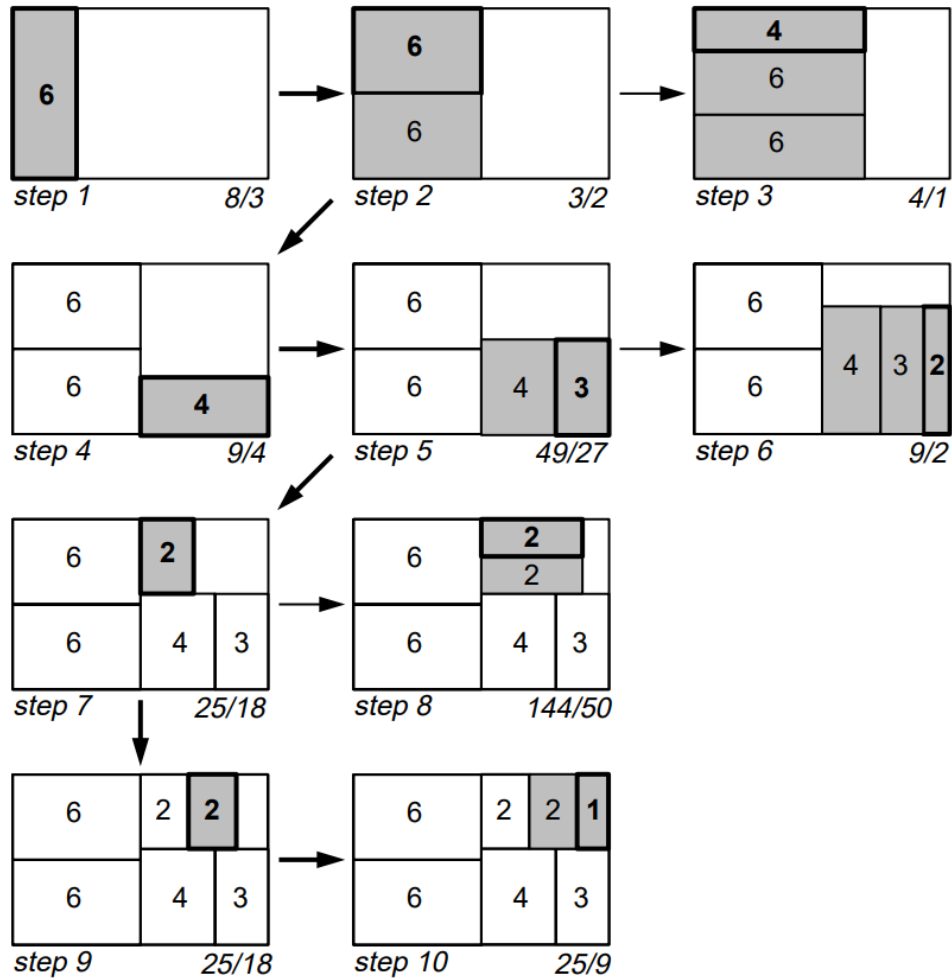
*Squarified Treemaps* maintain better aspect ratio and approximate square shape of the rectangles [BHV00]. Bruls et al. explained the squarified treemap generation algorithm (Listing 2.2) with an example of a rectangle with width 6 and height 4. The rectangle is subdivided either horizontally or vertically into seven more rectangles with areas 6, 6, 4, 3, 2, 2, and 1. Thus thin rectangles appear, with aspect ratios of 16 and 36, respectively as shown in Figure 2.9.

```

procedure squarify(list of real children, list of real row, real w)
begin
  real c = head(children);
  if worst(row, w) <= worst(row++[c], w) then
    squarify(tail(children), row++[c], w)
  else
    layoutrow(row);
    squarify(children, [], width());
  fi
end

```

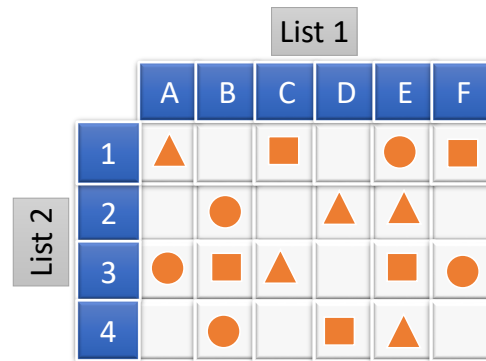
**Listing 2.2:** Squarified Treemap Algorithm



**Figure 2.9:** Squarified Treemap generation procedure proposed by Bruls [BHV00], where subdivision is chosen based on the aspect ratio of the remaining rectangles. In this way, a better aspect ratio is maintained throughout the procedure.

### 2.5.4 Matrix Diagram

According to Burge, a matrix diagram is a visualization technique that allows identifying the presence and contribution of relationships between two or more lists of items or sometimes between the items in a single list. Matrix diagram is a useful tool in information visualization to represent complex many-to-many relationships of varying strengths in a simple straightforward way. It also helps enable interactions and correlation between the items and understand complex dependencies [Bur13].



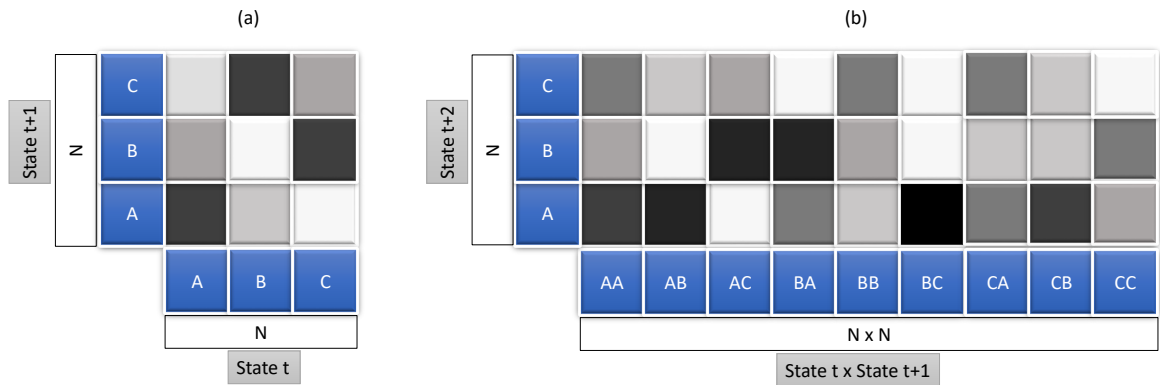
**Figure 2.10:** A simple matrix diagram, where the different shapes of symbol indicate the relationship between respective row and column and white cell indicates no relation, adapted from Burge [Bur13].

According to Burge, there can be various types of contents expressed in a matrix diagram if there is something common between them, which could be data, information, functions, concepts, actions, people, materials, equipments etc.

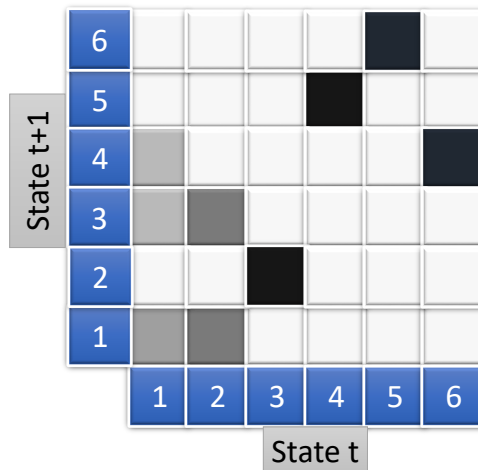
According to Burge, there can be a few basic types of matrix diagrams like L-type, T-type, Y-Type, X-Type, C-Type etc. Figure 2.10 shows an example of L-type matrix diagram.

### 2.5.5 Heatmap

A heatmap is a visualization technique where the value of individual data element contained in a two-dimensional image is represented with color. There can be different types of heatmaps used in different disciplines and each uses different visualization techniques. Heatmaps can be used with the combination of other visualization tools like treemaps, matrix diagrams, grid maps, geographic maps, histograms, process flow diagrams etc. Figure 2.12 shows a general example of heatmaps for first and second transition matrix and Figure 2.11 shows the heatmap representation of transition matrix used in Subsection 2.3.3. Both the example figures are drawn with single color hue proposed by Brewer as shown in Figure 2.13.

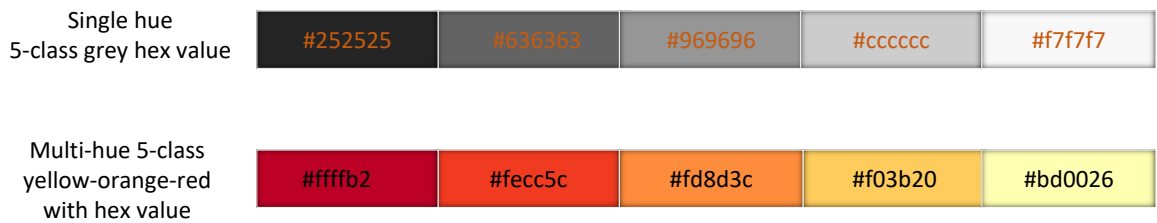


**Figure 2.12:** An example of single color hue heatmap representation for first (a) and second (b) transition matrix. For (a), the x-axis represents the states occurring at timestamp  $t$ , while the y-axis represents the corresponding states at timestamp  $t + 1$ . Similarly for (b), the x-axis carries the states occurring at the successive timestamps  $t$  and  $t + 1$ , whereas the y-axis has the values of corresponding states at timestamp  $t + 2$ .

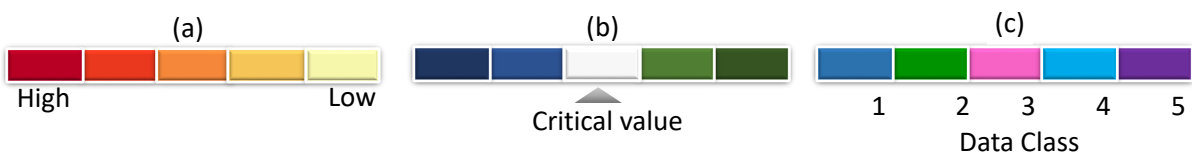


**Figure 2.11:** Single color hue heatmap representation of transition matrix used in Subsection 2.3.3, where darkness level of the cell indicates the transition probability between the respective states and white cell indicates zero transition probability.





**Figure 2.13:** 5-class single hue grey and multi-hue yellow-orange-red color scheme for sequential data, provided by colorbrewer2<sup>1</sup>, and adapted from Brewer et al. [BHH03]

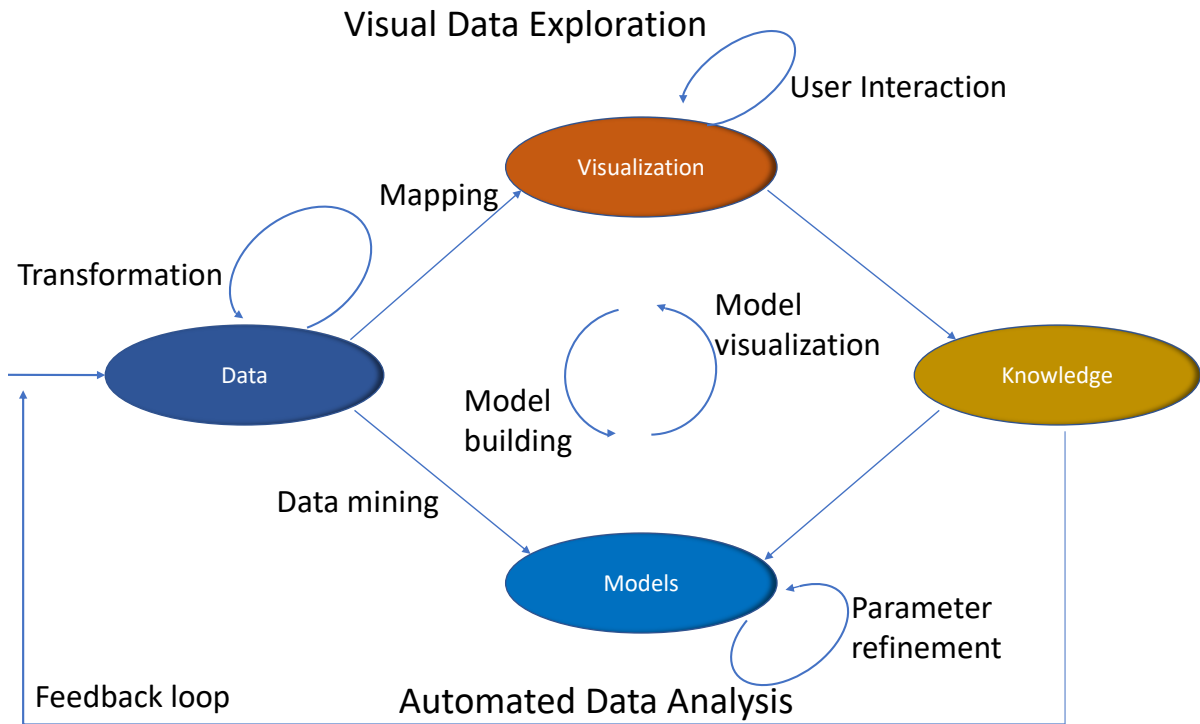


**Figure 2.14:** Three types of color schemes, sequential schemes (a), diverging schemes (b) and qualitative schemes (c), adapted from Brewer et al. [BHH03]

## 2.5.6 Color Theory and Mapping

There are three types of primary color schemes for mapping data [BHH03] as shown in Figure 2.14. Namely:

1. **Sequential schemes:** suitable for ordered data that change from low to high or high to low. Figures 2.13 and 2.14 (a) show the examples of this color scheme.
2. **Diverging schemes:** extreme values are put at both ends of data range, while equal emphasis is put on the mid-range values.
3. **Qualitative schemes:** individual color-coding is chosen for different data classes. This scheme is mostly suitable for categorical data.



**Figure 2.15:** The visual analytics model proposed by Keim. According to him, visual analytics is the process of human interaction between data, visualizations, data models and the users in order to explore knowledge, adapted from Keim [KKEM10].

## 2.6 Visual Analytics

Visual analytics process is the whole knowledge gathering process supported by visual analytics, which is evaluated and measured in terms of efficiency or knowledge gained [KAF+08]. According to Thomas, visual analytics is the science of analytical reasoning, which is facilitated by interactive visual interfaces [TC05]. Keim et al. describe that the automatic analysis techniques are improvised by visual analytics with the help of interactive visualizations to make them effective, easily understandable, reasoning and decision making based on very big and complex data sets [KAF+08]. Figure 2.15 depicts a theoretical overview of various steps, which are represented by ovals and also their transitions, which are depicted by an arrow in the visual analytics process [KKEM10].

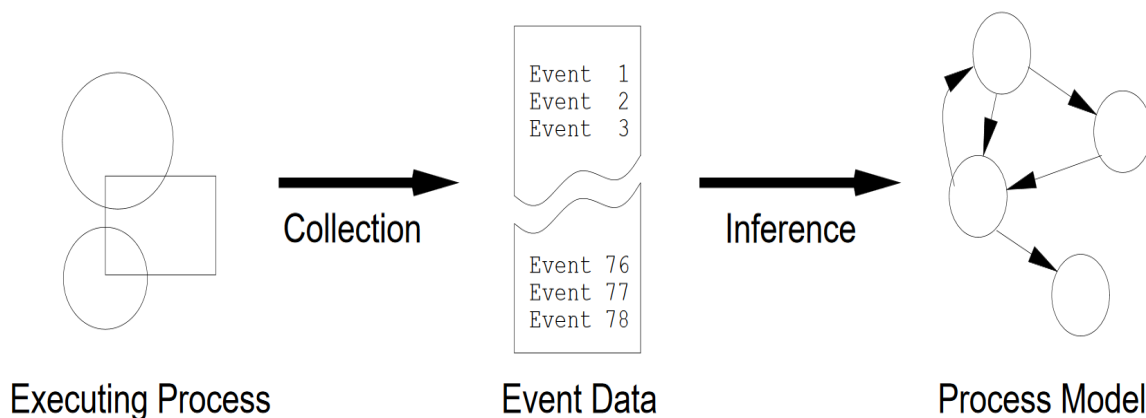
### 3 Related Works

In this section, similar works related to this thesis will be briefly outlined.

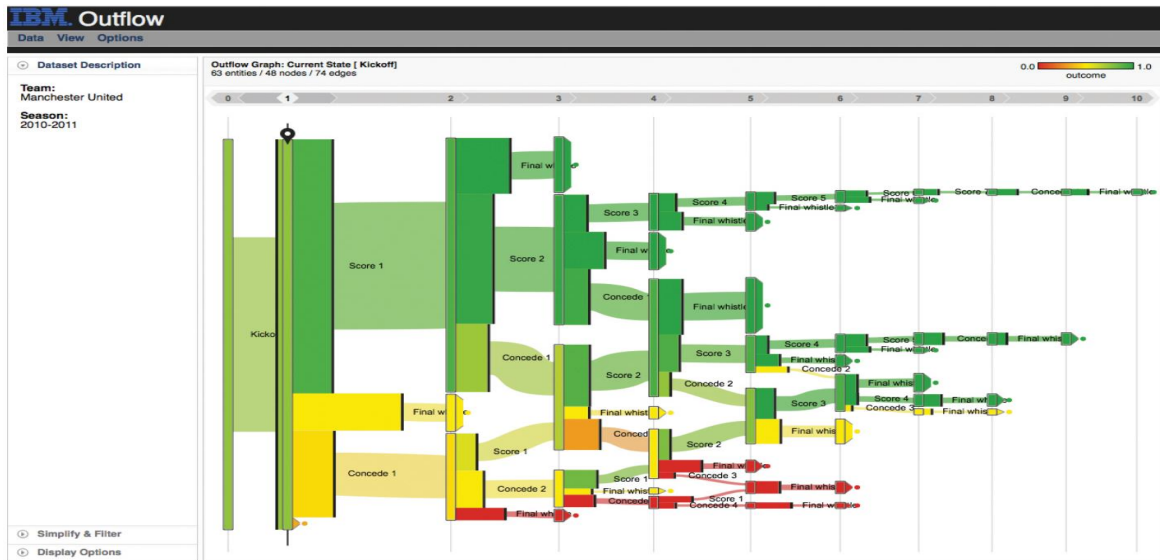
There are various domains where event-based sequence data is very common. Collecting this type of data is very crucial for analysis and future development of the respective domain. There have been researches in this field for more than a decade. This section briefly discusses a few of them.

Cook and Wolf developed a technique named *Process Discovery* [CW98] to analyze event-based data. The process events that describe the data are logged from a continuous process at the first step under this technique. That data are then used to create a formal model of the behavior of that process. Along with *Markov* model, they also used two more methods called *RNet* and *Ktail* and shown comparative results among them. As this uses Markov model too, it should help better understand the problems defined for this thesis. Figure 3.1 shows the capturing process of sequential and event-based data.

*Outflow* [WG12] is another visualization tool that deals with the sequential events and helps find the correlations among them. It can be used to accumulate multiple sequences of events by visualizing the accumulated pathways through different states of events



**Figure 3.1:** Process model discovery to capture sequential and event-based data proposed by Cook and Wolf [CW98].

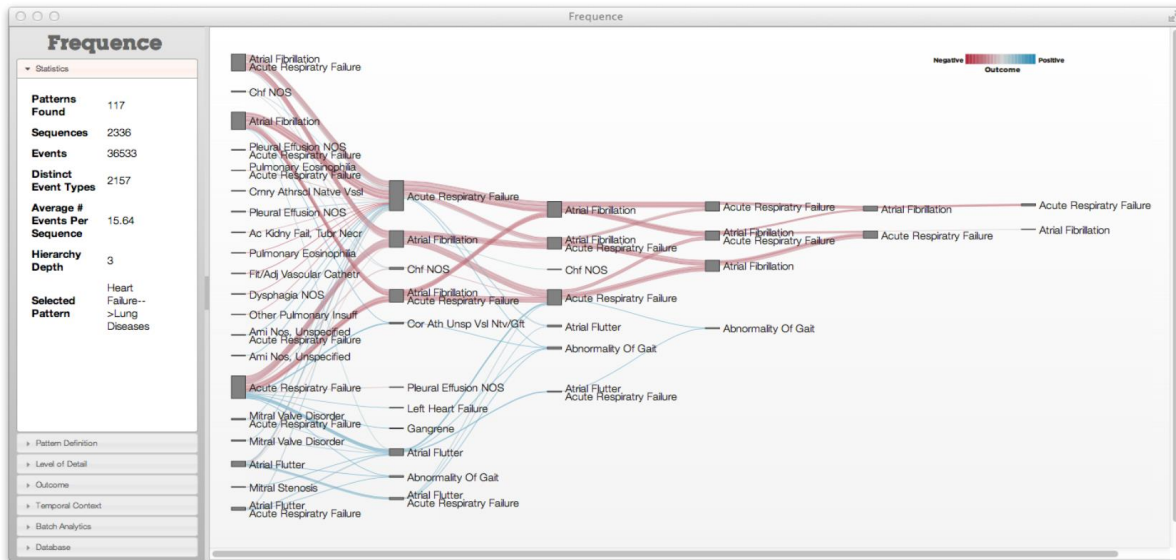


**Figure 3.2:** *Outflow* works with event-based data and graphically represents the progress of events with pathways along with some other information like the outcome, duration, and cardinality. *Outflow* is an interactive visualization tool that enables its users to explore the paths via which different entities arrive and depart different states. This particular screenshot visualizes football club Manchester United’s 2010-2011 season where wins are shown with green pathways while losses are with red [WG12].

with cardinality and timing. It also summarizes the results of pathways and enables user interaction to explore external factors that may be correlated with specific pathway state transitions. Figure 3.2 is a screenshot of *Outflow* that visualizes football club Manchester United’s 2010-2011 season where wins are shown with green pathways, while losses are with red.

*Frequency* [PW14] is an intelligent user interface that is used for data mining and interactive data visualization that helps explore the hierarchical information system and finding most occurring patterns from the sequences of longitudinal events. By using a novel frequent sequence data mining algorithm, it can work with several levels of data, temporal context and concurrency to analyze the outcomes. It also comes with an interactive visual interface that visualizes data insights and users can explore through the different patterns of the level-of-detail data. Figure 3.3 is a screenshot of this application that shows an exploration process of one of the use-cases. *Frequency* is related to this thesis work as this thesis also works with hierarchical data and different patterns of errors to find the correlations among them.

Plaisant et al. proposed a visualization environment, which they termed as *Lifelines* that can be applied to visualize personal event histories and other types of biographical



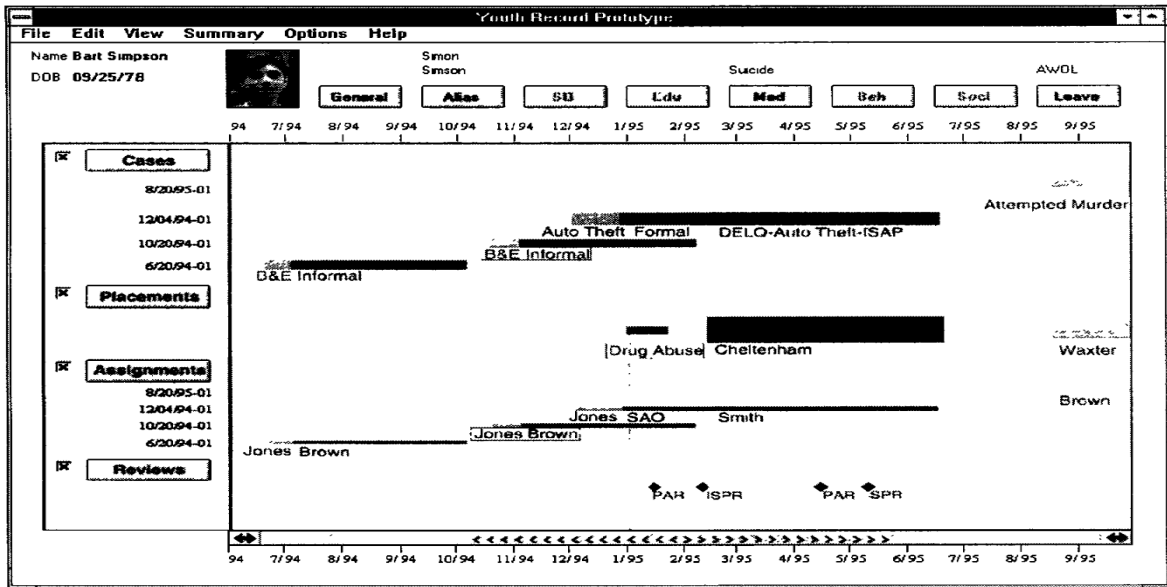
**Figure 3.3:** Screenshot of the exploration process one of the use-cases (lung disease patient dataset in this case) with *Frequence*. How some rough patterns of different symptoms can lead to lung disease and heat failure are shown here [PW14].

data [PMR+96] as shown in Figure 3.4. It can visualize different factors of an individual's key events, for example, relationships, legal cases or health conditions, physician or lawyer consultations etc. *Lifelines* helps minimize the chance of missing important information and enable the users to spot exceptions or trends. Although *Lifelines* works only with life events, it is also related with this thesis work as it helps find useful trends and patterns from sequential events.

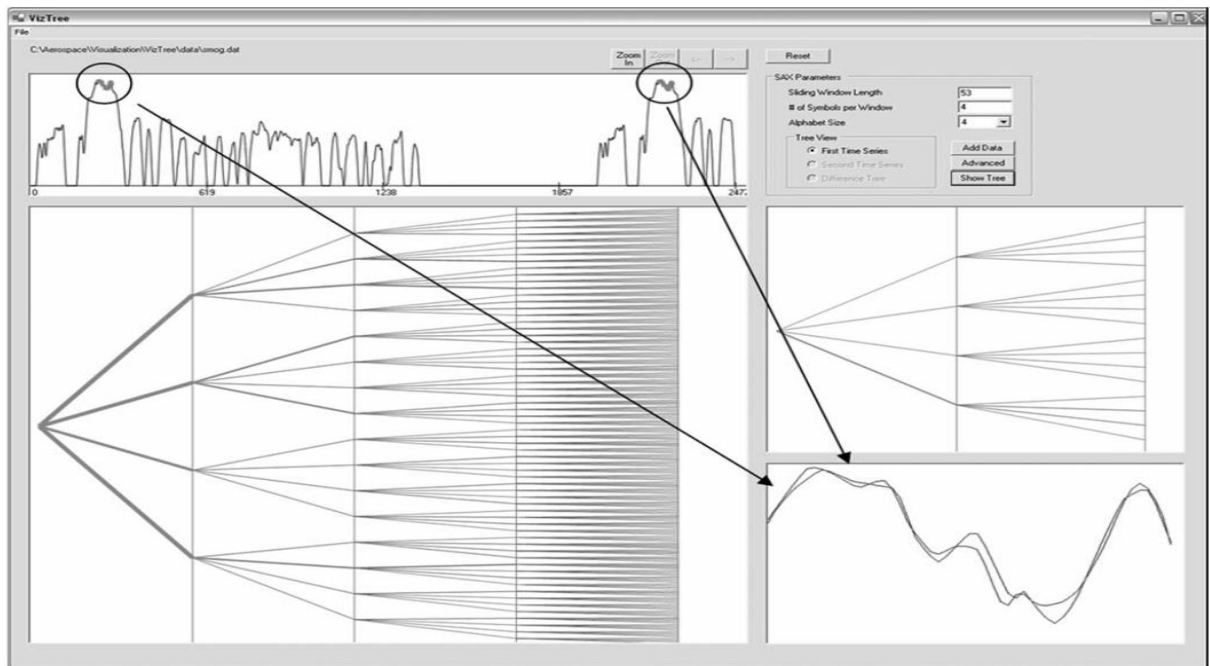
*VizTree* [LKL05] is a visualization system that helps discover pattern and trends from time series data sets based on augmenting suffix trees. *VizTree* enable users interactively explore most and least frequent patterns by visually summarizing the data and encoding them with color and other visual properties. Figure 3.5 is a screenshot of this application. Since this thesis work is also related to finding the different patterns of the time-series error events and visualize them with color encoding and other means of visual analytics process, adapting the tree-like visualization technique might help.

*Timeline Trees* [BBD08] is a visualization tool that deals with hierarchical and sequential data. *Timeline Trees* helps its users visualize advancement of transactions by the buyers in super markets. This is an interactive and hybrid data visualization tool that combines a tree layout and a timeline as shown in Figure 3.6. The tree layout helps sketch the hierarchy, while timeline representing the progression of transactions.

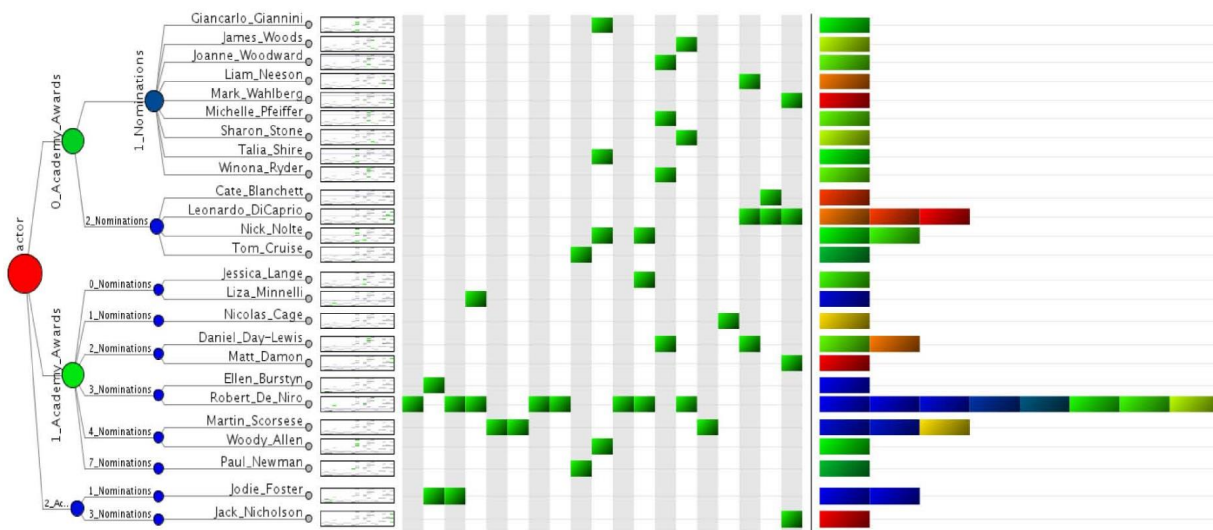
### 3 Related Works



**Figure 3.4:** One of the use-cases of Plaisant’s *Lifelines* is Juvenile Justice records. Cases, locations, employees assigned and available reviews are shown in this screenshot [PMR+96].



**Figure 3.5:** A screenshot of *Viztree* where the input time series is shown in the top panel and the subsequence tree for the time series is shown in the bottom left panel. Parameters can be set on the very right top. The zoom-in feature of the tree is shown on top window, whereas the actual subsequences are visualized on the bottom window [LKL05].



**Figure 3.6:** A screenshot of one of the use-cases of *Timeline Trees* where the movies directed by Martin Scorsese are shown as transactions, while the starring actors as items. Movies are grouped by Academy Award wins and as well as nominations [BBD08].





# 4 Concept

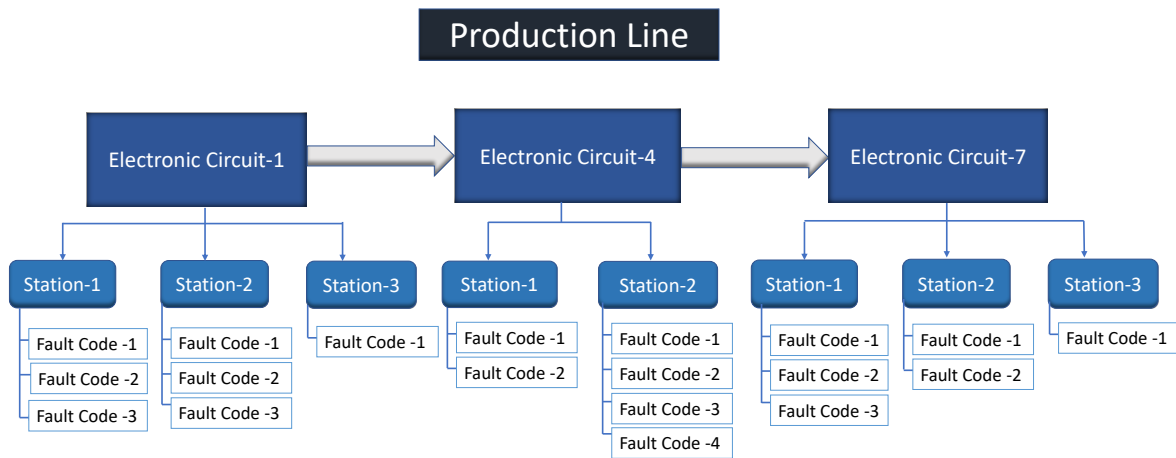
There are an increasing number of errors that occur in the manufacturing line every day. Finding the correlation among those errors and representing them with some visual analytics approaches are the main goals of this thesis. This chapter outlines the core concept and approach to achieve that goal.

## 4.1 Data Analysis

This section describes the core structure and preliminary filtering process of data. This section also outlines the concept of data analysis and preparation process.

### 4.1.1 Structure and Hierarchy of Data

The data consisting of the error reports, which are generated in the production line are structured in hierarchical order. The manufacturing line itself has a tree-like structure where *electronic circuits* being its leafs placed in series. Each *electronic circuit* consists of *stations* placed parallel to each other as shown in Figure 4.1. Whenever there is an error during the operation in any parts of the manufacturing line, it gets logged automatically in the central data source. There are different kinds of errors, which can be identified by *fault-code*. *Fault-codes* are the children of *stations*. The goal is to find out the dependencies and correlations among those *fault-codes* and visually represent that relations with the theory of Markov model.



**Figure 4.1:** Basic structure of production line hierarchy consisting of electronic circuits, stations and fault-codes, where an electronic circuit is the parent of stations and a station is the parent of fault-codes.

#### 4.1.2 Preliminary Filtering of Data

Although the number of errors logged from the production line is huge, not every error has the same importance. After the calculation of durations the fault-codes from the timestamp, it was noticed that a big portion of errors last for less than a few seconds or sometimes even less than a second. On the other hand, because the same error may cause itself, there may be some errors, which are recorded occurring forever, which is also not realistic. At the same time, there may be some errors, which get generated because the associated station or circuit may be broken, or sometimes the error reports are not even actual errors rather minor warnings or some basic status information. So, as a part of preliminary filtering, all these types of error reports could be discarded as per the recommendations from the domain experts.

The data could be filtered in two phases. Initial filtering could be done in the raw data source itself before sending them for further calculation and final preparation. The following parameters from the data source are useful to calculate the correlation and dependency among the errors:

- **Electronic Circuit:** the electronic circuit is the top most element in the production line. Each electronic circuit has its *ID*, *name* and physical *position* in the production line.

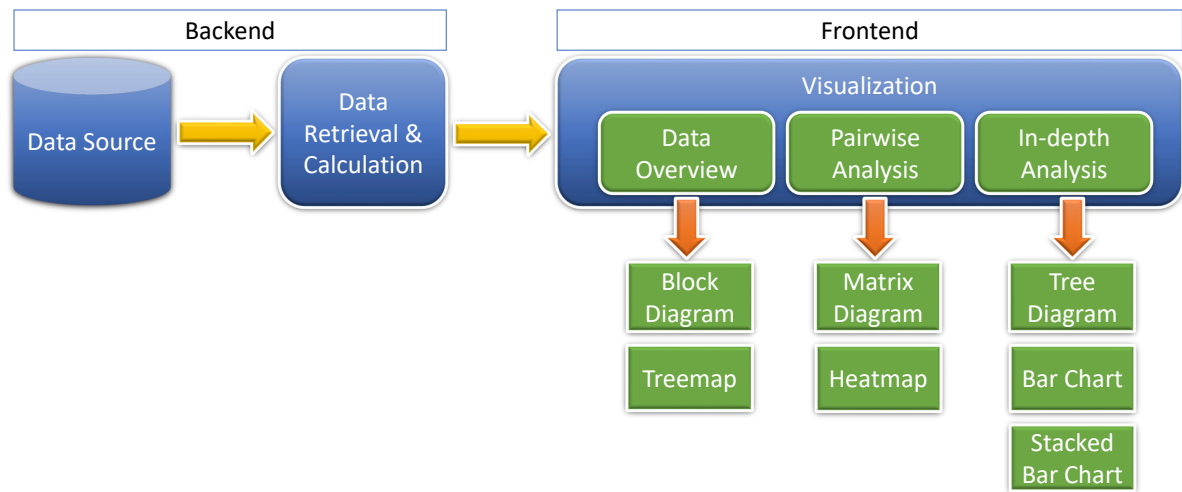
- **Station:** as shown in Figure 4.1, each electronic circuit consists of a few stations. The most important information about station is its ID and corresponding electronic circuit.
- **Fault-Code:** each error occurs in the production line is termed as fault-code. Each fault-code has its ID, corresponding station and electronic circuit.
- **Fault-Code Duration:** the occurring duration of a fault-code meaning how long a particular error lasted.
- **Timestamp:** the exact time of an error when it initiated.

## 4.2 Design and Architecture Concept

This section briefly describes the main design and architecture concept. To implement the desired prototype for analyzing the correlation of errors, there are two main areas need to be considered namely **Backend** and **Frontend**. These areas are briefly discussed in the following subsections.

### 4.2.1 Calculation and Preparation of Data

Retrieval, calculation, and preparation of data could be done in the **Backend** as shown in Figure 4.2. After the initial filtering, data source could be connected to the calculation and preparation process. Markov chain and transition matrix algorithm as discussed in Sections 2.2 and 2.3 could be applied on the data to calculate the correlations and multi-order dependencies. Various models could be developed to prepare the data for different types of visualizations.



**Figure 4.2:** This is the core idea of this thesis work. After the calculation and retrieval process, there can be three main types of data visualization. (1) Data overview visualization could provide quick insights about the data. Then to analyze the data further, (2) pairwise visualization like matrix diagram and (3) in-depth visualization like tree diagram could be used.

## 4.3 Visualization

**Frontend** could deal with all the visualization concepts. This section outlines different visualization proposals and concepts that might help find the correlations and dependencies among the errors in the production line. There can be three major types of visualizations namely:

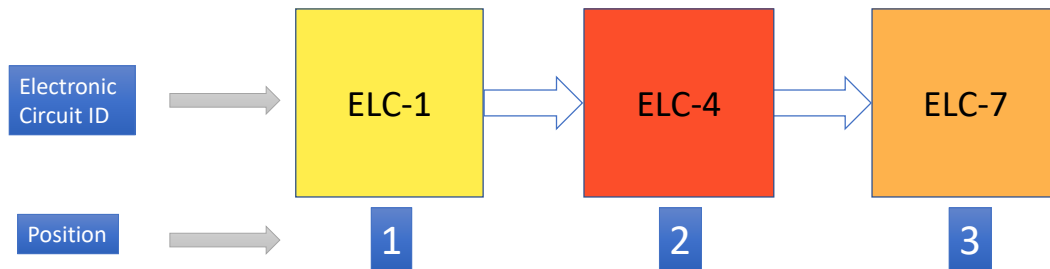
1. Visualization for data overview.
2. Visualization for pairwise analysis and
3. Visualization for in-depth analysis.

### 4.3.1 Data Overview Visualization

Data overview visualizations might help get insight about the retrieved data. One overview visualization might be a simple block diagram and another could be used to visualize the hierarchy of data and filter them according to the demands. Both types of overview visualizations are described in the following subsections.

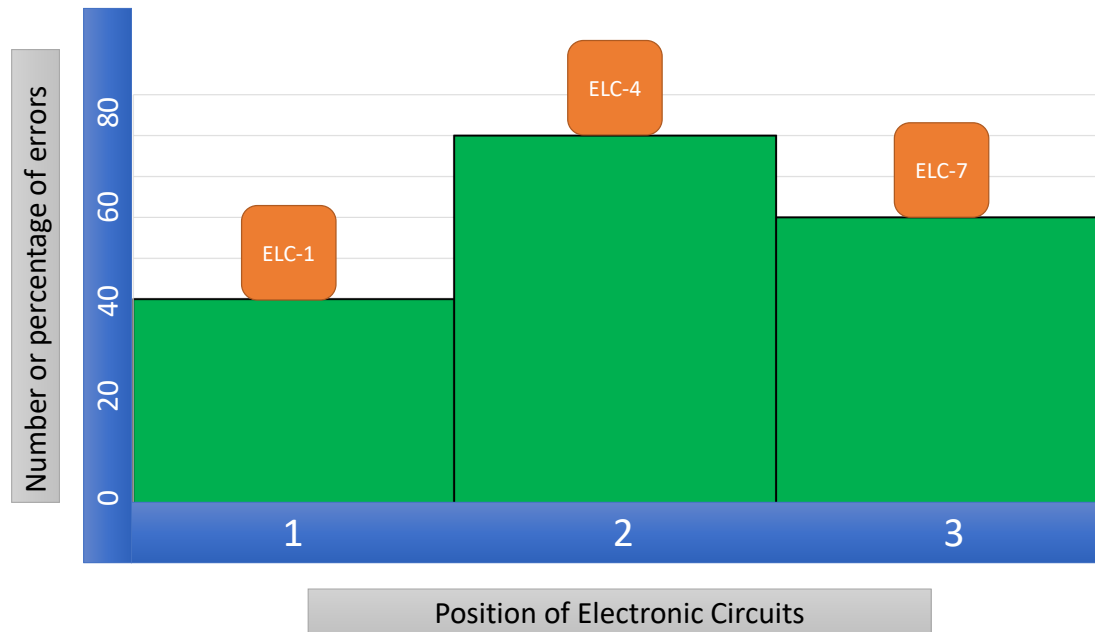
## Block Diagram

As discussed in Subsection 4.1.1, there are several electronic circuits in a production line placed in series. A simple block diagram might help realize their actual position where each block might indicate an electronic circuit labeled with information like its ID and corresponding position, the number of errors associated with this circuit etc.



**Figure 4.3:** The concept of a block diagram to visualize the positions of electronic circuits, where each block indicates an electronic circuit along with its ID and position. Blocks can also be color-coded based on the amount of errors associated with the respective electronic circuit. Color-coding can be applied based on the theory described in Subsection 2.5.6.

A histogram or bar chart could also help visualize the positions of electronic circuits. The height of each bar could show the number or percentage of errors for the respective electronic circuits. Figure 4.4 shows the bar chart representation of block diagram shown in Figure 4.3.

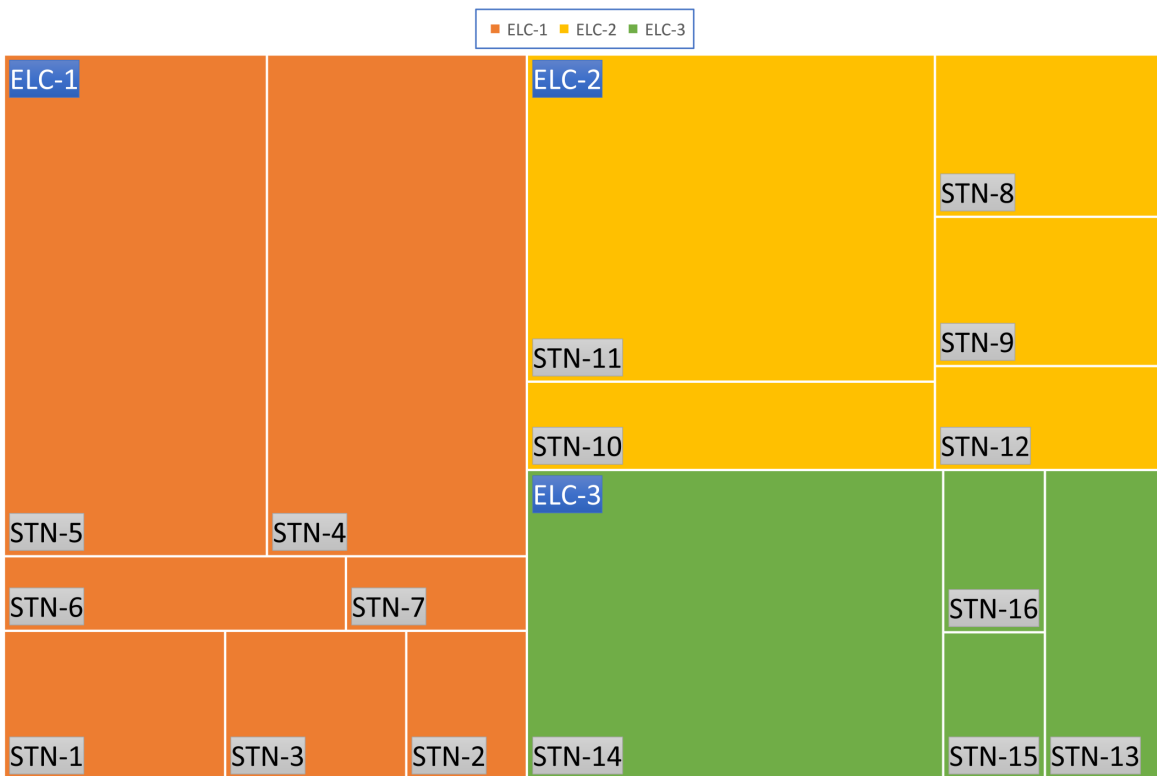


**Figure 4.4:** The concept of a block diagram with bar chart to visualize the positions of electronic circuits, where the height of each bar might show the amount of errors associated with the respective electronic circuit. Bars can be placed according to the actual position of the electronic circuit.

### Treemap

A treemap described in Subsection 2.5.2 could solve the problems related to the shortage of space and visual clutter for the bigger datasets. It could also be useful to exhibit insights about the elements of the production line. In limited space, treemap can be implemented with a few levels depending on the depth level of the production line. For this thesis work, it could be implemented up to level 4 as explained below:

- at depth level-0, the production line itself could be visualized, while electronic circuits being its children.
- at depth level-1, the electronic circuits could be parents, while associated stations could be shown as its children.
- at depth level-2, the stations could be parents, while corresponding fault-codes being its children.
- at last level, only the fault-codes can be shown with the rectangles.



**Figure 4.5:** The concept of a treemap to visualize the overview of production line elements like electronic circuits and stations. Size of each cell may define the amount of errors associated with the respective electronic circuit and station.

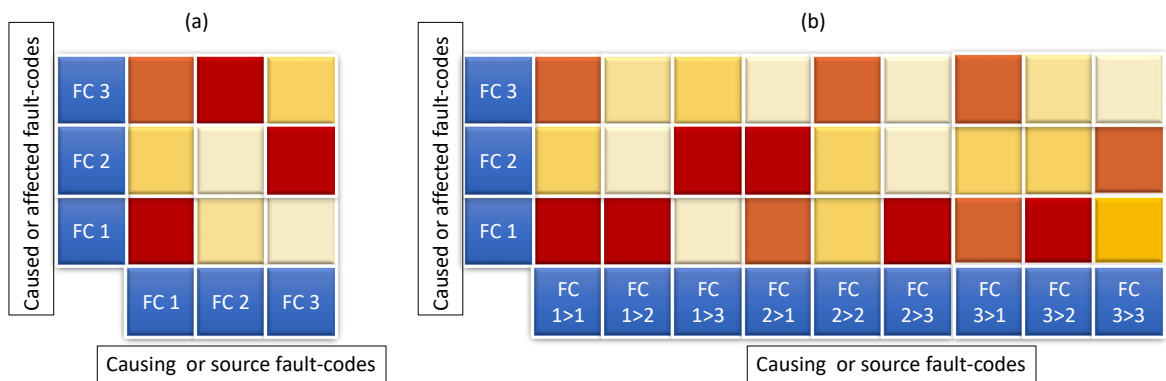
The size of each rectangle could be determined by the number of errors associated with the elements of production line i.e., electronic circuits, stations or fault-codes. To visualize the elements efficiently some scaling technique could be applied so that even if the difference between the largest and the smallest element gets higher, they could still be identified. Another advantage of using treemap would be using it for filtering and zooming into a particular element of the production line, which might help analyze the error correlations for that particular element. Figure 4.5 shows an example of a conceptual treemap, where stations (STN) are the children of electronic circuits (ELC).

#### 4.3.2 Visualization for Pairwise Analysis

After the overview visualization, data could be presented for more analysis. Heatmaps on top of transition matrix diagram could be useful for pairwise analysis of the errors.

Transition Matrix Diagram and Heatmap

Heatmaps described in Subsection 2.5.5 could be useful to visualize the probability of transition from one fault-code to another, which can be calculated using Markov chain and transition probability algorithm described in Subsection 2.2.2 and Section 2.3. It is possible to draw heatmaps both for first and m-order transition matrix. Each cell of heatmaps could be populated with transition probability value of corresponding causing fault-code and caused fault-code. Color-coding could also be applied to the cells reduce space related complexities. Figure 4.6 shows the concept of heatmaps for first (a) and second (b) order transition matrix with three fault-codes i.e., FC-1, FC-2 and FC-3.



**Figure 4.6:** The concept of the heatmaps for first (a) and second (b) order transition matrix to visualize the correlations of fault-codes.

Heatmap for first order can have  $N \times N$  number of cells in the transition matrix, while for the second order,  $N^2 \times N$ , for third order  $N^3 \times N$  and so on i.e., the number of cells in one dimension increases exponentially, which, for bigger datasets, might increase the complexity.

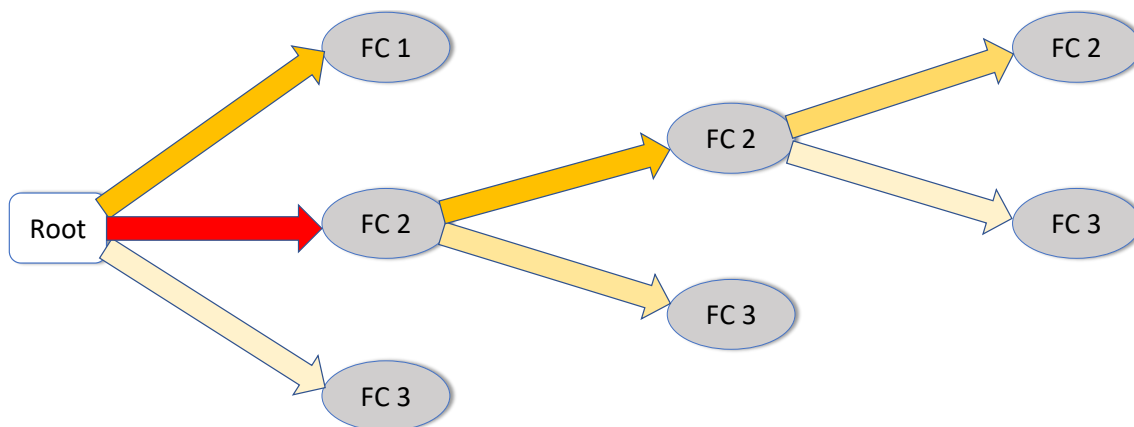


### 4.3.3 Visualization for In-depth Analysis

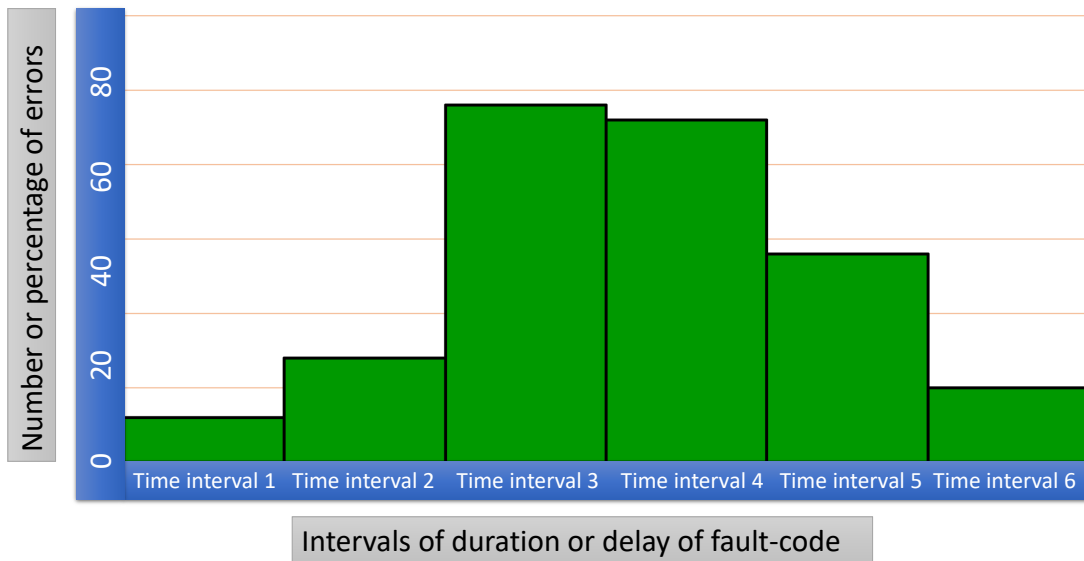
To minimize the problem to visualize the correlations for the errors from the bigger datasets and for in-depth analysis, tree diagram could help.

#### Rooted Tree Visualization

Rooted tree visualization as described in Subsection 2.5.1 could visualize the correlations among the different fault-codes, where root node can be shown as the source or cause of errors, while the children nodes can be shown as the caused or affected fault-codes. The tree could be expanded up to a few levels with user-interaction. Markov chain algorithm could be used here too for tree expansion. The edges of the tree can be color-coded based on the number of error for the respective transition. Sequential color-coding can be applied based on the theory described in Subsection 2.5.6. Figure 4.7 shows an idea of a tree visualization to show multi-order dependencies and correlations.



**Figure 4.7:** The concept of the tree visualizations to visualize the correlations among the fault-codes. The parent nodes are the causing or source fault-codes and the children nodes are the caused or effected fault-codes.



**Figure 4.8:** Concept with bar chart to visualize the amount of the fault-codes for a specific time interval or delay between successive fault-codes.

### Bar Chart

Each fault-code has a corresponding duration field in the data source as mentioned in Subsection 4.1.2. A histogram or bar chart could be used to visualize the number of errors for a specific interval of time, which might help an analyst realize how long the errors last as shown in Figure 4.8.

A bar chart could also visualize the distribution of time delay between successive fault-codes. Timestamp information of the successive fault-codes could be collected from the nodes of the tree diagram mentioned in Figure 4.7.

Intervals of time could be defined based on the theory related to Choropleth Maps of graded color series and nominal data [ilstu17]. According to this theory, there can be the following three types of intervals [CS11]:

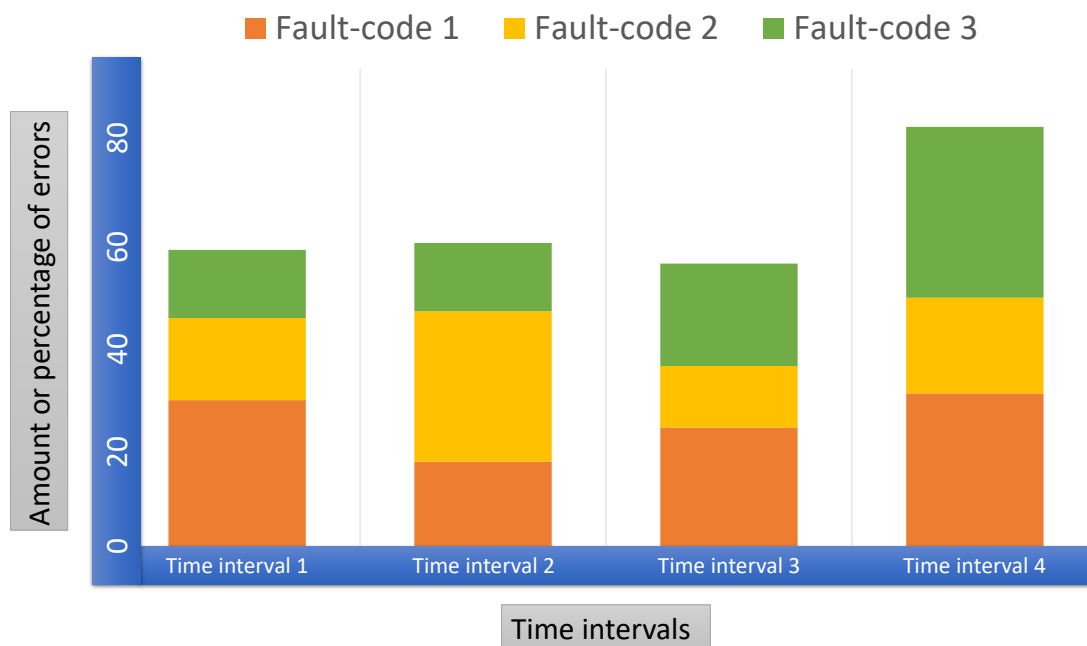
1. **Equal Interval Classification:** the whole data range is equally divided into  $n$  intervals.
2. **Quantile Classification:** interval size depends on the instances of data, so that each interval has approximately equal amount of data.

- Natural Breaks Classification:** intervals are made based on the natural peaks of the value distribution.

One of the classifications can be adapted for the bar chart interval distribution after observing the output of the given data.

### Stacked Bar Chart

Stacked bar chart could also be used to show the fault-code distribution for a particular station or electronic circuit. Time intervals of fault-code durations could also be shown using this chart. Figure 4.9 shows the concept of analysis of fault-code distribution with stacked bar chart. The main disadvantage of this type of chart is that only the values closer to the x-axis are easily comparable observing their value in the y-axis. The other values in the same bar are difficult to compare because of their variable starting positions.



**Figure 4.9:** Concept with stacked bar chart to visualize the contribution of the fault-codes for a specific time interval.



# 5 Implementation

This chapter discusses the implementation details of this thesis work, which includes most of the concepts discussed in Chapter 4. To visualize the correlations and higher order dependencies among the errors in the production line, a web application named *Correlation Analyzer* is implemented. This chapter describes what visualizations are implemented and how they are implemented. This chapter also outlines the analysis and interaction process of the errors of the production.

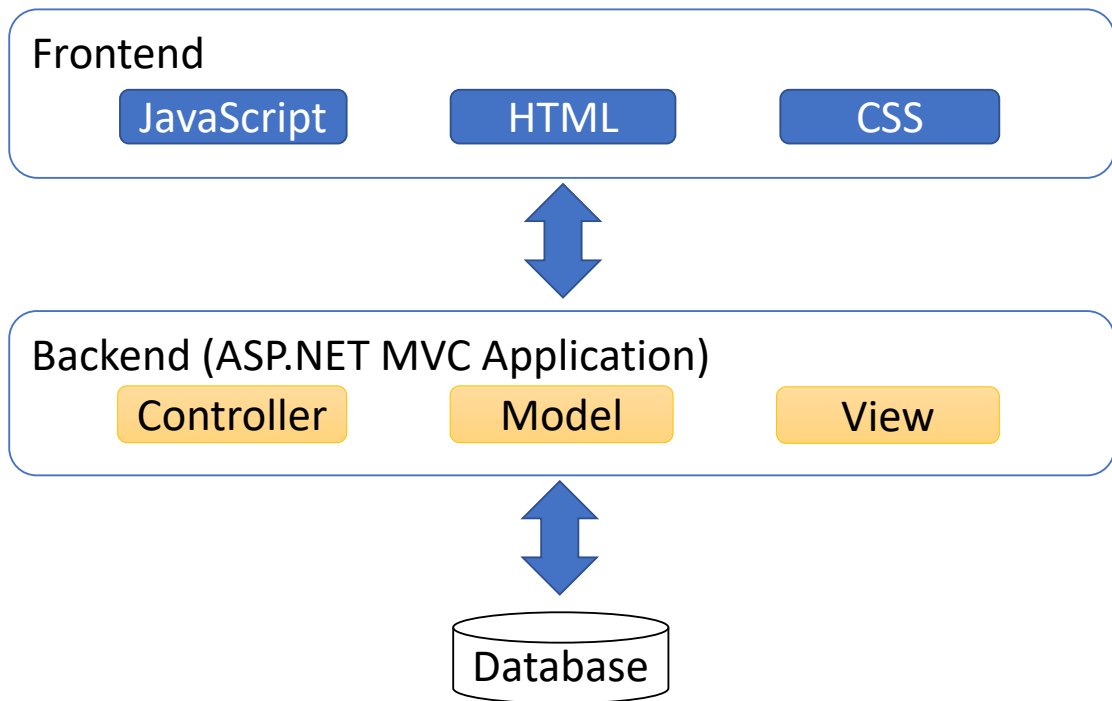
## 5.1 Architecture and Models

This section states the core models and architecture of the thesis. Figure 5.1 shows the core model and architecture of the web application implemented for the thesis.

### 5.1.1 Adaption of Frameworks

The core frameworks and libraries adapted in this thesis are as the following:

- **ASP.NET MVC:** The web application implemented for this thesis was developed in C Sharp (C#) with ASP.NET MVC (.NET version 4.5.2), which is a web application framework developed by Microsoft and implemented with Model-View-Controller (MVC) pattern.
- **MS SQL Server:** A copy of a database from the production line of a manufacturing company is the data source for the thesis, which is hosted in Microsoft SQL Server. The database fields and view are discussed in Subsection 5.1.2.
- **D3, Plotly, and Other JS:** This thesis uses JavaScript libraries like D3 and Plotly that help create dynamic and interactive data visualizations with the help of SVG, HTML and CSS. Along with these libraries, this thesis also uses libraries like *Bootstrap*, *jQuery*, *d3-interpolate*, *d3-context-menu*, *alertify*, *colorbrewer* etc.

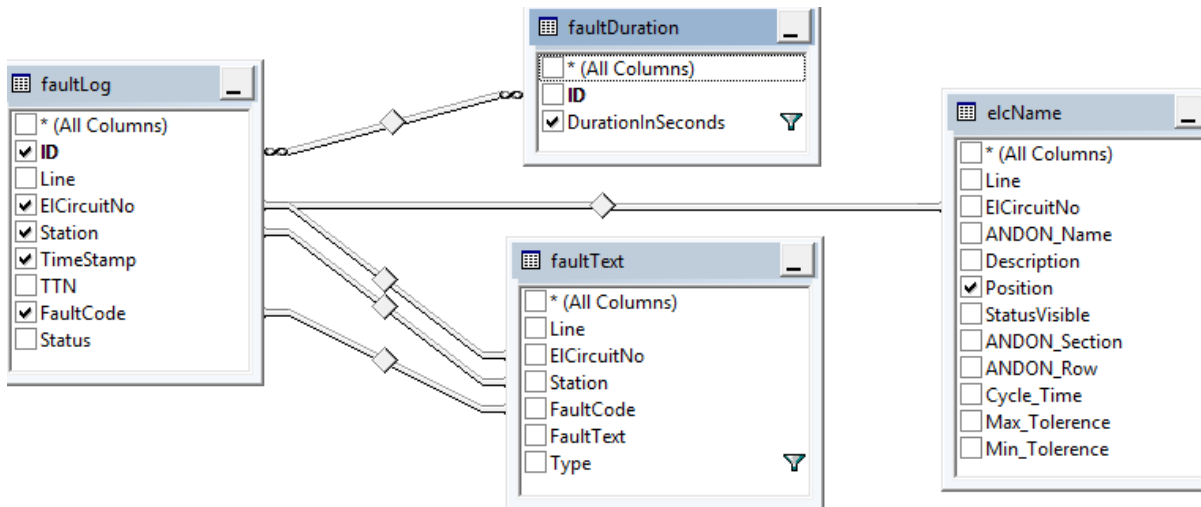


**Figure 5.1:** The core model and architecture of the web application implemented for the thesis. The prototype is implemented in ASP.NET MVC Application pattern.

### 5.1.2 Database Fields and View

A database-view is created with the fields only required for the thesis, which is shown in Figure 5.2. The related columns are briefly discussed below.

- **ID:** the primary key of the database, determines an occurrence of error.
- **ElCircuitNo:** this is the identifier of an electronic circuit associated with the error.
- **Position:** determines the physical position of corresponding electronic circuit in the production line.
- **Station:** this is the identifier of station corresponding to electronic circuit.
- **TimeStamp:** the exact time of an error when it initiated.
- **FaultCode:** the identifier of the error itself.
- **FaultCodeDurationInSeconds:** the total time span of the associated error.



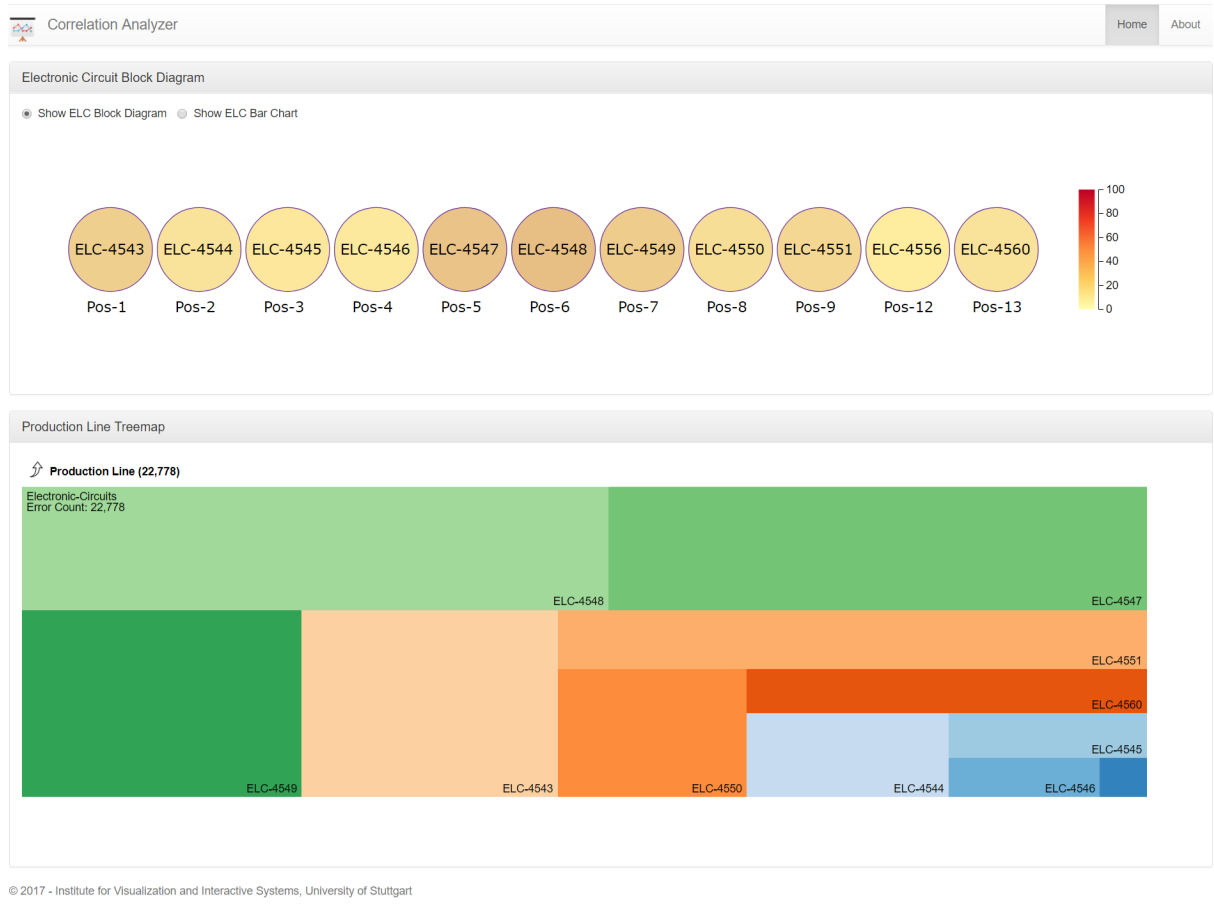
**Figure 5.2:** A Screenshot of the database view. The columns that are used for the thesis are *ID*, *ELCircuitNo*, *Position*, *Station*, *TimeStamp*, *FaultCode*, *FaultCodeDurationInSeconds*.

### 5.1.3 Core Models

There are a few core models implemented in the back-end, which create the foundation of the prototype implemented for this thesis. The models are as the following:

- **Production Line Elements Model:** defines the architecture, fields and the behavior of different elements of the production line. The three most important elements of the production line are the electronic circuit, station, and fault-code. As they have some common characteristics and are used very often in different implementations, they are included in this model.
- **Electronic Circuit Position Model:** defines the overview of the electronic circuits and their positions. Additional information like the number of errors for the respective electronic circuits and contribution percentage to the total amount is also defined in this model.
- **Markov Chain Transition Matrix Model:** calculates the values required to create transition matrix based on the Markov chain algorithm. The values include the transition probabilities, the number of causing and caused fault-codes etc.
- **Fault-Code Model:** defines the fields related to a fault-code. It helps create the distribution for the durations and delays of fault-codes.

## 5 Implementation



**Figure 5.3:** The home page of the *Correlation Analyzer* with two data overview visualizations. User interactions displays the analysis of error with the other visualizations.

### 5.2 Implementation of Visualizations

All three main types of visualizations discussed in Section 4.3 are implemented in this thesis. This section describes how these visualizations are implemented and how they can be used to analyze the correlations and dependencies among the errors. Figure 5.4 shows the complete view of the implemented web application with all the visualizations displayed.



## 5.3 Implementation of Data Overview Visualizations

This section describes the generation, implementation, and exploration of the data overview visualizations. All three data overview visualizations i.e., electronic circuit block diagram, electronic circuit bar chart, and treemap as discussed in Subsection 4.3.1 are implemented. Figure 5.3 shows the home page of the implemented web application with two data overview visualizations.

### 5.3.1 Electronic Circuit Block Diagram

As discussed in Subsection 4.1.1 and shown in Figure 4.1, the electronic circuits are placed in certain order in the production line. To better understand the physical position of electronic circuits in the industrial production line, two simple block diagrams are implemented. Both the diagrams make use of back-end *Electronic Circuit Position Model* as discussed in Subsection 5.1.3.

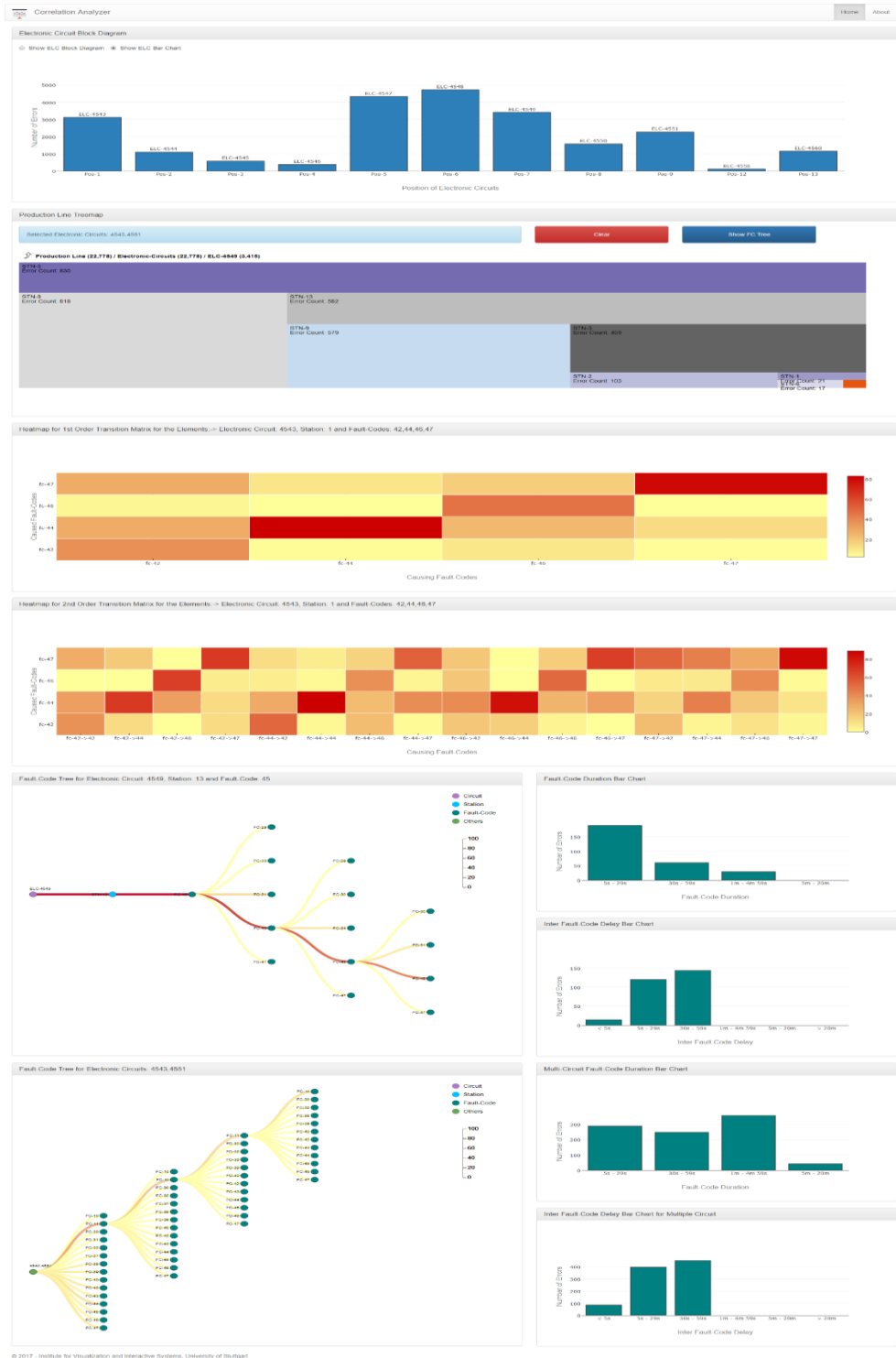
One of them is drawn with circles as shown in Figure 5.5 b). The *Electronic Circuit Block Diagram* panel on the home page of the *Correlation Analyzer* displays this visualization. Circles are placed sequentially according to the actual position data from database where each of the circles indicates an electronic circuit. Yellow-orange-red sequential colors schemes as described in Subsection 2.5.6 visualizes the contribution of errors to the whole production line. A legend is placed on the right side of the visualizations to display the scale of the used color-code. The tooltip provides additional information like the number of errors, its contribution percentage etc.

Another visualization that helps visualize the positions and error contribution is created with a bar chart. A click on the *Show ELC Bar Chart* radio button, which is also on the *Electronic Circuit Block Diagram* panel on the home page of the *Correlation Analyzer* displays this visualization. As shown in Figure 5.5 c), each bar indicating the position of an electronic circuit is placed vertically in the x-axis based on its position, while the height of the bars in y-axis indicates the number or percentage of errors for the associated electronic circuit. Like the block diagram with circles, the tooltip provides with information like the number of errors and its contribution percentage.

### 5.3.2 Treemap

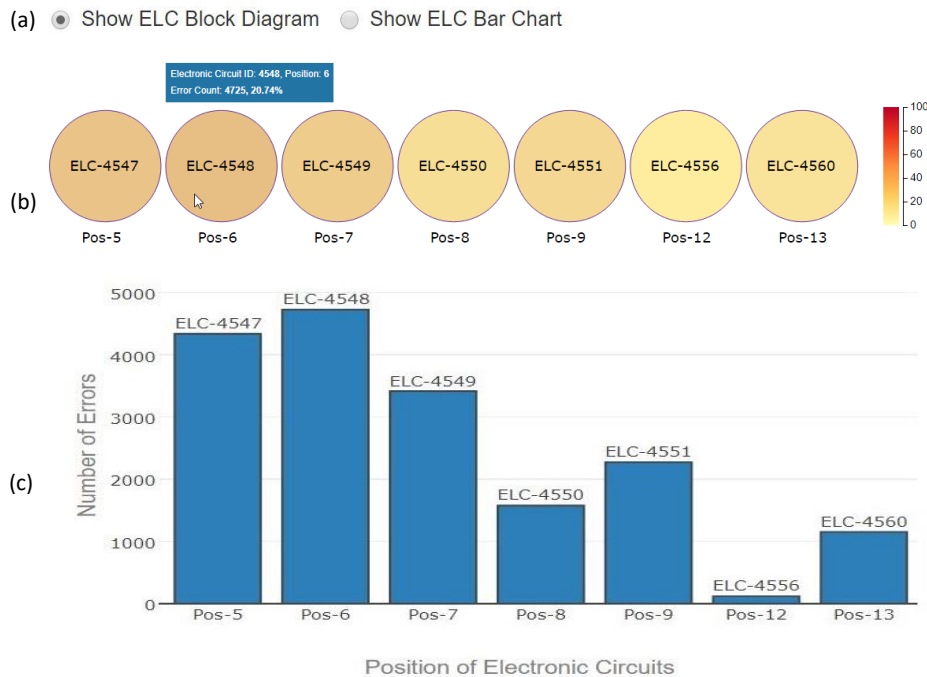
To visualize this hierarchical structure and browse through different production line elements, a squarified treemap as discussed in Subsections 2.5.2 and 4.3.1 is implemented based on the back-end *Production Line Elements Model* as described in Subsection 5.1.3.

## 5 Implementation



**Figure 5.4:** The Screenshot of complete view of the web application with all the visualizations displayed. The top most visualization is the ELC block diagram and ELC bar chart followed by the treemap, heatmaps for first and second order transition matrix, FC tree along with related bar charts and multi-circuit FC tree along with bar charts.

## 5.3 Implementation of Data Overview Visualizations



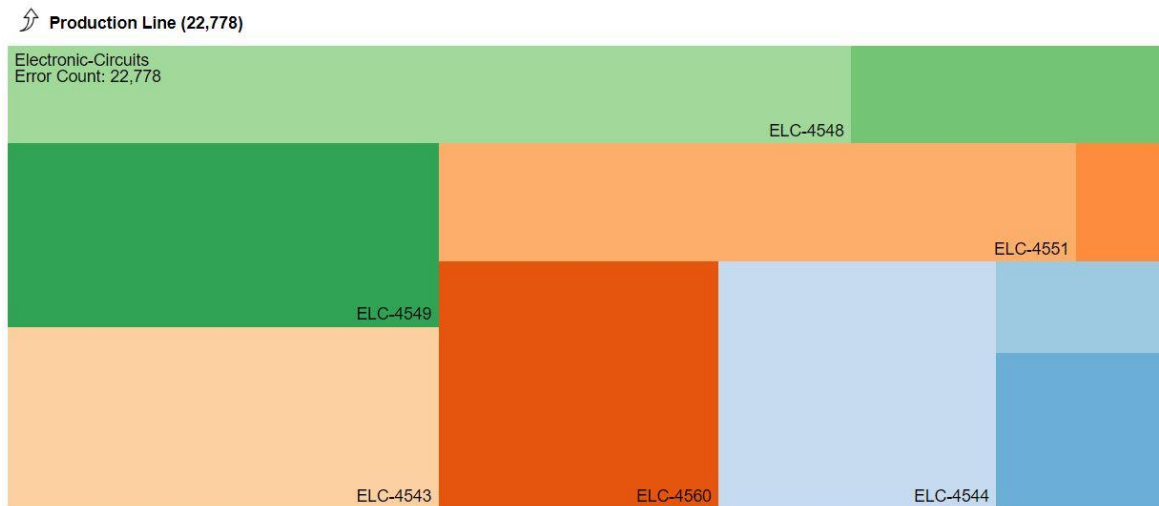
**Figure 5.5:** The screenshots of the implemented electronic circuit block diagrams to display the positions of the electronic circuits. Part (a) shows the radio buttons to select one of them, part (b) is a partial screenshot the block diagram with circles and part (c) shows the ELC bar chart. The tooltip provides with additional information in both the visualizations.

Treemap generation code is adapted and customized d3 code by Mike Bostock<sup>1</sup>. *Production Line Treemap* panel in the *Correlation Analyzer* displays this visualization. Only the rectangles corresponding to the electronic circuits are visible on top-most level or at depth-0 as shown in 5.6. The circuits can be distinguished with the size, color, and label of the associated rectangle. Users can explore a circuit further by clicking anywhere on them. One click exposes the stations associated with the respective circuit, while clicking again on a station reveals all the fault-codes under the respective circuit and station as shown in Figure 5.8. To visualize each and every element efficiently, logarithmic scaling is applied so that even if the difference between the largest and the smallest element gets higher, they are still visible in the treemap. At every level, the total number of errors for that level is shown in the upper-left corner of the rectangle, while the label of the rectangle is depicted on the lower right corner, provided that the required amount of space is there. Colors of the rectangles are used with a view to distinguishing the rectangles from each other. Context-menu is implemented for each level in the

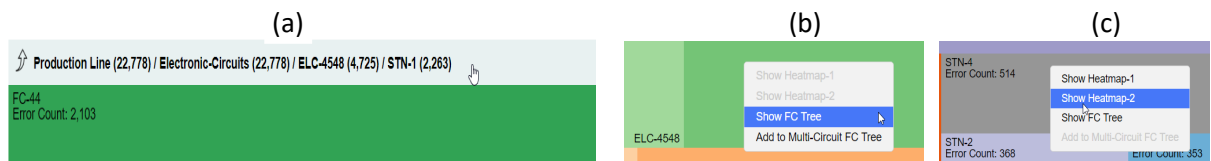
<sup>1</sup><http://bost.ocks.org/mike/treemap/>

## 5 Implementation

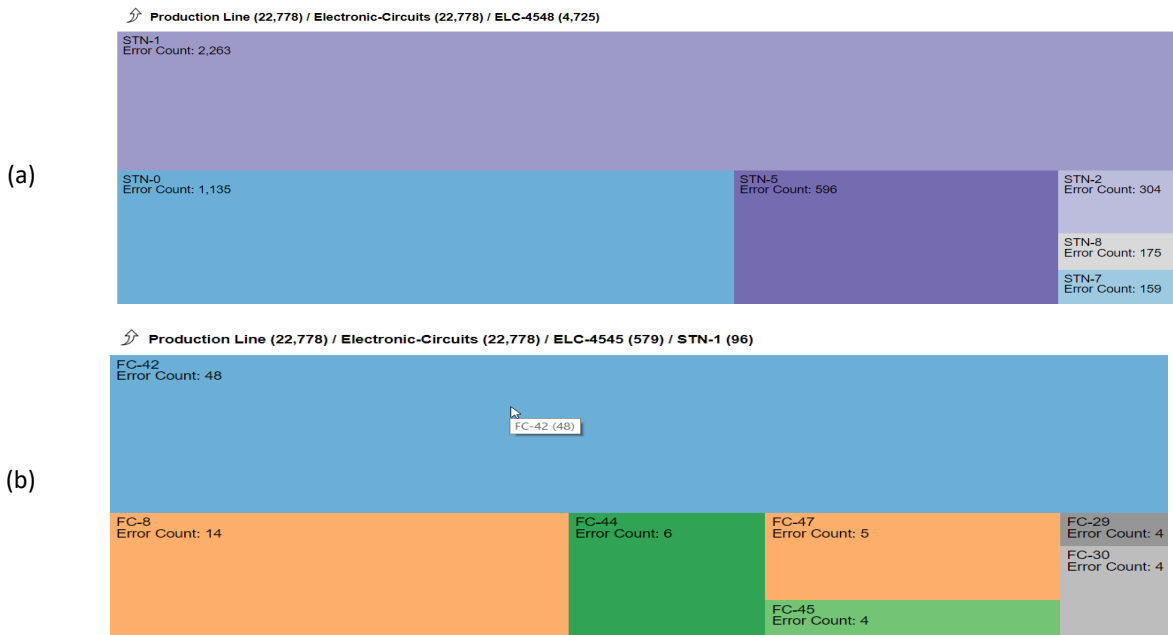
treemap, which shows the options to initiate analysis of errors with more visualizations like i.e., heatmap for first and second order transition matrix diagram, fault-code tree, multi-circuit fault-code tree as shown in Figure 5.7, which are described further in the following subsections.



**Figure 5.6:** The screenshot of the production line treemap at depth-0, where only the rectangles corresponding to the electronic circuits are visible. The other elements can be explored with user-interaction.



**Figure 5.7:** The navigation bar (a) and the available options in the context-menu (b, c) of the treemap at different hierarchy levels.



**Figure 5.8:** (a) Station (b) Fault-code view of the treemap that can be explored at depth-2 and depth-3.

## 5.4 Pairwise Analysis Visualizations

This section describes the generation, implementation, and exploration of the visualizations for pairwise analysis.

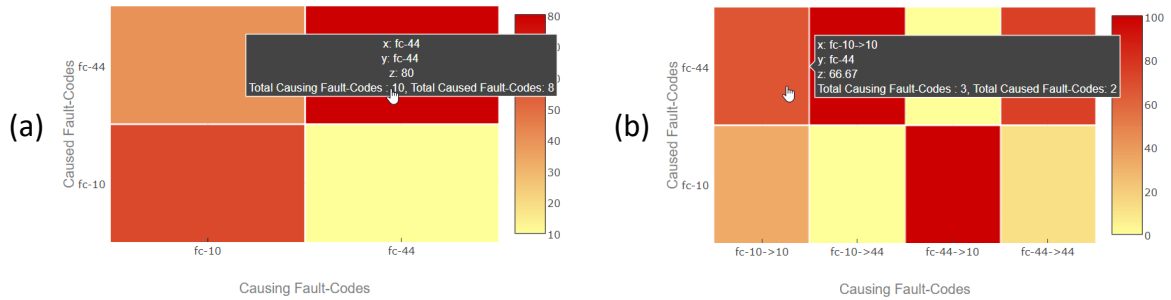
### 5.4.1 Generation of Transition Matrix

To find out the pairwise correlations among fault-codes, heatmaps both for first and second order transition matrix as defined in Subsection 4.3.2 are implemented with the help of *Markov Chain Transition Matrix Model* as described in Subsection 5.1.3.

### 5.4.2 Analysis with Heatmap for First Order Transition Matrix

To explore this visualization, one is to click on *Show Heatmap-1* from the options of the context-menu of the treemap on any station or fault-code rectangle. *Heatmap for 1st*

## 5 Implementation



**Figure 5.9:** The screenshots of the implemented heatmaps for first (a) and second (b) order transition matrix. The tooltip provides with additional information.

*Order Transition Matrix* panel of *Correlation Analyzer* displays this visualization. The associated electronic circuit ID, station ID, and fault-codes are displayed in the panel header. The source or initiating fault-codes, which are labeled as *causing* fault-codes are placed on the x-axis, while the affected or following fault-codes, which are labeled as *caused* fault-codes are placed on the y-axis. The probability of occurrence of a caused fault-code (y-axis) after a causing fault-code (x-axis) is color-coded by yellow-orange-red sequential schemes as described in Subsection 2.5.6 i.e., the darker the color, the higher the co-occurrence probabilities. The tooltip provides with more information like x-axis value (causing fault-codes), y-axis value (caused fault-codes), z-value (the percentage of probability), the number of error counts. Legend for the heatmap is displayed on the right side of the diagram that shows the scale of color coding.

### 5.4.3 Analysis with Heatmap for Second Order Transition Matrix

Heatmap for second order transition matrix can be created in the same way described above. X-axis, in this case, represents two consecutive fault-codes, the earlier one occurring at timestamp  $t$  and the second one occurring at timestamp  $t + 1$ . Y-axis represents the fault-code, which followed them i.e., at timestamp  $t + 2$ . The tooltip and legends provide with similar information as described above. Figure 5.9 (b) shows a heatmap for second order transition matrix.

## 5.5 In-depth Analysis Visualizations

This section describes the generation, implementation, and exploration of the in-depth analysis visualizations. The tree visualization is implemented to show the transitions of the and their correlations. Besides, to show the duration and occurrence delay distribution, two bar charts are implemented. This section discusses these visualizations in details.

### 5.5.1 Generation of Fault-Code Tree

Based on the concept discussed in Subsection 4.3.3, tree visualization, named fault-code tree, is implemented with the help of back-end *Production Line Elements Model* as described in Subsection 5.1.3. There are two tree visualizations implemented in the application. One of them is for a single production line element i.e., electronic circuit, station or fault-code, while with the other can be created for multiple electronic circuits. D3 collapsible tree code by Mike Bostock<sup>2</sup> is adapted and customized according to the needs of the thesis to generate this visualization.

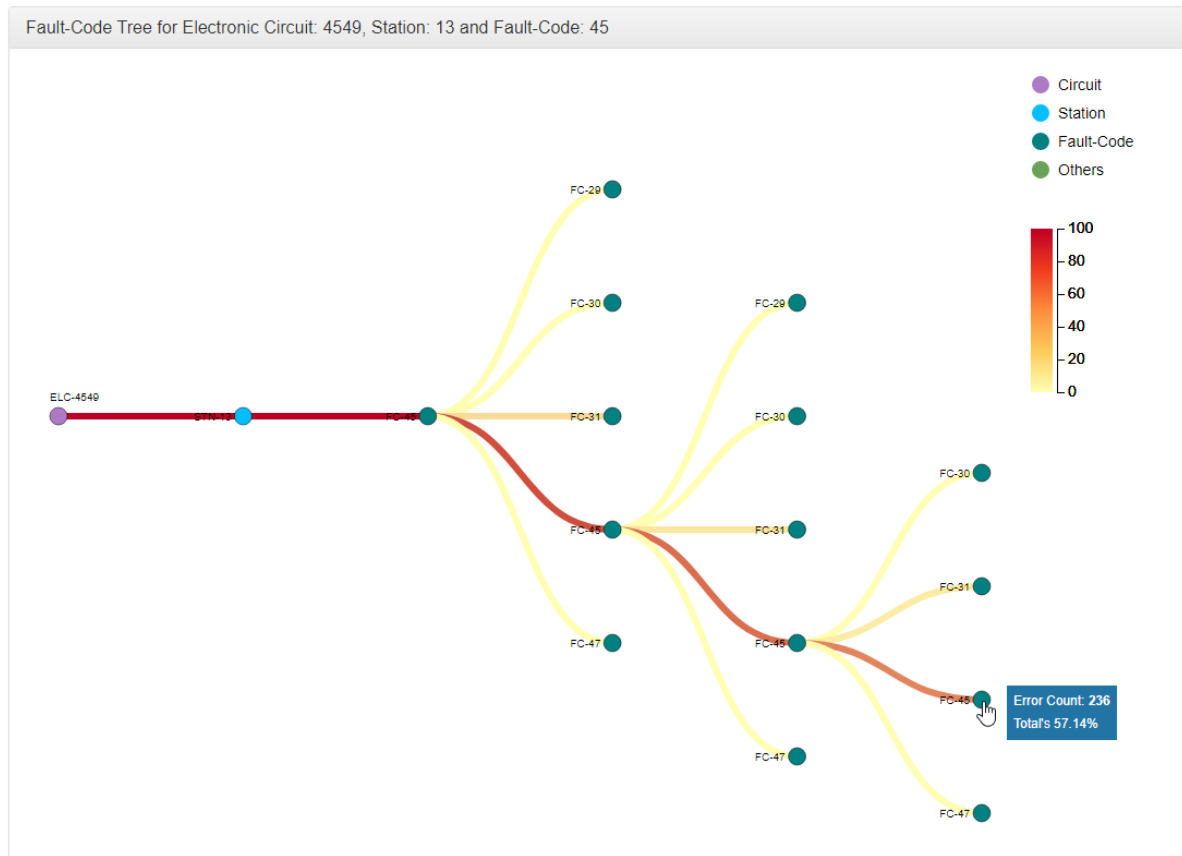
### 5.5.2 Analysis with Fault-Code Tree

*Fault-Code Tree* panel in the correlation analyzer displays the fault-code tree visualization that can be generated from the "Show FC Tree" option of the context menu of the treemap. Upon initial creation, this tree automatically expands up to first children element, which is an FC. Continuous click on any node expands the tree until the depth from the parent FC node is 3. The children of any parent node correspond to the number of fault-code caused by the parent node. The edges of this tree visualization are color-coded based on the contribution of the associated transition of fault-codes. The color of the nodes corresponds to the type of production line. The tooltip provides with this information in the text. New analysis can be initiated from the context-menu of any FC node of the fault-code tree, which replaces the current tree and generates a new tree for the selected elements in the same panel. Legends both for the production line elements and color-scale are placed at the top-right corner of the *Fault-Code Tree* panel. To distinguish the different elements of production line, qualitative color schemes as described in Subsection 2.5.6 are implemented.

---

<sup>2</sup><https://bl.ocks.org/mbostock/4339083>

## 5 Implementation

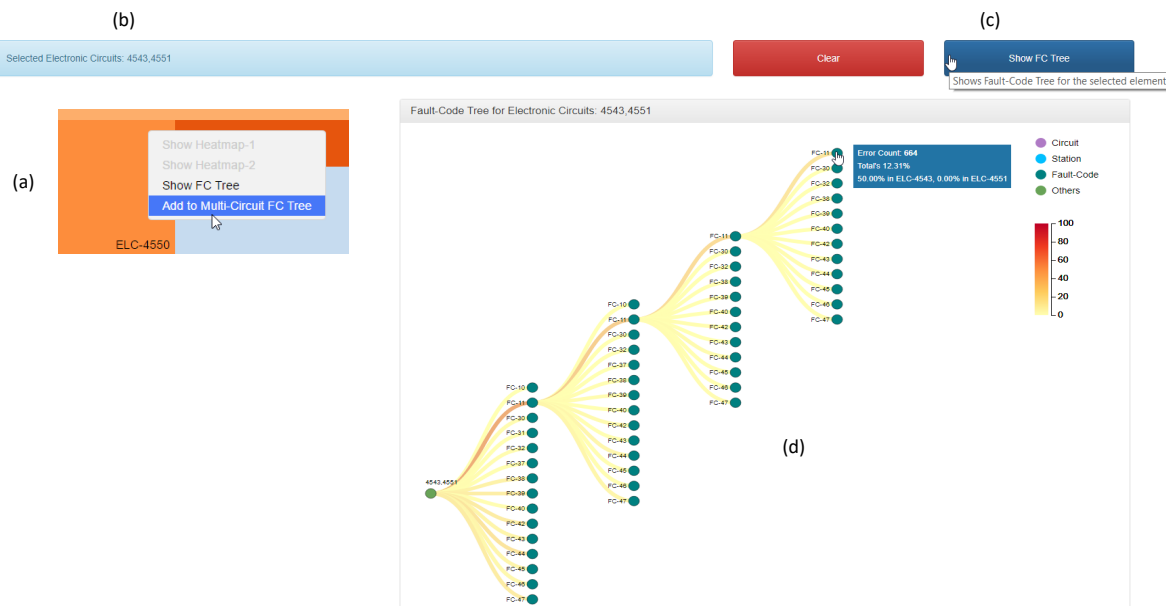


**Figure 5.10:** The screenshot of the implemented fault-code tree. Additional analysis process can be started from the context-menu of the nodes of the tree. Corresponding color and element legend have also been displayed on the top-right corner of the fault-code tree.

### 5.5.3 Analysis with Multi-Circuit Fault-Code Tree

To analyze errors with the multi-circuit fault-code tree, at first, one is to select the electronic circuits one by one by clicking on "Add to Multi-Circuit FC Tree" option from the context-menu of treemap at depth 0. The selected electronic circuit IDs are displayed in a text-form which is displayed on top of the treemap. There are two buttons on the right side of the text-form, "Show FC Tree" button displays fault-code tree for the selected electronic circuits in the *Multi-Circuit Fault-Code Tree* panel of the *Correlation Analyzer* and "Clear" button resets this process. Figure 5.11 shows the different steps to display this visualization.





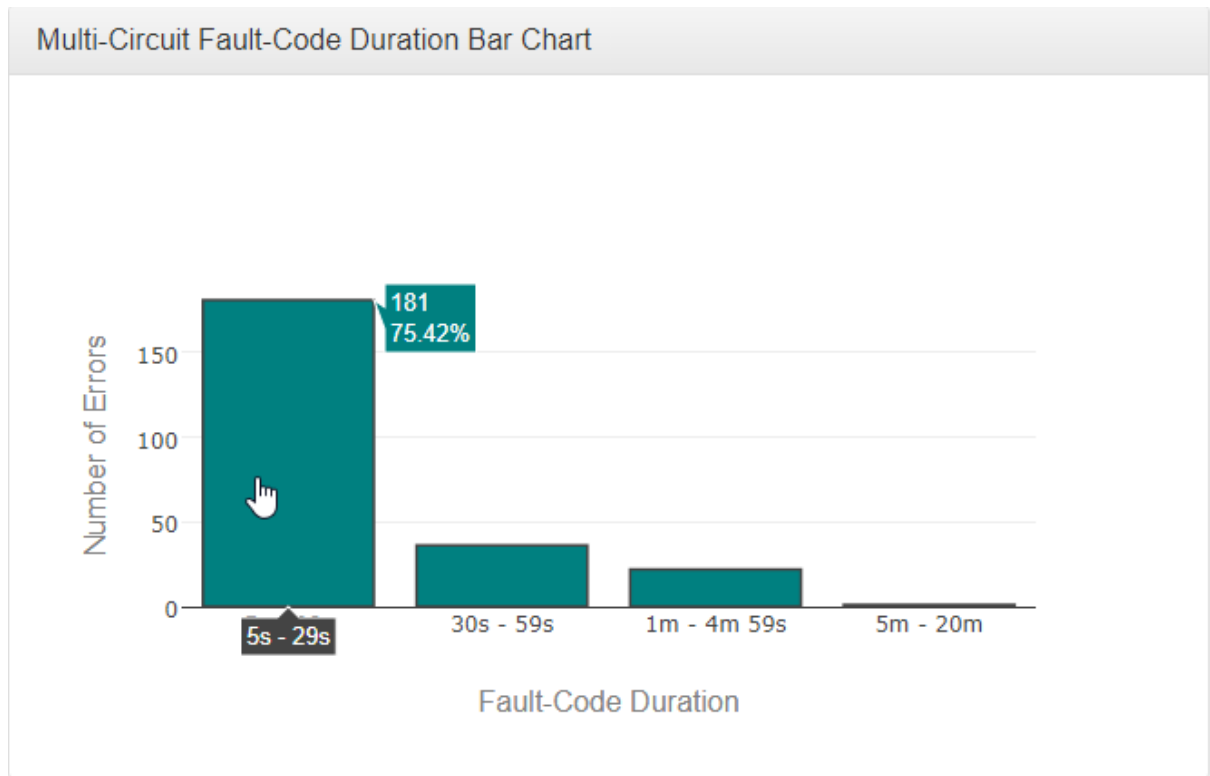
**Figure 5.11:** The screenshot of the multi-circuit fault-code tree (a), which can be launched from the context-menu of the treemap (b), (c), (d). Like the fault-code described above, additional analysis process can also be started from the context-menu of the nodes of the tree. Corresponding color and element legend have also been displayed on the top-right corner of the tree.

#### 5.5.4 Generation of Bar Charts from Fault-code Tree

Based on the concept discussed in Subsection 4.3.3, two bar charts are implemented with *Fault-Code Model* as described in Subsection 5.1.3. One of them helps analyze the duration distribution of any fault-code in the tree, while the other helps analyze the delay distribution between the occurrences of root-causing fault-code and affected fault-code i.e., how and when a specific error could cause another. Both the visualizations can be initiated from the context-menu of FC nodes of any fault-code tree and are displayed on the right-side panels of *Fault-Code Tree* panel.

#### 5.5.5 Analysis with Fault-Code Duration Bar Chart

This visualization depends on the fault-code tree and can be retrieved from any fault-code node of the tree. The x-axis of this bar chart carries the static intervals, while the height of each bar indicates the number of fault-codes occurred in that interval. Based on the suggestions from the domain experts, fault-codes with durations less

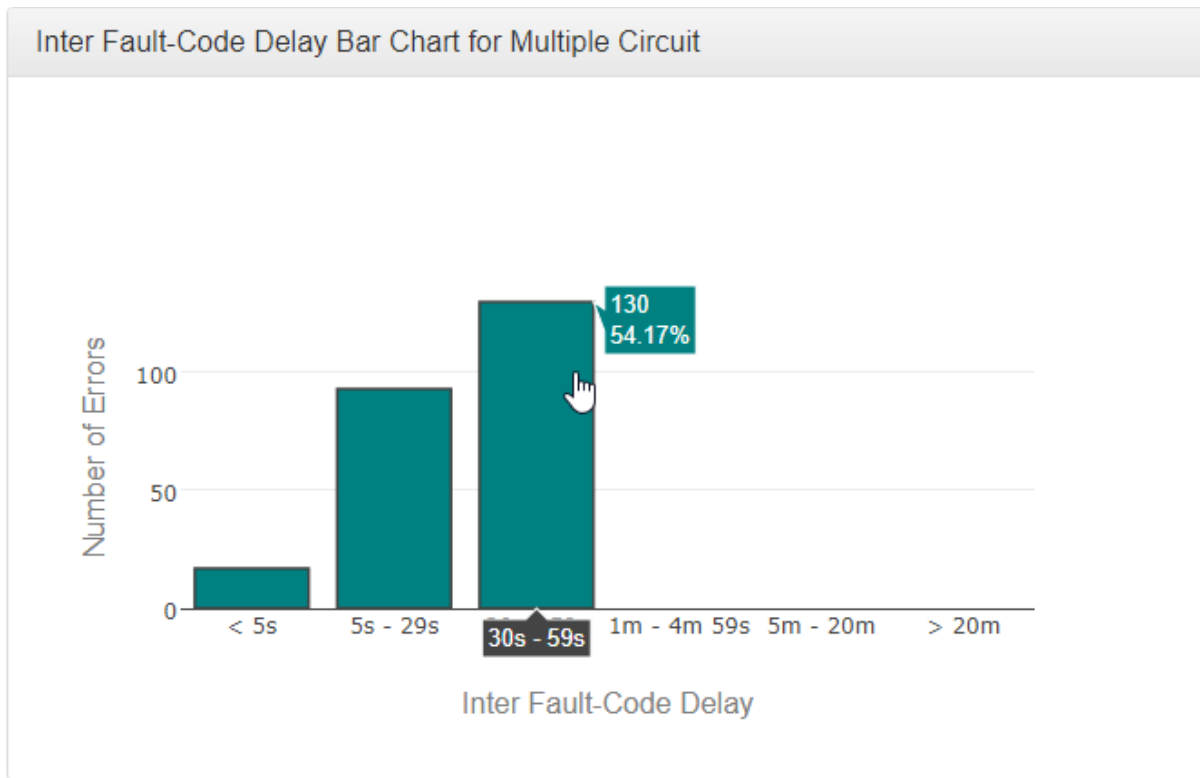


**Figure 5.12:** The screenshot of the implemented fault-code duration bar chart. The tooltip shows the number of errors for the respective interval.

than 5 second and more than 20 minutes are discarded from the calculation and also from visualization. It is observed that the durations of most fault-codes are less than a minute. Therefore, *quantile classification* is adapted for the intervals as described in Subsection 4.3.3. Figure 5.12 is a screenshot of this visualization.

### 5.5.6 Analysis with Inter Fault-Code Delay Bar Chart

Another bar chart named *Inter Fault-Code Delay Bar Chart* is implemented that can be created from any children fault-code node whose parent is also a fault-code tree. This visualization shows the distribution of the timestamp difference between the correlated errors. The x-axis in this visualization displays the delay distribution between the children and root FC, while the y-axis shows the amount of error in that particular interval of time. Figure 5.13 is a screenshot of this visualization.



**Figure 5.13:** The screenshot of the implemented fault-code delay bar chart. The tooltip shows the number of errors for the respective interval.



# 6 Evaluation and Discussion

This chapter outlines the evaluation of this thesis in details, which include the general information about the study setup, collecting expert feedback about the visualizations and analysis process. This chapter also presents a use case scenario along with the results from the evaluations task and general discussion of the evaluation.

## 6.1 General Information

Three expert visualization researchers (1 female, 2 male) aged between 30 to 32 years with 5 to 6 years of experience in the field of information visualization and visual analytics participated in the evaluation. The laptop computer used for the evaluation process had Windows 10 operating system installed. The laptop was built with Intel Core i5 (4th Generation) with two 1.7 GHz processors, 8 GB of DDR3 RAM, 1920 x 1080 pixel of screen resolution and 2048 MB of onboard graphics card. All the analysts used the same wireless mouse to work with the prototype. There were three sequential steps in the evaluation:

1. Reading the user manual
2. Performing a list of tasks along with a use-case and
3. Answering the questionnaire

The steps are discussed briefly in the following Subsections 6.1.1, 6.1.2 and 6.1.3 respectively.

### 6.1.1 Reading the User Manual

At first, the participants were supplied with a color-printed user manual to read so that they could have some preliminary idea about the thesis work and how they could use the prototype. The different interactions and analysis procedures with the prototype were described in the how-to with relevant screenshots. There was also a brief explanation about all the visualizations and how they can be used for error analysis in the production

line. While reading the how-to, they were also asked to use the prototype and explore the functionalities.

### 6.1.2 Performing a List of Tasks

In the second step, the participants were asked to perform a set of tasks consisting of more interactions with some use-case scenarios. They were also asked to observe the data overview visualizations like ELC block diagram and treemap as well as to analyze some fault-codes with the correlation analysis visualizations (heatmaps and fault-code trees) and whether they find any interesting correlations by using them. They were given some use-case pattern of fault-codes, which they were asked to analyze with both the fault-code trees and both the bar charts.

### 6.1.3 Answering the Questionnaire

At last, the participants were asked to answer an online questionnaire. The questionnaire consists of filling up basic information about the participant, answering individual visualization specific questions, rating the visualizations based on the usefulness to analyze the errors, writing down or mentioning important suggestions or comments etc.

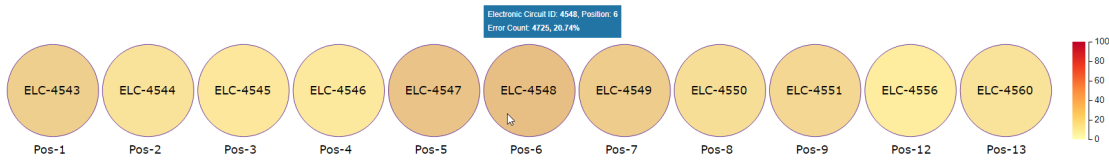
## 6.2 Evaluation for Data Overview Visualizations

The evaluation starts with the exploration of the data overview visualizations (electronic circuit block diagram, electronic circuit bar chart, and treemap). The participants were presented with the home page of the prototype and asked to interact with them to get the idea about the positions of electronic circuits in the production line and how they are affected by the errors.

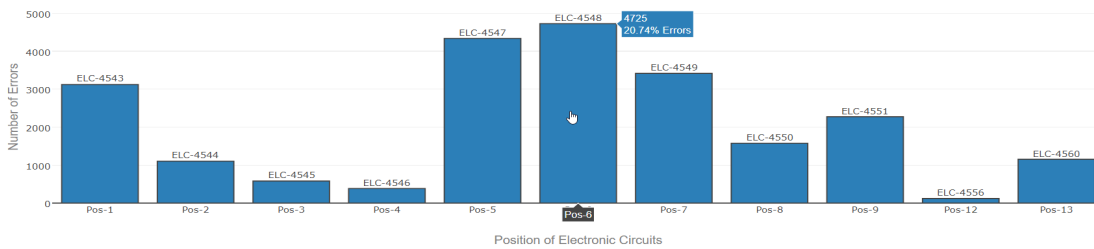
### 6.2.1 ELC Block Diagram

The analysts were asked to switch between ELC block diagram with circles and ELC bar chart and then to analyze their positions in the production line as well as the corresponding error contributions. They were also asked to find out the ELC that is most affected by errors and its position as shown in Figures 6.1 and 6.2. All three of the participants found the ELC bar chart more useful than the one with the circles.

## 6.3 Evaluation for Correlation Analysis Visualizations



**Figure 6.1:** Evaluation of the ELC Block Diagram with Circles. The participants of the evaluation found it difficult to distinguish the circles based on the color-coding.



**Figure 6.2:** Evaluation of the ELC Bar Chart. The participants of the evaluation found this chart more useful than ELC block diagram with circles.

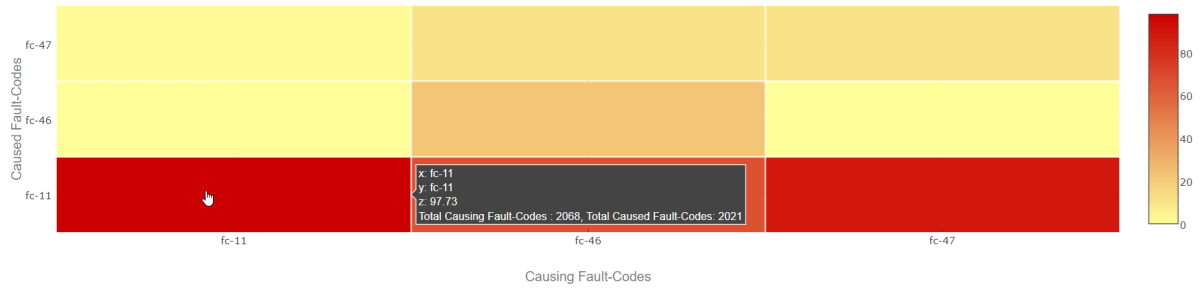
According to them, the ELC bar chart was easier to distinguish the differences in the values, while it was difficult to compare them based on the color-coding in the ELC block diagram with circles. There were suggestions to vary the size of the circles to show their actual number and improve the color coding.

### 6.2.2 Treemap

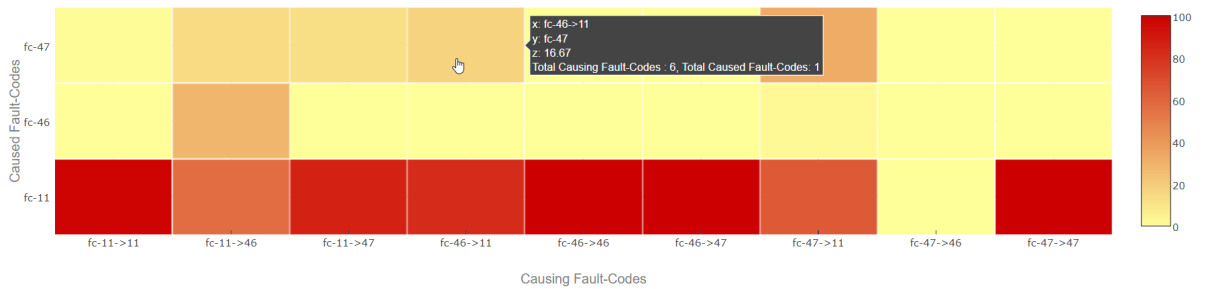
The participants were asked to analyze and filter different elements of the production line with the treemap. There were some suggestions to improve it. They mentioned that initially, they had needed some time to get used to the hierarchy levels of the treemap, hence asked for some means of orientation or guidance to display the hierarchy levels in a better way. There were also suggestions to improve the color coding and sizing of the rectangles in the treemap.

## 6.3 Evaluation for Correlation Analysis Visualizations

This section discusses the evaluation of the heatmaps both for first and second order transition matrix as well as both the fault-code trees along with the two bar charts.



**Figure 6.3:** The screenshot of the analysis of FC-11 in STN-2 under ELC-4543. The participants were asked to find the correlation between two FC-11s from this visualization.



**Figure 6.4:** The screenshot of the analysis of FC-47 follows FC-47 and FC-47 follows FC-11 and in STN-2 under ELC-4543.

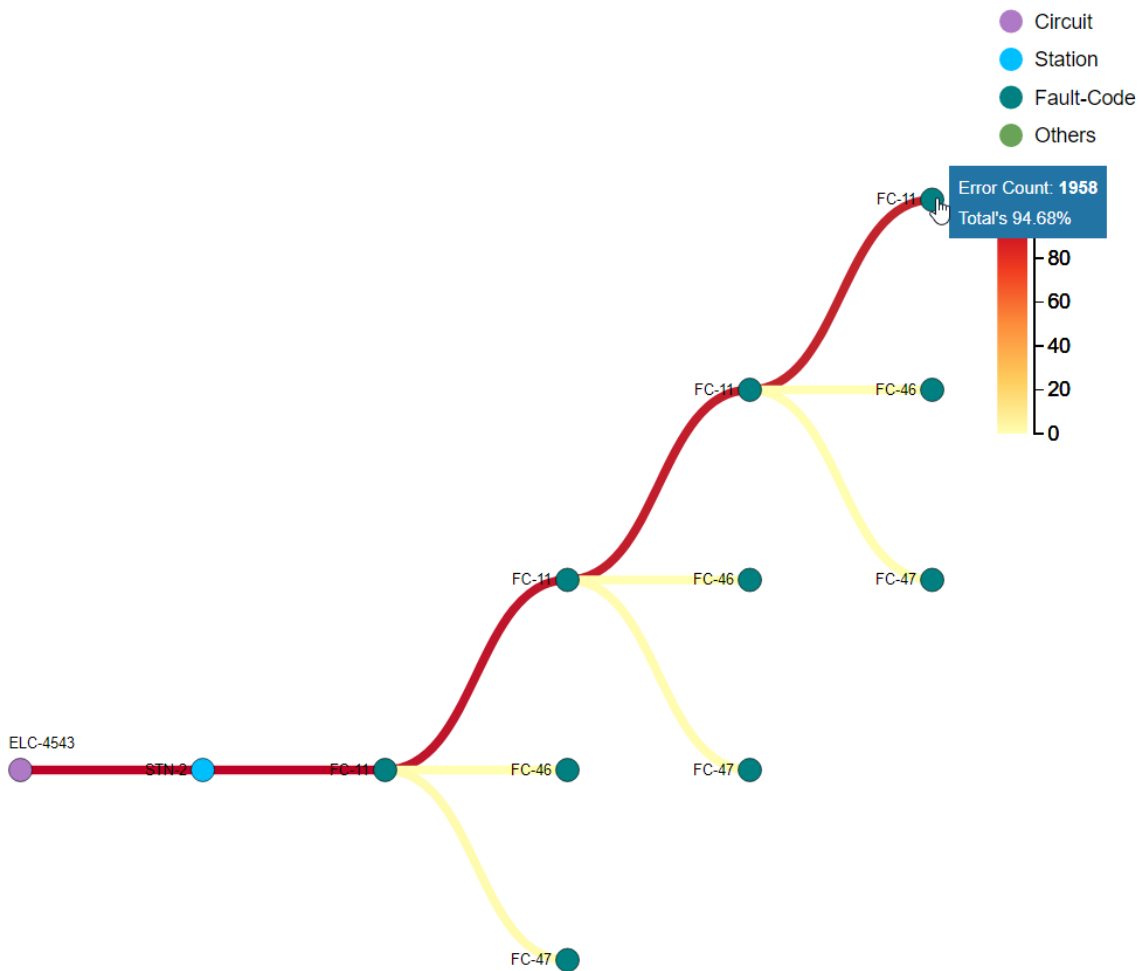
### 6.3.1 Heatmaps

The participants were asked to generate the heatmaps both for the first and second order transition matrix for STN-2 under ELC-4543 and analyze them. They were asked to find the correlation between the fault-codes from the generated heatmaps. Figure 6.3 shows this analysis process.

On the other hand, from the heatmap for the second order for the same STN and ELC, they were asked to find the correlation in the case where FC-46 caused FC-11 and then FC-11 caused FC-47. Figure 6.4 shows this analysis process.

The participants mostly found the heatmaps straightforward. According to them, once one knows which axis represents what, finding the respective correlation becomes easier. On top of color coding of the heatmap, they also found the tooltip useful. There was a suggestion to clarify the  $z$  value in the tooltip, as there was no indication that it meant the co-occurrence probability.





**Figure 6.5:** The screenshot of the analysis of a fault-code FC-11 in STN-2 and ELC-4543 from FC Tree. The participants of the evaluation followed the darkest path and found the most interesting correlation for this case.

### 6.3.2 Fault-Code Tree

The analysts were asked to generate an FC tree for FC-11 from STN-2 of ELC-4543 and analyze the fault-codes. They also mentioned that the tooltip helped them analyze the errors easily. Figure 6.5 depicts this analysis.

The visualization experts found the fault-code tree useful to find out the cause-effect correlations among the errors. There were some suggestions to improve it, which include using a consistent color coding between the tree visualization and treemap, improving the visual clutter problem and adding a filter that automatically shows the most interesting pattern or path (error transitions). There were also suggestions to add

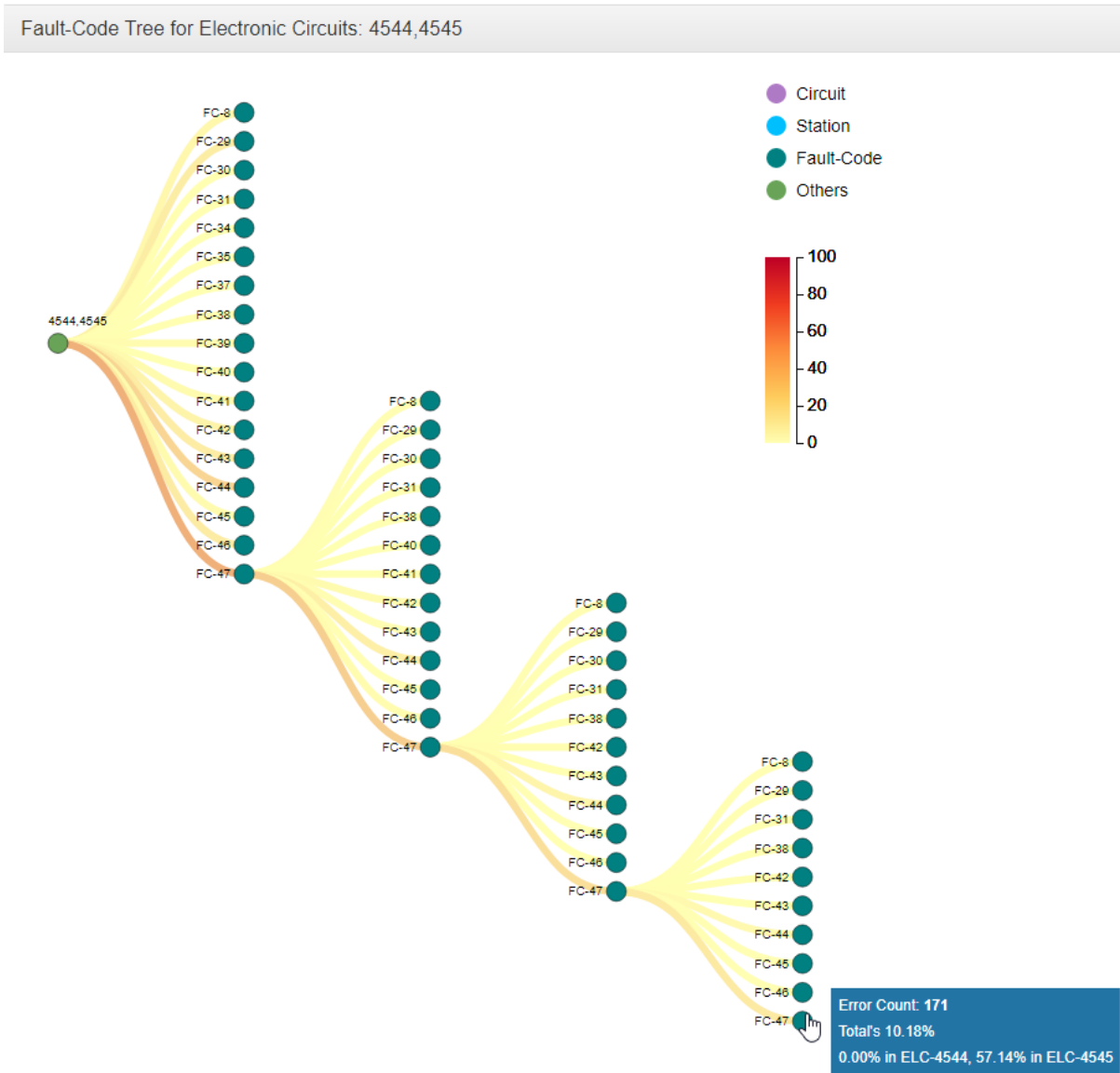
the functionality of expanding the whole tree in the case when the number of children FC is presentable within the defined size of the FC tree panel.

### 6.3.3 Multi-Circuit Fault-Code Tree

The analysts were asked to generate this visualization for ELC-4544 and ELC-4545. They were then asked to find out the most interesting correlation from the generated tree. Figure 6.6 exhibits this analysis process.

The analysts initially found the generation of multi-circuit fault-code tree a little bit difficult. Once they were able to generate once, it became easier for them. There was a suggestion to change the 'others' legend to 'multi-circuit' or something like that as 'others' in this case always represents multiple electronic circuits. There was also a suggestion to add some message to let the users know exactly where they were in the process of generating multi-circuit FC tree. Because it took some time for the multi-circuit FC tree to get displayed, they suggested adding some progress bar during generating the tree. This feature was implemented after the evaluation not only for this, but also for all the visualizations.

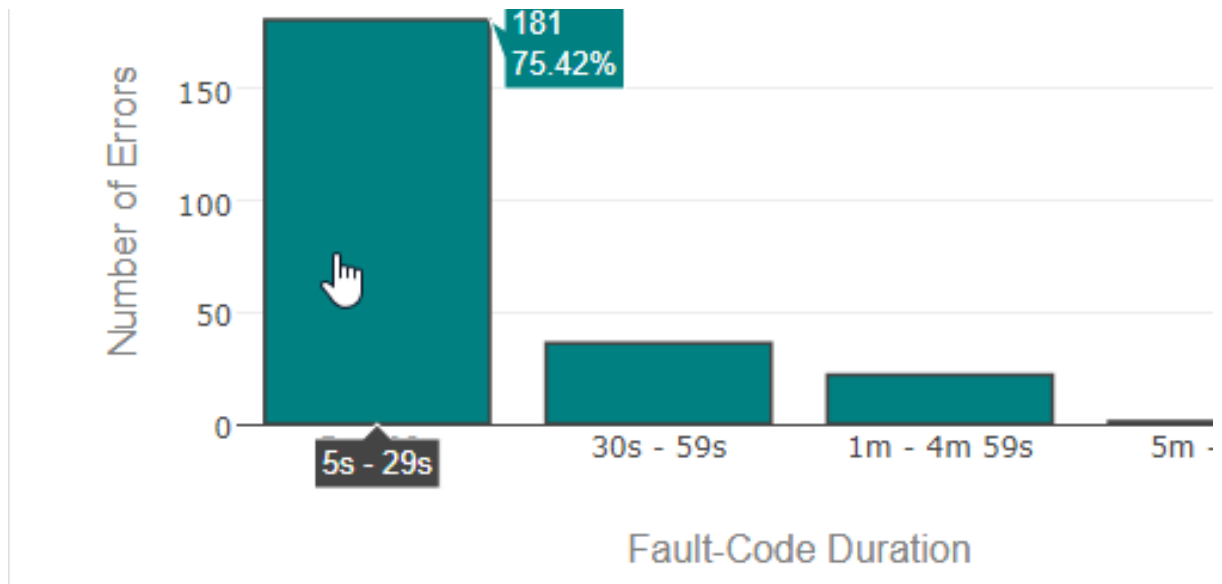
### 6.3 Evaluation for Correlation Analysis Visualizations



**Figure 6.6:** The screenshot of the analysis of a fault-code FC-47 with pattern FC-47→FC-47→FC-47 with multi-circuit FC Tree. The participants were asked to analyze this pattern from this multi-circuit FC tree.

### 6.3.4 Fault-Code Duration Bar Chart

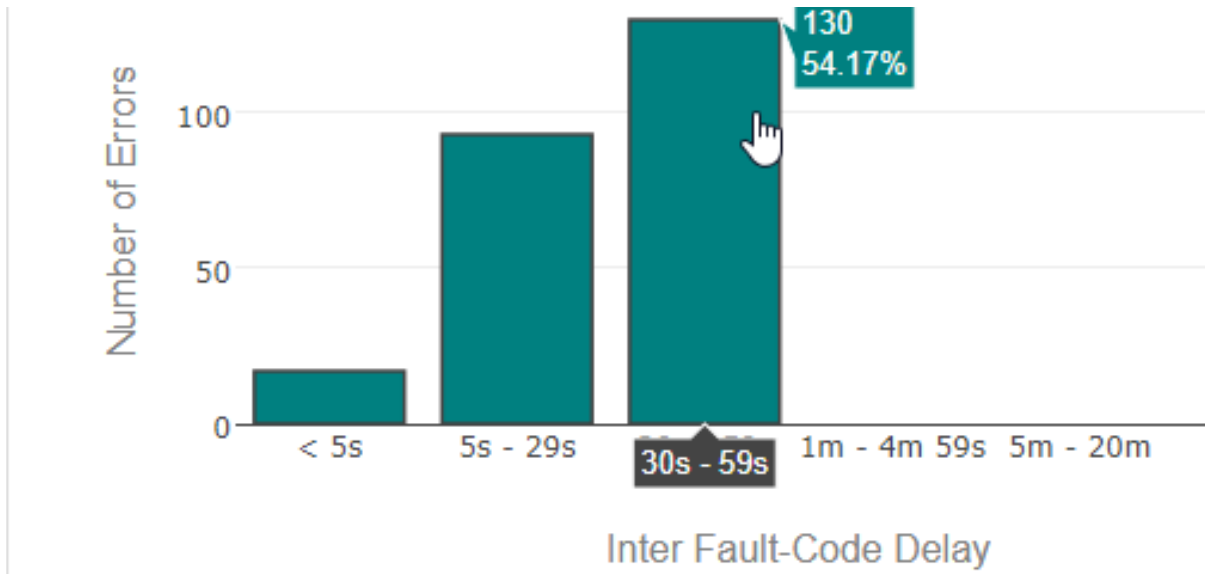
The analysts were asked to analyze the distribution of the durations and find out the most interesting one from the fault-code pattern FC-47→FC-47→FC-47→FC-47 from the multi-circuit fault-code tree for ELC-4544 and ELC-4545 as shown in Figure 6.7.



**Figure 6.7:** Screenshot of the analysis of a fault-code FC-47 with pattern FC-47→FC-47→FC-47→FC-47 with FC bar chart generated from multi-circuit FC tree as shown in Figure 6.6.

### 6.3.5 Inter Fault-Code Delay Bar Chart

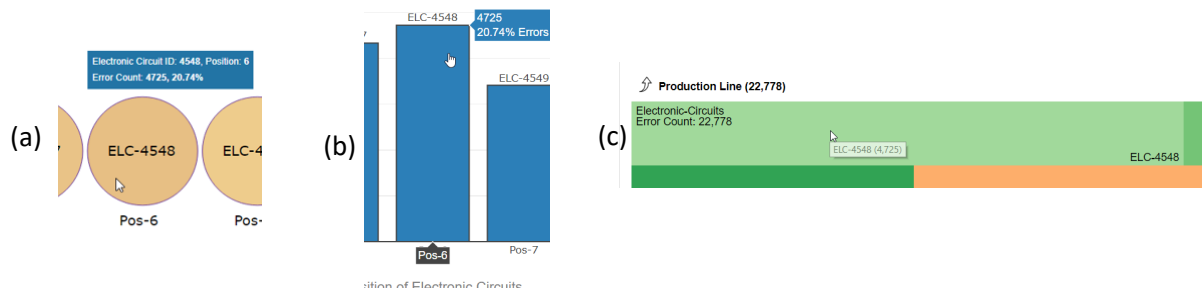
Inter fault-code delay bar chart helps analyze the time delay distribution between the last and first fault-code of any fault-code pattern. The analysts were asked to create an inter fault-code delay bar chart for the same fault-code pattern above and find out the most interesting observation from the created visualization as shown in Figure 6.8. According to the participants, as long as someone has the idea about the data and production line domain, both the fault-code duration and inter fault-code delay bar chart might help error analysis process.



**Figure 6.8:** The screenshot of the analysis of a fault-code FC-47 with pattern FC-47→FC-47→FC-47→FC-47 with inter FC delay bar chart generated from multi-circuit FC tree as shown in Figure 6.6.

### 6.3.6 Use Case Scenario

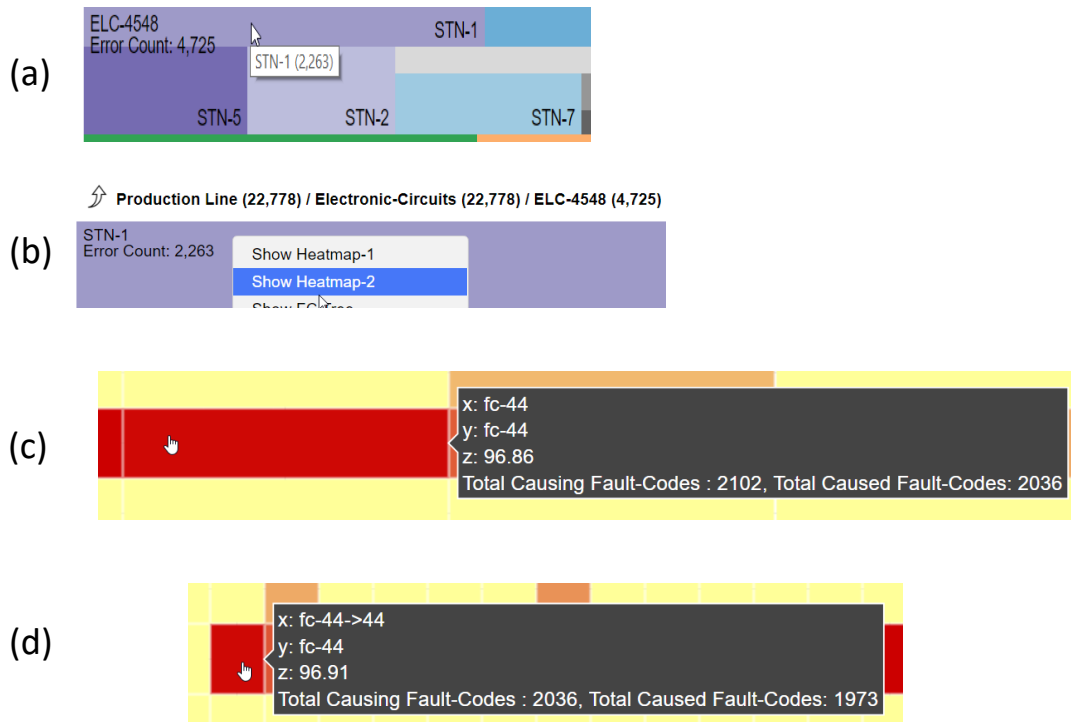
With the implemented prototype, if an analyst initiates an analysis to find out the electronic circuit that is most affected by the errors, it is possible to do that either from the darkest circle or the tallest bar or the biggest rectangle in the ELC block diagram with circles, ELC bar chart, and treemap respectively, which results ELC-4548. The position of ELC-4548 is possible to know from the first two visualizations mentioned above. Figure 6.9 shows this analysis. Now, for the detailed analysis of that ELC, treemap can be used. Now, to conduct pairwise error analysis for this ELC, a STN or FC in ELC-4548 is to be selected from the treemap.



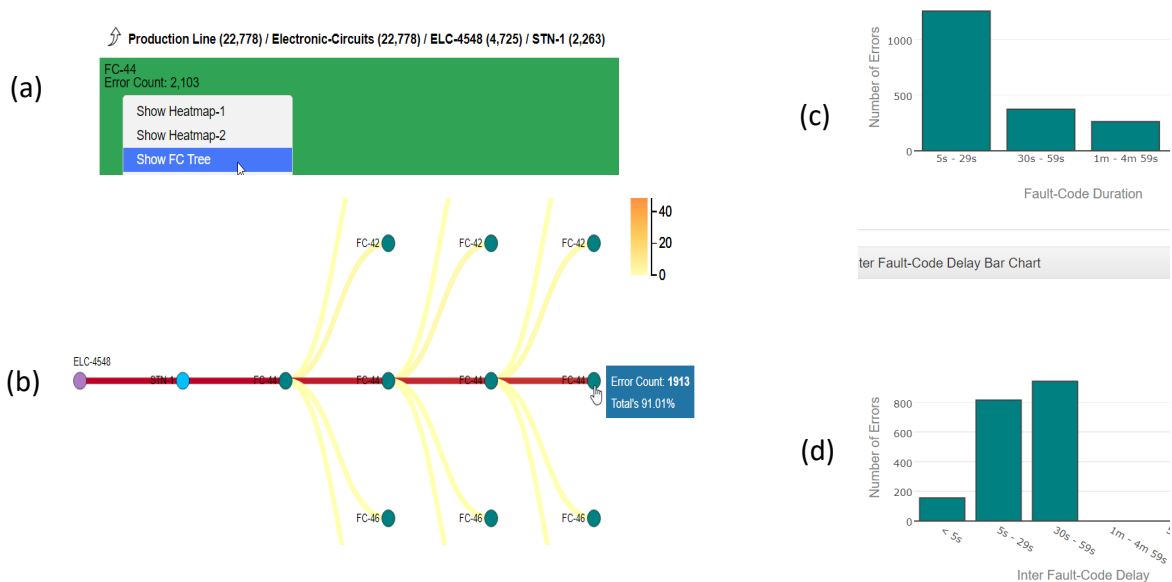
**Figure 6.9:** The screenshot of an analysis of ELC-4548 from ELC block diagram with circle (a), ELC bar chart (b), and treemap (c). The color-code, height and size are the respective variables that help identify how much an ELC is affected by errors. The first two visualizations also show their positions in the production line.

Figure 6.10 shows the analysis of the most affected STN (STN-1) in ELC-4548, and the generated heatmaps from this STN. This analysis also shows that the co-occurrence between two FC-44s is the highest for first and second order, which is almost 97% for both the orders. Now, for higher order correlations analysis, an analyst can initiate the FC tree for this FC-44 of STN-1 under ELC-4548. From the displayed tree, it can be observed that for third order, the co-occurrence reduces to 91%, which is shown in Figure 6.11(b). The two bar charts as shown in Figure 6.11 (c and d), help find the duration and delay distribution respectively.

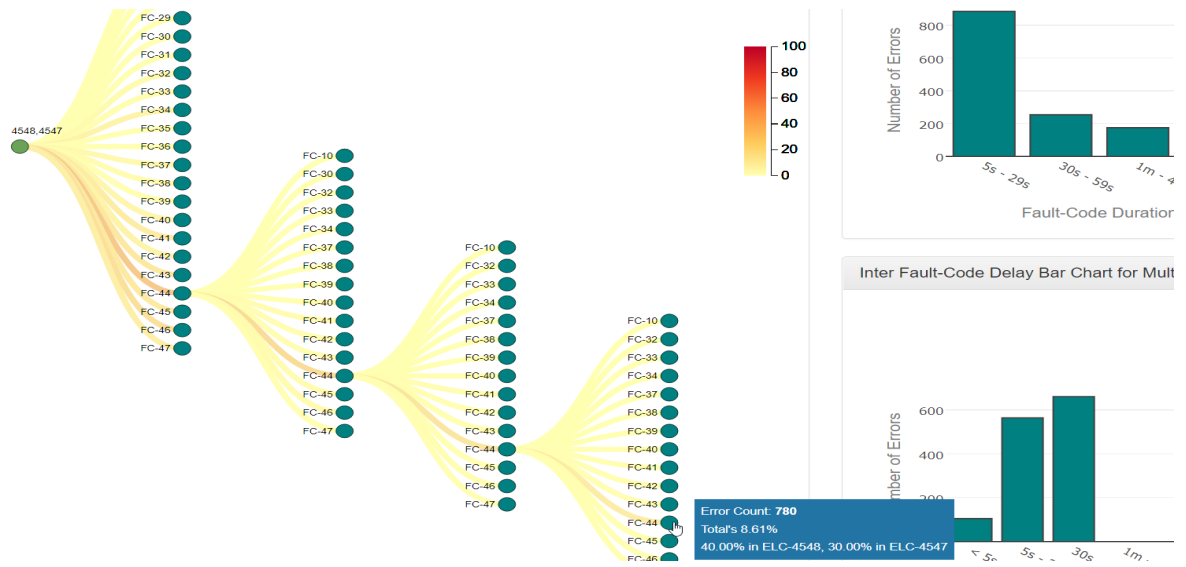
## 6.3 Evaluation for Correlation Analysis Visualizations



**Figure 6.10:** The screenshot of an analysis of STN-1 from ELC-4548 (a, b) and the generated heatmaps. The darkest cells (c, d) in the heatmap show the most co-occurrence of fault-code



**Figure 6.11:** The screenshots of an analysis of FC-44 with FC tree generated from STN-1 of ELC-4548 (a, b). Parts (b, c) show corresponding bar charts.



**Figure 6.12:** The screenshot of the analysis of FC-44 generated from ELC-4548 and ELC-4547, corresponding bar charts.

Now, if an analyst analyzes a fault-code for two most affected ELCs, which are ELC-4548 and ELC-4547 in this case, it is possible to launch a multi-circuit FC tree from the treemap as shown in Figure 6.12. This analysis shows that the co-occurrence between two FC-44s reduces to less than 10% in third order if the tree is created for the above mentioned two ELCs.



## 6.4 Result and Discussion

This section summarizes the results of the tasks that were given to the analysts and general discussion about them.

### 6.4.1 Discussion for Data Overview Visualizations

Although it was difficult to distinguish the ELCs by observing the color-difference in the ELC block diagram with circles, this information was clear in the ELC bar chart as shown in Figure 6.2.

There were some feedback from the participants to improve the data overview visualizations. According to them, an analyst might want to analyze the most occurring or causing fault-codes in the whole production line. Along with color-coding, the size of the circles could be varied accordingly to visualize the error contributions. The color-coding of electronic circuit block diagram could be made consistent with the color of treemap rectangles. The colors of treemap rectangles could be improved so that the adjacent rectangles do not have similar colors. The hierarchy of the elements of the production line could be visualized more efficiently by initiating some small tree visualization, by which an analyst could navigate from any level to any level.

Although there is room for improvement, the analysts were able to filter and navigate through the different production line elements by using the implemented treemap. With the context-menu in the different hierarchy levels, they were able to generate the other visualizations for the correlation analysis.

### 6.4.2 Discussion for Pairwise Analysis Visualizations

The participants of the evaluation were able to find out the correlation between two FC-11s from the heatmap for first order transition matrix. After observing this matrix diagram, they were able to find out that there were total 2068 instances of causing FC-11 in the STN-2 under ELC-4543 and the following (caused) fault-code were also FC-11 in 2021 instances. This leads to the result that almost 98% of the time FC-11 causes itself, which makes it an interesting correlation. This analysis is shown in Figure 6.3.

From the heatmap for the second order transition matrix for the same STN and ELC as above, the participants were able to find out the correlation in the case where FC-46 caused FC-11 and then FC-11 caused FC-47 as shown in Figure 6.4.

### 6.4.3 Discussion for In-depth Analysis Visualizations

From the fault-code tree for STN-2 under ELC-4543, the analysts were able to find out that for this case FC-11 almost in 95% case follows itself until depth 3. The color-code of edges of led them to the most interesting correlation. They also observed that although there were other three types of fault-codes, which are caused by FC-11, but those were not that interesting for this particular case.

Similarly, it was also possible to find out the most interesting pattern from the multi-circuit FC tree generated for ELC-4544 and ELC-4545, which is FC-47→FC-47→FC-47→FC-47. With the observed pattern, they were also able to realize that most of the time FC-47 just causes itself. Using the tooltip, they were able to find out the exact probability of that fault-code pattern as shown in Figure 6.6.

Figures 6.7 and 6.8 show the results of the use-case scenario regarding the fault-code duration and inter fault-code delay bar charts, where the fault-codes that follow the pattern shown in Figure 6.6 lasted mostly 5 to 29 seconds and had the delay distribution mostly between 30 to 59 seconds.

From the above observations, results, and discussions, it can be said that both the heatmaps, and both the fault-code tree and multi-circuit fault-code tree are intuitive ways to visualize the higher order dependencies and correlations among the fault-codes.

# 7 Summary and Future Work

This chapter summarizes the thesis work. Some future work suggestions are also outlined in this chapter.

## 7.1 Summary

The goal of this thesis was to develop a visual analytics approach which would help analyze the correlations and dependencies of the industrial production line errors. This thesis used Markov model to calculate correlations and higher order dependencies among the increasing number of errors that get logged in an industrial production line every day.

To visualize the actual positions of electronic circuits that are the top most hierarchical elements of the production line, simple block diagram with circles and bar chart were implemented. To analyze any particular element that might be interesting to the analyst for deeper analysis, a Treemap was implemented.

To visualize the higher order dependencies and correlations, both the first and higher order of Markov model were applied to the implemented visualizations. Because of the constraints like space and time required for data-rendering, the heatmaps only for the first and second order transition matrix diagrams were implemented. To visualize the cause-effect correlations among the errors, rooted tree visualization was initiated. This visualization named fault-code tree helps analyze the dependencies among the errors up to third order of Markov chain theory. To ease the analysis process and help visualize the interesting transitions of the fault-codes, color-coding was applied to the edges of the tree visualization. To help visualize the distribution of error durations and the delays between the causing and the caused fault-codes, the bar charts were implemented.

The evaluation also showed that it is possible to find out the correlations among the fault-codes with the implemented prototype, which would help the analysts in the respective domain analyze the errors. But with the implemented prototype, it is not possible for the analysts to get an idea about the actual meaning of a fault-code. So, there were suggestions to somehow visualize what the fault-codes mean in reality.

### 7.2 Future Work

Based on the feedback from the visual analytics experts and after the evaluation, some future works are identified at the end of the thesis work. For the fault-code tree, there could be an option, which automatically expands the most interesting path of transition without user interaction. At the same time, this rooted tree visualization could also be improved by adding a functionality, which will expand all the children nodes. But the problem for this approach might yield another visualization problem like visual clutter. To deal with that challenge, there could be some threshold value of the number of children nodes, under which the tree would expand. The threshold could also be defined by the analysts. The colors of the nodes of the fault-code trees could be made consistent with the colors of treemap rectangles. There could be another option to change the shape and size of the nodes and the edges of the tree, which would vary depending on the type of fault-codes and amount of fault-codes corresponding to the node. Some scaling techniques could be applied to deal with the problems that might occur if the size of the nodes or edges varies based on data. For the bar charts, there could be some option to select the nodes and then visualize the distribution of the selected nodes only.

In general, visualizations could be improved by initiating interaction techniques like *Linking and Brushing*, which would let an analyst choose a subset of the data with an input device like a mouse, and apply some brush effect like highlighting on the selected subsets. Thus, an analyst would be able to focus only on the interesting section of the visualization. To achieve this feature, hybrid visualizations like *nodetrix* [HFM07] could be implemented. To visualize what an actual fault-code mean, some semantic information could be added. Some symbols, pictures etc. could be used to visualize the meaning of the different elements of the production line or the different type of the errors. Instead of opening a new visualization below the currently open ones, multiple linked views that fit in a single screen could be added.

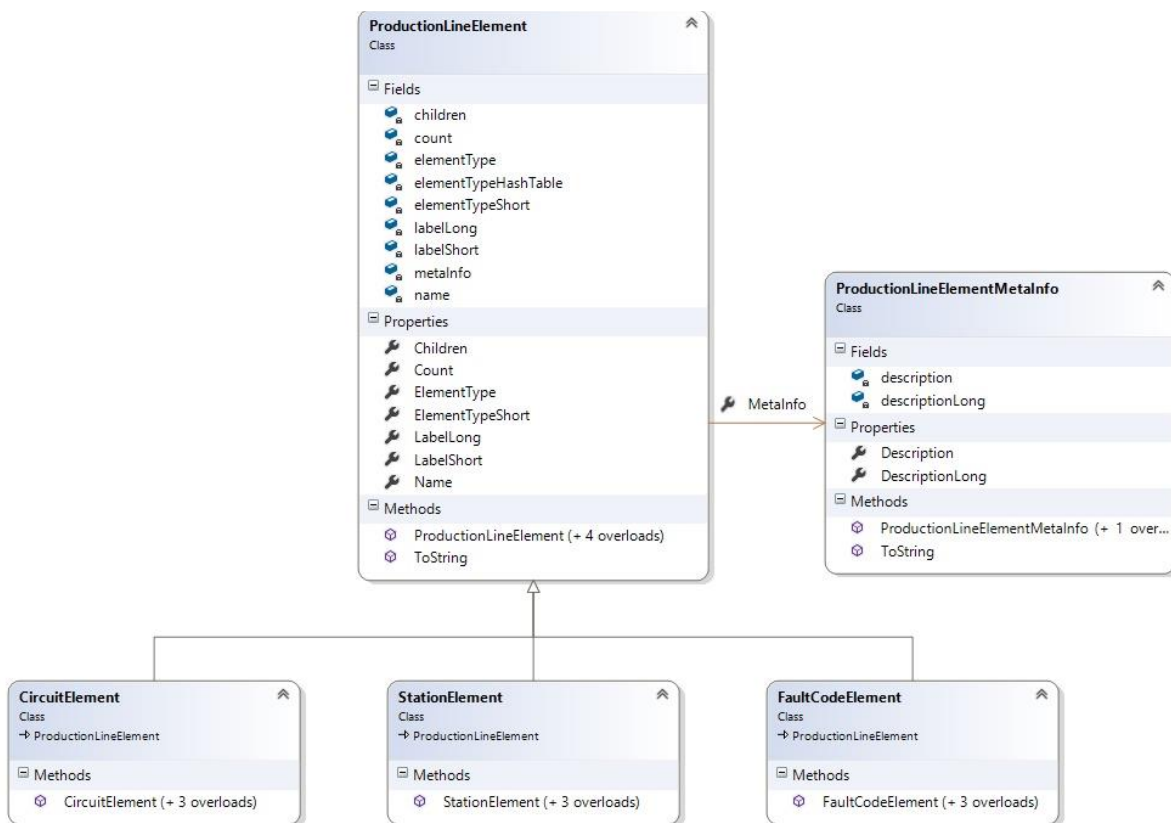
# Appendices



# A Class Diagrams

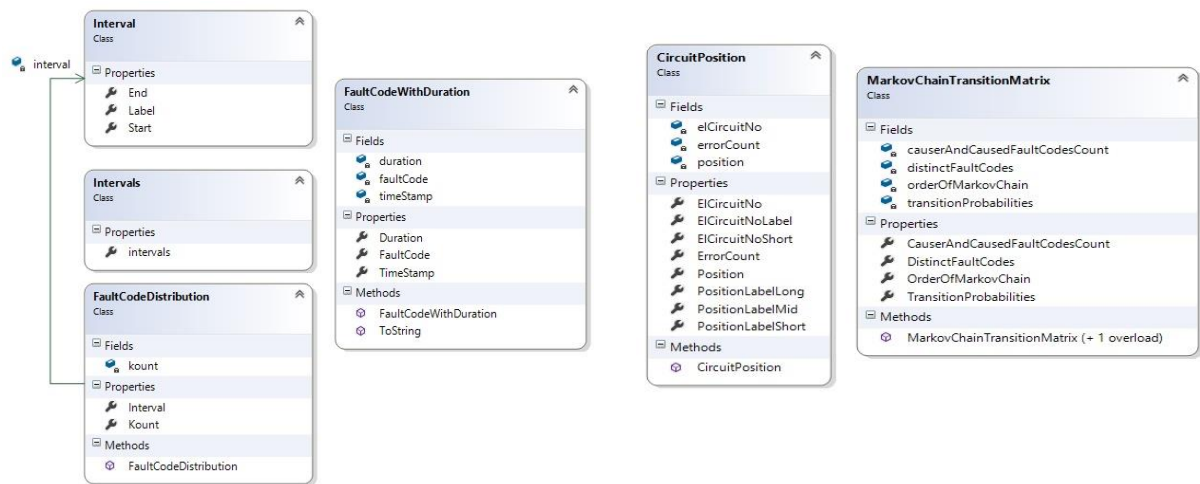
## A.1 Back-end Models

Figure A.1 and A.2 in this section displays the class diagrams for different models described in Subsection 5.1.3.



**Figure A.1:** Class diagram of production line elements.

## A Class Diagrams



**Figure A.2:** Class diagram of fault-code distribution, electronic circuit position and Markov chain transition matrix model.



# B Evaluation Appendices

## B.1 How-to Used for the Evaluation

Appendix B.1 presents the user-guide used for the evaluation.

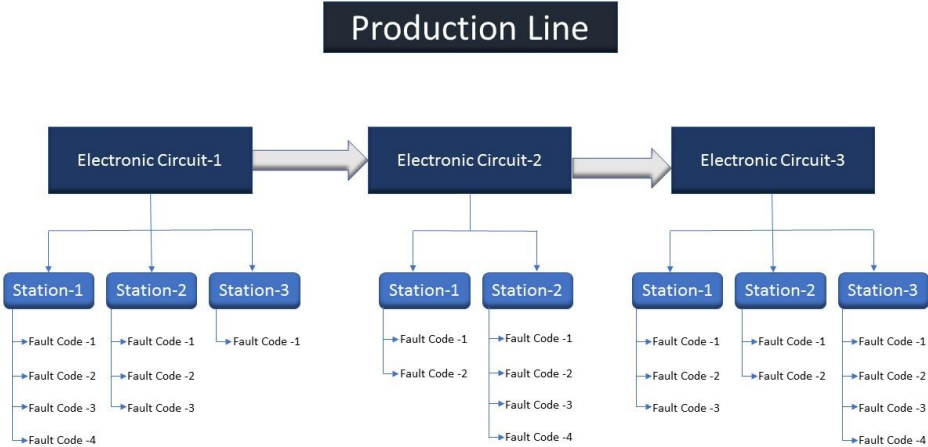
# How to use the Correlation Analyzer Prototype

## Background

With ever growing digitalization and automation in the industrial production domain, huge amount of error reports become available. Thus, the generated error reports become very important to study the errors and find effective solutions that might prevent similar future errors. However, not each error possesses equal importance, as some errors may be the result of previously reported errors. Hence, it is important for the domain experts to understand, how the reported errors are related to each other. Finding a suitable model and implementing a visual analytics approach upon that model may help understand these correlations and thus may reduce the number of errors in significant amount.

The goal of the evaluation is to collect expert feedbacks about the implemented prototype. The experts will be given this how-to at first and then will be requested to perform some tasks related to finding error correlations. Then some more feedbacks will be taken about different visualizations implemented for the prototype and how they can help find the error correlations.

## Data Structure



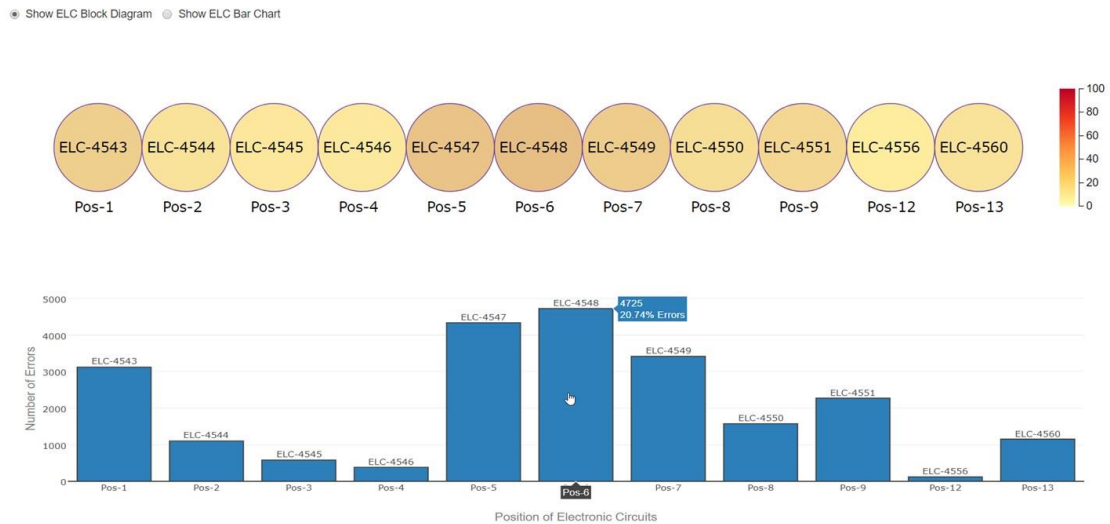
The Figure above shows hierarchical data structure, where *electronic circuits* are the top-most elements of production line and placed in one after another in series. Each *electronic circuit* consists of more *stations* placed parallel to each other. Each error occur in a *station* is identified by *fault-code*.

## Introduction to the Prototype

The first two visualizations shown at the start of the application are electronic circuit block diagram and production line treemap. The other visualizations can be retrieved from treemap context-menu.

# Visualization 1: Electronic Circuit Block Diagram

The goal of this visualization is to show the physical position of electronic circuits in the production line which is shown with two visualizations. One of them is called ELC Block Diagram which is drawn with the circles and the other one is ELC Bar Chart. The default electronic circuit block diagram is the one with circles.

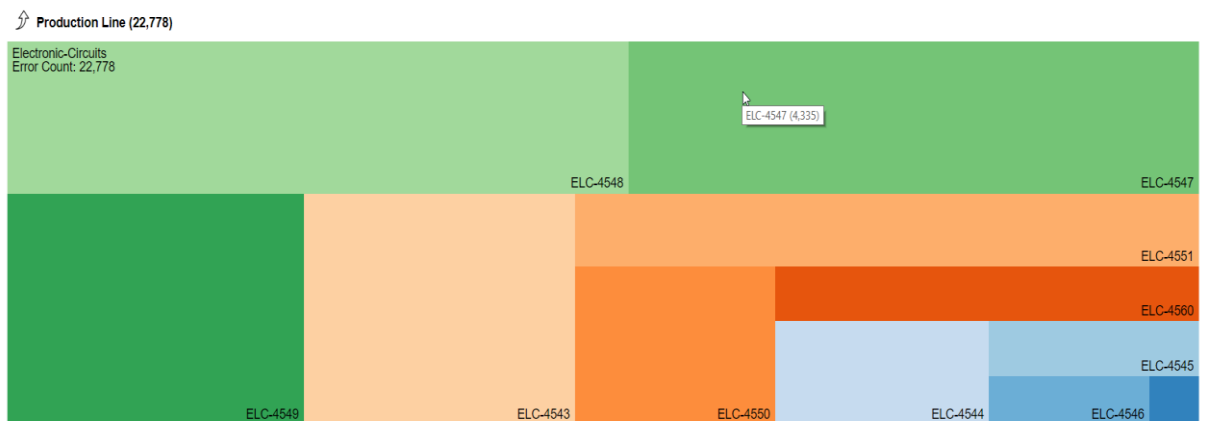


1. Click on the radio buttons (“Show ELC Bar Chart” and “Show ELC Block Diagram” ) to switch between the visualizations.
2. In the bar chart, a tooltip provides with additional information.

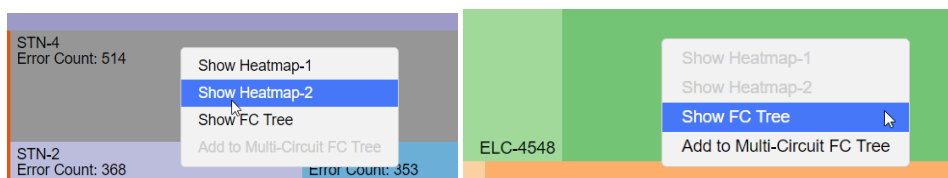
## Visualization 2: Production Line Treemap

A treemap visualization allows browsing through different hierarchy levels of production line and inspect them.

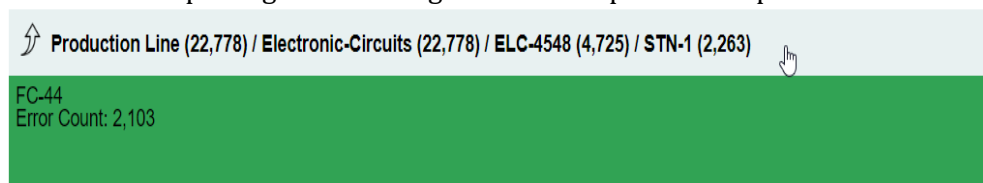
- at depth level-0, the production line itself could be visualized while electronic circuits being its children.
- at depth level-1, the electronic circuits could be parents while associated stations could be shown as its children.
- at depth level-2, the stations could be parents while corresponding fault-codes being its children.



1. Click-on any cell of the visualization to explore different hierarchy levels of production line. The elements can be browsed until fault-code (depth 3).
2. Right click on any element to explore more options. The options are different depending on the element type. Different visualizations can be explored from here.



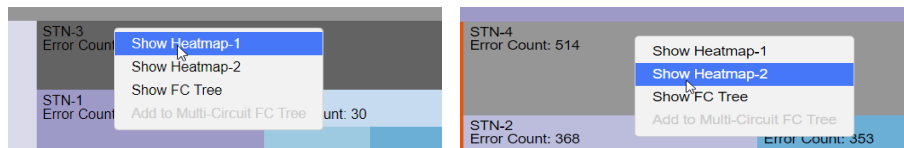
3. Click-on the top navigation bar to go back to the previous depth.



## Visualization 3: Heatmap for 1<sup>st</sup> and 2<sup>nd</sup> Order Transition Matrix

The purpose of this visualization is to show the correlation between the errors with a transition matrix. This visualization is implemented with the theory of first and second order Markov Model. Color encoding is also implemented to show the correlation.

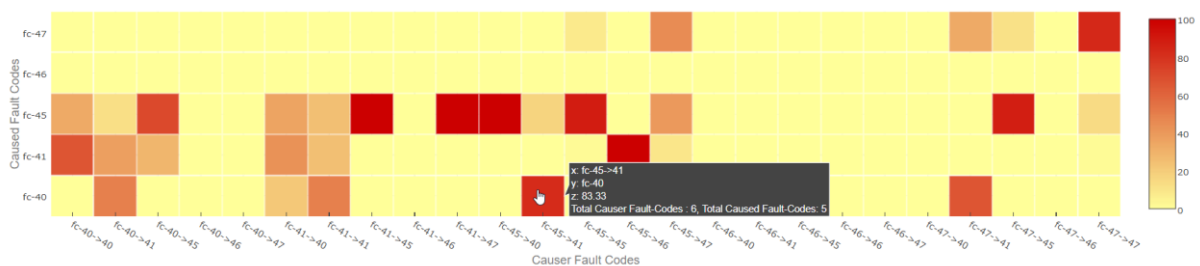
1. Right click on the appropriate elements (inner views of treemap - station or fault-code) and select “Show Heatmap-1” or “Show Heatmap-2” to see this visualization.



2. The values in x-axis are the causing or source errors and the ones in y-axis are the following or effected errors which occurred in the next timestamp.



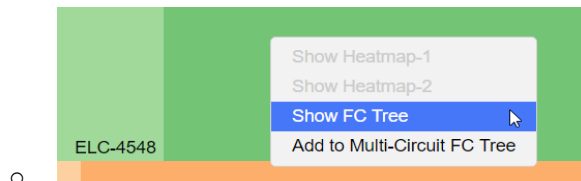
3. Tooltip provides with additional information about the correlation of the errors.



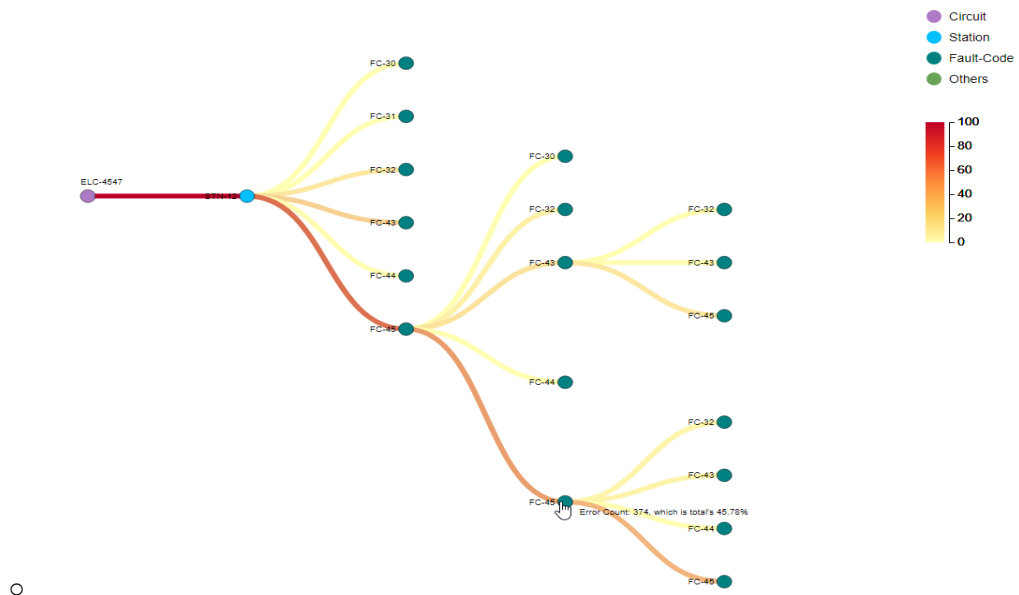
## Visualization 4: Fault-Code Tree for Single Element

The goal of this tree visualization to analyse the cause-effect correlation of the errors. The root of the tree is the selected element of production line. Each node in the fault-code level can be described as the source or causing error while its children are the affected errors.

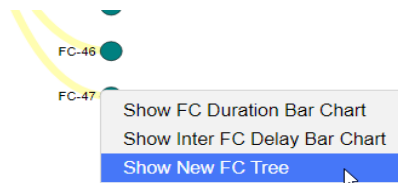
1. Right click on any elements and select “Show FC Tree” to see this visualization.



2. Click on the fault-code node (FC-XX) to expand the tree further. Click back on any parent node to collapse its children.



3. Right click on any fault-code node (FC-XX) and select “Show New FC Tree” to see new fault-code tree for the selected node.

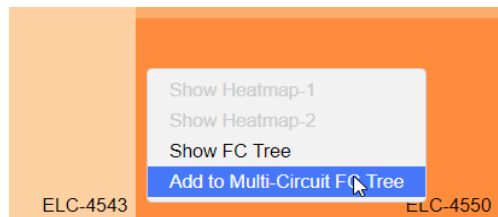


4. A tooltip provides the total number of the same errors and their percentages.

## Visualization 5: Fault-Code Tree for Multiple Electronic Circuits

This visualization is similar to the one above, but root of this tree visualization in this case is multiple electronic circuits together.

1. Right click on the electronic circuit element (top most view of treemap) and select “Add to Multi-Circuit FC Tree” to add this element to the selection list.



2. Repeat the above step to select more electronic circuits to the list. The selection is shown in a textbox on top of the treemap. Click on “Show FC Tree” button (which is on top of treemap) to see this visualization. Use “Clear” button to reset the selection.

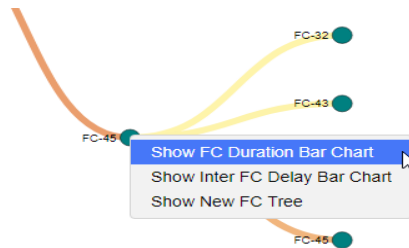


3. Other interactions are similar to the above visualization.

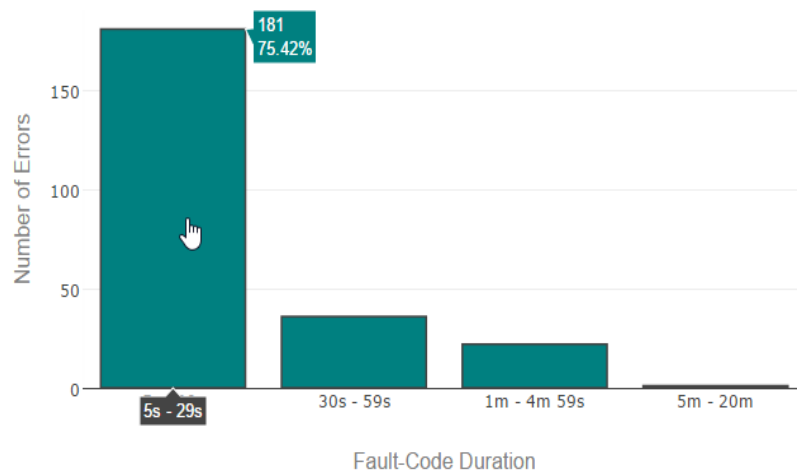
## Visualization 6: Fault-Code Duration Bar Chart

The goal of this visualization is to show the distribution of the durations of the errors for some given interval of time. This visualization is dependent on the Fault-Code Tree visualization and can be retrieved from any fault-code node of the tree.

1. Right click the fault-code nodes (FC-XX) and select “Show FC Duration Bar Chart” to see this visualization.



2. The values in x-axis are the intervals of duration while the ones in y-axis are the number of errors for that interval.



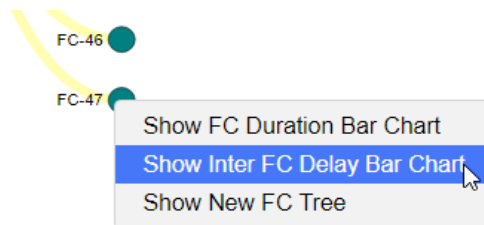
3. Tooltip provides with the number of errors and their percentage in given interval.



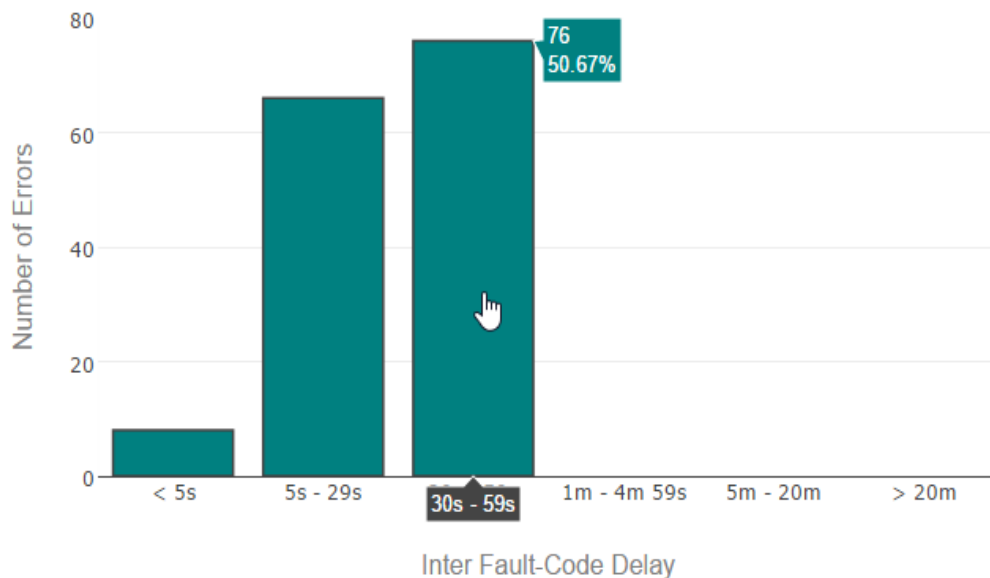
## Visualization 7: Inter Fault-Code Delay Bar Chart

The goal of this visualization is to show distribution the time difference between the correlated errors. This visualization is also dependent on the Fault-Code Tree visualization. As this deals with the inter fault-code time difference, it can be retrieved only on the children fault-code nodes i.e. which has at least another fault-code as parent.

1. Right click on any child fault-code nodes (FC-XX) and select “Show Inter FC Delay Bar Chart” to see this visualization.



2. The values in x-axis are the intervals of delay i.e. after the causer error, after how much time a particular error occurred. The values in y-axis are the number of errors for that interval.
3. Tooltip provides with the number of errors and their percentage in given interval.



# Cheat sheet for how-to

## Abbreviations:

ELC – Electronic Circuit	STN – Station	FC – Fault-Code
--------------------------	---------------	-----------------

### Vis – 1: ELC Block Diagram

- Use “Show ELC Bar Chart” and “Show ELC Block Diagram” radio buttons. ELC Bar Chart provides with tooltip information.

### Vis – 2: Treemap

- Each cell is clickable. Click to go deep.
- Top navigation bar to go up/back.
- Use context-menu for more visualizations.

### Vis – 3 and 4: Heatmap 1<sup>st</sup> and 2<sup>nd</sup> Order

- Can be created for station or fault-code from the treemap context-menu.
- X-axis values are source or causing error, Y-axis values are following or caused error.
- More information is shown with the tooltip.

### Vis – 5: Fault-Code Tree

- Can be created for any element. Use treemap context-menu.
- Any fault-code node is source or causing error and its children are following or caused error.
- Tooltip has more information.

### Vis – 6: Fault-Code Tree

- Can be created only for electronic circuit element. Use context-menu in treemap top most hierarchy.
- Any fault-code node is source or causing error and its children are following or caused error.
- The tooltip shows extra information.

### Vis – 7: Fault-Code Duration Bar Chart

- Dependent on fault-code tree visualization.
- Can be created for any fault-code node (FC-XX). Use context-menu in Fault-code tree.
- X-axis represents the intervals of duration of the fault-codes, y-axis has the number.
- More information is shown with the tooltip.

### Vis – 7: Inter Fault-Code Delay Bar Chart

- Dependent on fault-code tree visualization.
- Can be created for only child/leaf fault-code node (FC-XX) which has at least another fault-code as parent. Use context-menu in Fault-code tree.
- X-axis represents the intervals of delay between the fault-codes, y-axis has the number.
- More information is shown with the tooltip.

## B.2 Task List

Appendix B.2 presents the task-list used for the evaluation.

# Task List

Please try to complete the following tasks and answer the questions (if found useful).

## ELC Block Diagram

1. Click on the radio buttons to visualize both the block diagrams and answer the following two questions.

a. In ELC block diagram with circles, which electronic circuit do you think is most affected by errors? What is its position?

ELC-4548, Pos-6  ELC-4543, Pos-1  ELC-4544, Pos-2

b. In ELC bar chart, which electronic circuit do you think is least affected by errors? What is its error percentage?

ELC-4548, Pos-6  ELC-4543, Pos-1  ELC-4556, Pos-12

## Treemap

2. Please explore different hierarchy levels and also have a look at corresponding context-menu in the treemap.

## Heatmap 1st and 2nd Order

3. Try to generate both the 1st and 2nd order heatmap for STN-2 under ELC-4543 from the context-menu of the treemap and try to visualize the correlations.

a. From heatmap-1: what is the probability that if FC-11 occurs, the next error will be FC-11 too?

less than 30 %  30-70%  More than 70%

b. From heatmap-2: try to visualize the correlations of FC-46 > FC-11 and FC-47. What correlation did you find among them?

## Fault-Code Tree

3. Try to generate an FC Tree for FC-11 under STN-2 which is under ELC-4543 using the context-menu of the treemap.

a. Did you find any interesting correlation from the generated tree? If so, please specify.

### Multi-Circuit Fault-Code Tree

4. Try to generate a Multi-Circuit FC Tree for ELC-4544 and ELC-4545 using the context-menu of the treemap.

a. Try to find the fault-code pattern with highest probability and write down in the text box.

b. Please expand the tree to find fault-code pattern FC-47 > FC-47 > FC-47 > FC-47. What is its probability?

Less than 20%  20-60%  More than 60%

### Fault-Code Duration Bar Chart

5. Keep the FC Tree from 4. open and generate an FC Duration bar chart for the pattern from the last node of the pattern FC-47 > FC-47 > FC-47 > FC-47.

a. Could you analyze the durations of the above fault-code pattern? Which duration interval distribution is the longest in the bar chart?

5s - 29s  30s - 59s  1m - 5m  5m - 20m

### Inter Fault-Code Delay Bar Chart

6. Keep the FC Tree from 4. open and generate an Inter FC Delay bar chart for the pattern FC-47 > FC-47 > FC-47 > FC-47.

a. Could you analyze the inter-delay of the fault-codes? Which delay interval distribution is the longest in the bar chart?

less than 5s  5s - 29s  30s - 1m  1m - 5m

## B.3 Evaluation Questionnaire

Appendix B.3 presents the questionnaire used for the evaluation.

# Questionnaire: Evaluation Form for Master Thesis

## General Information

1. Gender \*

- Male
- Female

2. Age \*

3. Research Position

4. Research Field

5. Experience in visualization research (in years):

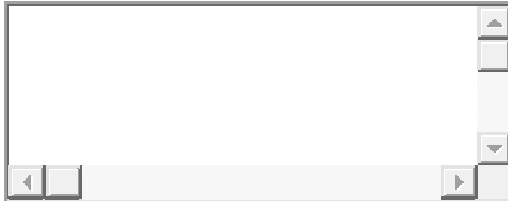




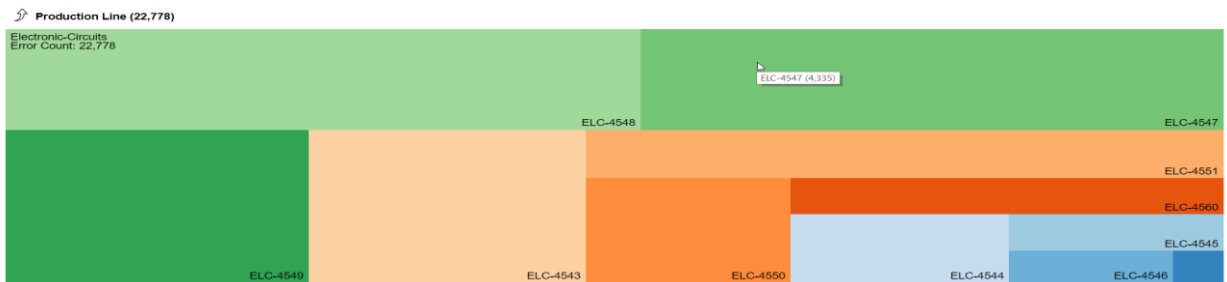
1. d) How helpful did you find the tooltip of the chart 2? \*

	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	
Not at all understood	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very well understood

1. e) Any suggestions or comments?



## Visualization 2: Treemap



2. a) How helpful did you find the above visualization to analyze and filter different elements of production line? \*

1 2 3 4 5

Not very helpful      Very helpful

2. b) How difficult was it to navigate between different hierarchy levels? \*

1 2 3 4 5

Very easy      Very difficult

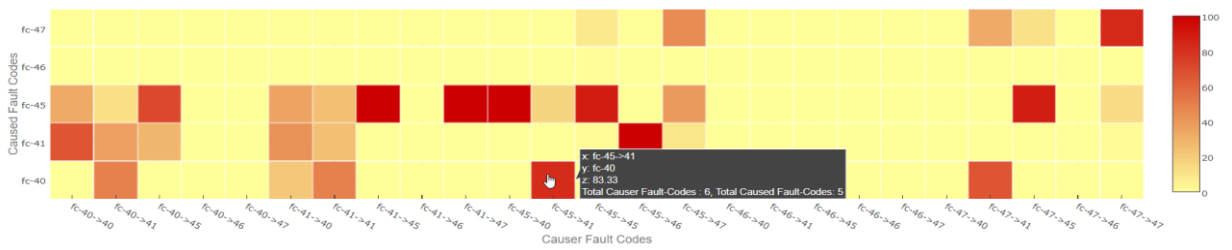
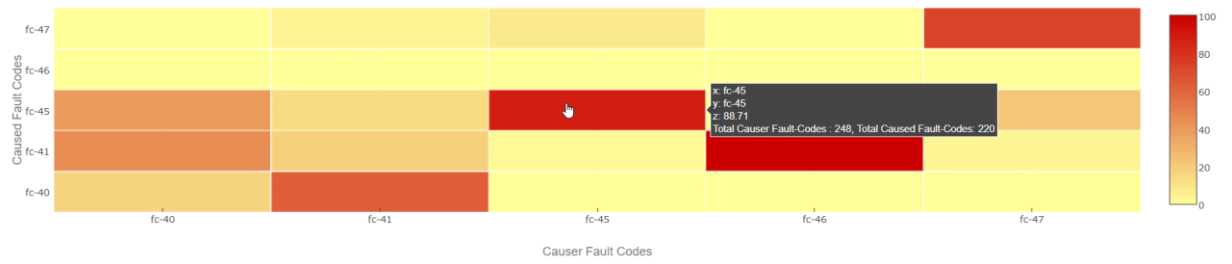
2. c) How difficult was it to use the context-menu of the above visualization? \*

1 2 3 4 5

Very easy      Very difficult

2. d) Any suggestions or comments?

### Visualization 3: Heatmap for 1st and 2nd Order Transition Matrix



3. a) How well did you understand the above visualization? \*

**1**
**2**
**3**
**4**
**5**

Not at all understood 
   
 
   
 
   
 
   
 
   
 Very well understood

3. b) How difficult was it to find the correlation of errors with the above visualization? \*

**1**
**2**
**3**
**4**
**5**

Very easy 
   
 
   
 
   
 
   
 
   
 Very difficult

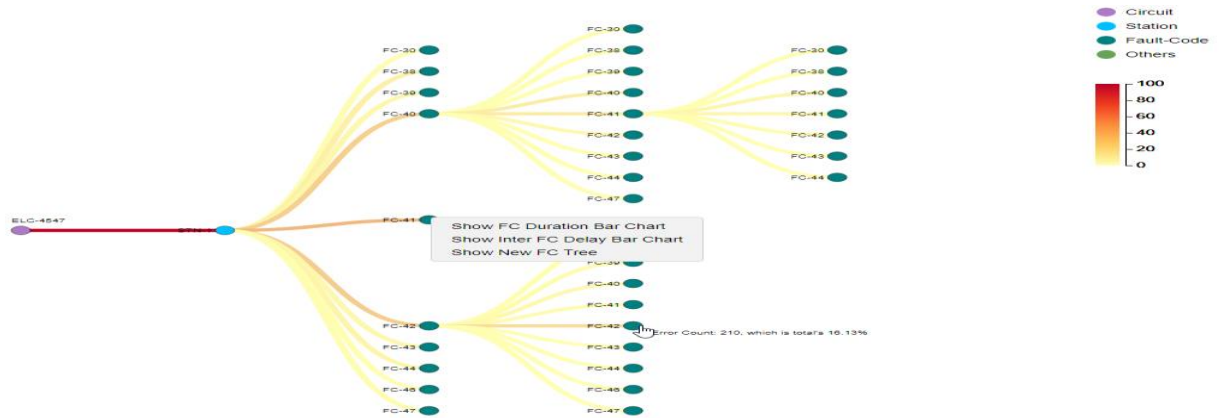
3. c) How useful was the tooltip of the above visualization? \*

**1**
**2**
**3**
**4**
**5**

Not very useful 
   
 
   
 
   
 
   
 
   
 Very useful

3. d) Any suggestions or comments?

## Visualization 4: Fault-code tree for single production line element



4. a) In general, how well did you understand the above visualization? \*

	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	
Not at all understood	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very well understood

4. b) How well did you understand the cause-effect relation of different errors with the above visualization? \*

	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	
Not at all understood	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very well understood

4. c) How difficult was it to find the correlation of errors with the above visualization? \*

	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	
Very easy	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very difficult

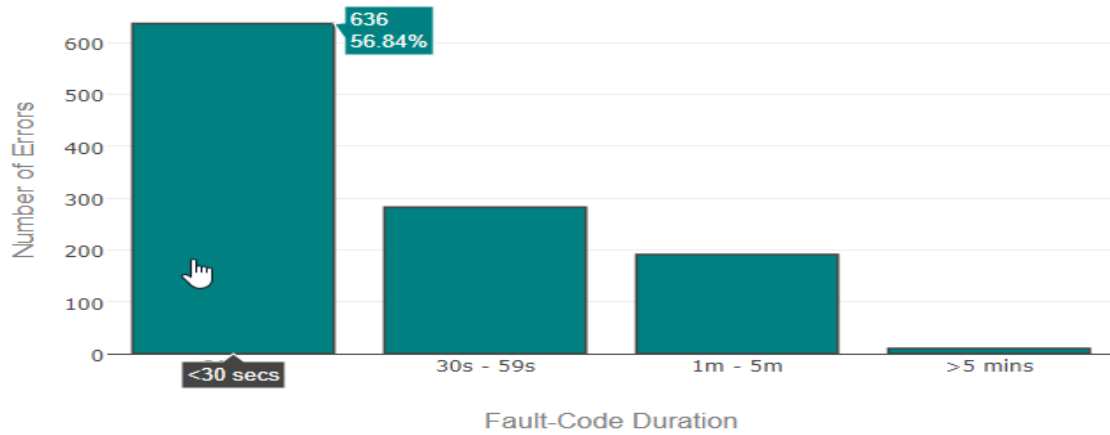
4. d) How useful was the tooltip of the above visualization? \*

	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	
Not very useful	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very useful

4. e) Any suggestions or comments?



## Visualization 6: Fault-code duration bar chart



6. a) How well did you understand the above visualization? \*

	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	
Not at all understood	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very well understood

6. b) How difficult was it to find the distribution of fault-code duration with the above visualization? \*

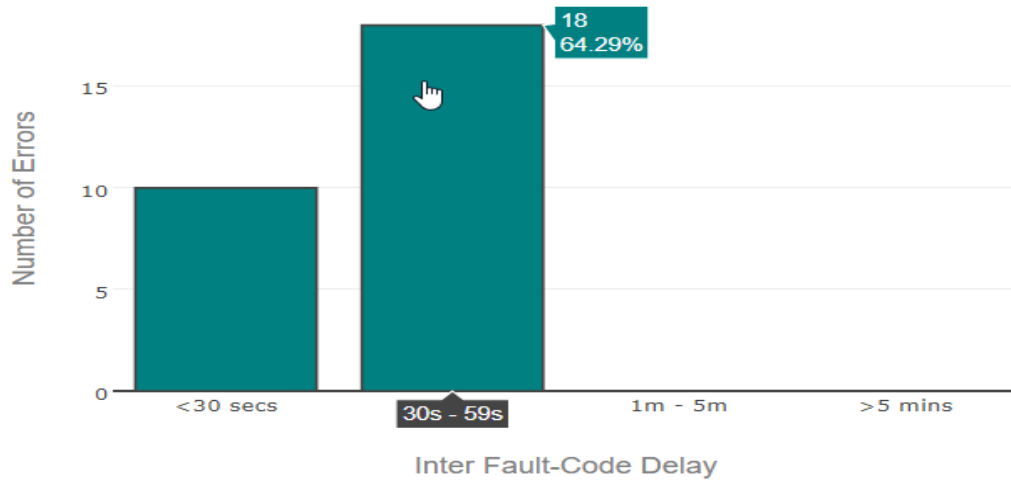
	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	
Very easy	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very difficult

6. c) How helpful did you find the above visualization to analyze the error? \*

	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	
Not very helpful	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very helpful

6. d) Any suggestions or comments?

## Visualization 7: Inter fault-code delay bar chart



7. a) How well did you understand the above visualization? \*

	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	
Not at all understood	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very well understood

7. b) How difficult was it to find the distribution of inter fault-code delay with the above visualization? \*

	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	
Very easy	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very difficult

7. c) How helpful did you find the above visualization to analyze the error? \*

	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	
Not very helpful	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very helpful

7. d) Any suggestions or comments?

# General Visualization Questions

1. Please rate each of the following visualization, in terms of finding the correlation of errors. \*

1    2    3    4    5

**Heatmap for 1st order transition matrix**

**Heatmap for 2nd order transition matrix**

**Fault-code tree for single element**

**Fault-code tree for multiple electronic circuits**

**Fault-code duration bar chart**

**Inter fault-code delay bar chart**

2. Please rate each of the following visualization, in terms of difficulty level to find the correlation of errors.

\*

1    2    3    4    5

**Heatmap for 1st order transition matrix**

**Heatmap for 2nd order transition matrix**

**Fault-code tree for single element**

**Fault-code tree for multiple electronic circuits**

**Fault-code duration bar chart**

**Inter fault-code delay bar chart**

3. Please rate each of the following visualization, in terms of tooltip information. \*

1    2    3    4    5

**Heatmap for 1st order transition matrix**

**Heatmap for 2nd order transition matrix**

**Fault-code tree for single element**

**Fault-code tree for multiple electronic circuits**

**Fault-code duration bar chart**

**Inter fault-code delay bar chart**



# Application-Specific Questions

1. How difficult was it to use the prototype? \*

Very easy      **1**      **2**      **3**      **4**      **5**      Very difficult

2. Did you encounter any problem or found any bug while using the prototype? \*

- Yes
- No
- Don't know

If yes, please specify.

5. Any final suggestions or comments?



# Bibliography

- [BBD08] M. Burch, F. Beck, S. Diehl. “Timeline trees: visualizing sequences of transactions in information hierarchies.” In: *Proceedings of the working conference on Advanced visual interfaces*. ACM. 2008, pp. 75–82 (cit. on pp. 37, 39).
- [Ber83] J. Bertin. “Semiology of graphics: Diagrams, networks, maps, trans.” In: *WJ Berg. Madison, WI: The University of Wisconsin Press, Ltd* (1983) (cit. on p. 24).
- [BHH03] C. A. Brewer, G. W. Hatchard, M. A. Harrower. “ColorBrewer in print: a catalog of color schemes for maps.” In: *Cartography and geographic information science* 30.1 (2003), pp. 5–32 (cit. on p. 33).
- [BHV00] M. Bruls, K. Huizing, J. J. Van Wijk. “Squarified treemaps.” In: *Data Visualization 2000*. Springer, 2000, pp. 33–42 (cit. on pp. 29, 30).
- [Bur13] S. Burge. “The Systems Thinking Tool Box.” In: *Retrieved May 1* (2013), p. 2016 (cit. on pp. 30, 31).
- [Cin13] E. Cinlar. *Introduction to stochastic processes*. Courier Corporation, 2013 (cit. on p. 17).
- [CMS99] S. K. Card, J. D. Mackinlay, B. Shneiderman. *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999 (cit. on p. 24).
- [CS11] J. Campbell, M. Shin. *Essentials of Geographic Information Systems*. 2011 (cit. on p. 50).
- [CW98] J. E. Cook, A. L. Wolf. “Discovering models of software processes from event-based data.” In: *ACM Transactions on Software Engineering and Methodology (TOSEM)* 7.3 (1998), pp. 215–249 (cit. on p. 35).
- [DL03] M. F. De Oliveira, H. Levkowitz. “From visual data exploration to visual data mining: a survey.” In: *IEEE Transactions on Visualization and Computer Graphics* 9.3 (2003), pp. 378–394 (cit. on p. 23).
- [Doo42] J. L. Doob. “What is a Stochastic Process?” In: *The American Mathematical Monthly* 49.10 (1942), pp. 648–653. ISSN: 00029890, 19300972. URL: <http://www.jstor.org/stable/2302572> (cit. on p. 17).

- [gartner20] Gartner. *Gartner IT Glossary*. 2020. URL: <http://www.gartner.com/it-glossary/interactive-visualization> (visited on 03/26/2017) (cit. on p. 25).
- [HFM07] N. Henry, J.-D. Fekete, M. J. McGuffin. “NodeTrix: a hybrid visualization of social networks.” In: *IEEE transactions on visualization and computer graphics* 13.6 (2007), pp. 1302–1309 (cit. on p. 84).
- [Hol06] D. Holten. “Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data.” In: *IEEE Transactions on Visualization and Computer Graphics* 12.5 (Sept. 2006), pp. 741–748. ISSN: 1077-2626. DOI: [10.1109/TVCG.2006.147](https://doi.org/10.1109/TVCG.2006.147) (cit. on p. 26).
- [How12] R. A. Howard. *Dynamic probabilistic systems: Markov models*. Vol. 1. Courier Corporation, 2012 (cit. on pp. 18, 20, 21).
- [ilstu17] ilstu. *CHOROPLETH MAPS, graded color series and nominal data*. 2017. URL: <http://my.ilstu.edu/~jrcarter/Geo204/Choro/Tom/> (visited on 06/21/2017) (cit. on p. 50).
- [Kac13] F. Kachapova. “Representing Markov chains with transition diagrams.” In: (2013) (cit. on pp. 22, 23).
- [KAF+08] D. Keim, G. Andrienko, J.-D. Fekete, C. Görg, J. Kohlhammer, G. Melançon. “Visual analytics: Definition, process, and challenges.” In: *Information visualization*. Springer, 2008, pp. 154–175 (cit. on p. 34).
- [KKEM10] D. Keim, J. Kohlhammer, G. Ellis, F. Mansmann. *Mastering the information age solving problems with visual analytics*. Eurographics Association, 2010 (cit. on p. 34).
- [LKL05] J. Lin, E. Keogh, S. Lonardi. “Visualizing and discovering non-trivial patterns in large time series databases.” In: *Information visualization* 4.2 (2005), pp. 61–82 (cit. on pp. 37, 38).
- [LRS83] S. E. Levinson, L. R. Rabiner, M. M. Sondhi. “An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition.” In: *The Bell System Technical Journal* 62.4 (1983), pp. 1035–1074 (cit. on p. 19).
- [Man10] L. Manovich. “What is visualization?” In: *paj: The Journal of the Initiative for Digital Humanities, Media, and Culture* 2.1 (2010) (cit. on p. 24).
- [Mun14] T. Munzner. *Visualization analysis and design*. CRC Press, 2014 (cit. on pp. 23, 25).
- [PMR+96] C. Plaisant, B. Milash, A. Rose, S. Widoff, B. Shneiderman. “LifeLines: visualizing personal histories.” In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM. 1996, pp. 221–227 (cit. on pp. 37, 38).

- [PW14] A. Perer, F. Wang. “Frequence: interactive mining and visualization of temporal frequent event sequences.” In: *Proceedings of the 19th international conference on Intelligent User Interfaces*. ACM. 2014, pp. 153–162 (cit. on pp. 36, 37).
- [SDC04] M. Smith, D. Davenport, T. Capone. *Treemap visualization engine*. US Patent App. 10/608,878. Dec. 2004. URL: <https://www.google.com/patents/US20040263513> (cit. on p. 27).
- [Shn92] B. Shneiderman. “Tree visualization with tree-maps: 2-d space-filling approach.” In: *ACM Transactions on graphics (TOG)* 11.1 (1992), pp. 92–99 (cit. on pp. 26, 27).
- [TC05] J. Thomas, K. Cook. “Illuminating the Path IEEE.” In: *Los Alamitos, CA, USA* (2005) (cit. on p. 34).
- [WG12] K. Wongsuphasawat, D. Gotz. “Exploring flow, factors, and outcomes of temporal event sequences with the outflow visualization.” In: *IEEE Transactions on Visualization and Computer Graphics* 18.12 (2012), pp. 2659–2668 (cit. on pp. 35, 36).
- [Zha15] J. Zhao. “Interactive Visual Data Exploration: A Multi-Focus Approach.” PhD thesis. Citeseer, 2015 (cit. on p. 25).

All links were last followed on July 5, 2017.



## **Declaration**

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

---

place, date, signature