VISUAL ANALYTIC SYSTEM FOR FAULT DETECTION AND DIAGNOSIS USING TIME SERIES DATA OF INDUSTRIAL PROCESSES

by

Gideon Pieter Slabbert

Submitted in partial fulfillment of the requirements for the degree

<missing degree code option>

in the

Department of Chemical Engineering

Faculty of Engineering, Built Environment and Information Technology

UNIVERSITY OF PRETORIA

October 2020

SUMMARY

VISUAL ANALYTIC SYSTEM FOR FAULT DETECTION AND DIAGNOSIS USING TIME SERIES DATA OF INDUSTRIAL PROCESSES

by

Gideon Pieter Slabbert

Supervisor:	Mr C Sandrock
Department:	Chemical Engineering
University:	University of Pretoria
Degree:	Master of Engineering (Control Engineering)
Keywords:	Fault Detection, Fault Diagnosis, Visual analytics, Time series

Summary paragraph one summary paragraph one summary paragraph one

LIST OF ABBREVIATIONS

FDD	Fault Detection and Diagnosis

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION

1.1 PROBLEM STATEMENT

1.1.1 Context of the problem

The modern industrial complex has a large number of automated processes. The automation of processes enables sustainable and predictable production. Control loops are one of the key building blocks that make automation possible.

Each sub section of a process consists of multiple control loops that together form a complex system. Continuous maintenance and fast response to disturbances is required to ensure that the overall goals of the automated process is sustained.

However, the complexities in analyzing the various number of control loops that form part of industrial control systems causes a time delay between the onset of a fault and the detection of that fault. Furthermore, once a fault is detected it can be very difficult to discern the source of a disturbance without time consuming interventions that add risk to process stability.[1]

A data driven approach is mostly applied in solving these kind of problems and a lot of focus is given in developing methods and tools that support this approach. In industry, the exact process model is rarely known or available when performing fault finding tasks using time series data.[2] Tools that only use routine operating data like the method developed by [3] provide an exciting new perspective on how to solve these complex problem using data driven methods. An approach that provides the user with an intuitive and interactive tool to analyze process behavior can add value to the fault detection and diagnosis process. By highlighting fundamental interaction paths using operating data it can translate into faster response times to address process instabilities.

A key element of any tool that assists in process fault finding is the visualization and interaction with the data. Every decision made in a industrial process involves reacting to process data. In modern industrial processes most short term decisions are made using the process data as visualized on the process control system. This same data is used to monitor and detect deviations in the process. When deviations are detected a fault finding methodology needs to be followed to determine what caused the deviation and how to correct it.

The fault finding method often requires a lot of data analysis using a certain frame of reference to determine what is normal behavior. With any process there needs to be some kind of visual interaction with the data in order for the human brain to process it. This visual interface can range from the simplest form of simply looking at how a point value on the control system changes over a period of time to constructing graphs of complex time series transformations of multiple data sets.

1.1.2 Research gap

The intent of this study is to explore approaches to the fault detection and diagnosis process with specific focus on the human interaction with data to analyze and solve a problem. The core element of this process is the visual context in which the data is viewed and transformed in order to generate understanding and knowledge.

In the modern industrial era a lot of fault detection problems are solved using computer based tools that are very efficient at processing large amounts of data and performing complex data transformations in a relatively short period of time.

A time series fault detection and diagnosis problem is often solved by creating a specialized tool that would transform data into knowledge which would most likely have a visual representation as the end result. The problem and solution are then presented as the end result of the study. If the audience then want to re-produce the results or test the method on a different data set they have to reproduce the tool that was used to generate the knowledge as well. This is a time consuming process and a barrier to further development of the proposed method.

A researcher might also make the specialized tool available to the audience but if the tool was developed on a platform that uses proprietary code then it also presents a barrier to further development. There are a lot of tools available for visualization of data but very few of them are built on open source

platforms. Recent development in the computer science industry has made available a large number of open source tools and platforms, but there are very few examples in literature of them being applied in the visual analytics realm.

New developments in the fault detection and diagnosis problem have presented many new views of how to identify that a fault has occurred or how to identify the root cause of a fault. These new methods display the results in static graphs that highlight the most important elements of the transformed data set. One key element of the user experience that still has a lot of potential for improvement is the ability to interact with the data. Interaction allows for faster knowledge transfer and easier understanding of how the parameters used in the transformation effect the final result. If the audience could be able to interact with the presented data at every step of the transformation process then a much faster and deeper understanding could be achieved.

1.2 RESEARCH OBJECTIVE AND QUESTIONS

The focus of this dissertation will be to explore the ways in which visual analytic systems can be used to aid in the fault detection and diagnosis (FDD) process.

FDD will be discussed in general and with regards to new methods that have been developed that attempt to solve the challenges to this process. Questions that will be explored in this study will include:

- What methods have been developed for fault detection and diagnosis and how are they used in the industrial landscape?
- What role does visual analytics play in the fault detection and diagnosis process?
- What tools are currently available for visualization?
- How can the FDD process be improved using visualization techniques and open source software?

1.3 HYPOTHESIS AND APPROACH

It is assumed that there is a logical thought process that is applied when doing FDD. This logical thought process combined with proven analytical methods can provide accurate an actionable results to the FDD process. Visual analytics plays a key role in this process and the different ways that data is collected, processed and transformed into knowledge about the existence of a fault and where the fault is located.

The approach taken in this study will be to outline how the FFD process can be completed using time series data and graphical knowledge of the plant topology to formulate a method for detecting and diagnosis process faults. Visual analytics will be used to facilitate this process and the interaction with data in a way that guides the user in the FDD process.

1.4 RESEARCH GOALS

The goal of the study will be to highlight key aspects of the FDD process and the visual analytics systems available to use in this process. Different ways in which to develop these visual systems will be explored with a strong focus on developing computer aided tools that will facilitate a visual fault finding process.

1.5 RESEARCH CONTRIBUTION

One particular fault finding technique that involves complex transformations of data to determine the root cause of disturbances will be discussed and demonstrated. A visual analytic system will be developed based on methodologies explored in the research portion of this study to attempt improving the user experience when analyzing a complex control problem.

1.6 OVERVIEW OF STUDY

Chapter 2 will provide an overview of the FDD process, how it is used in industry and where new methods fits into the historical framework. This chapter will also explore how visualization is used in the fault finding process and how visual analytics support and facilitate FDD.

Chapter 3 will contain the results of work done to create a open source visualization tool that incorporates the insight obtained from Chapter 2.

Chapter 4 will demonstrate the results from applying this tool the results obtained from applying the analytical methods described in Chapter 2. This chapter will also discuss these results and how they improve the FDD process.

Chapter 5 will summarize the literature review and contribution made.

CHAPTER 2 LITERATURE STUDY

2.1 CHAPTER OBJECTIVES

This chapter explores literature in the field of fault detection and diagnosis with a specific focus on application in industrial processes. It then elaborates on visualization techniques and how they could be used to aid in the fault detection and diagnosis process.

2.2 FAULT DETECTION AND DIAGNOSIS

The automated control of industrial processes has developed significantly in recent decades. Advances in computing power had allowed the distributed control system to become common place in most industrial processes. This together with advanced control techniques like model predictive control has allowed for automated real time optimization of the key economic variables. Despite these advances one of the most important tasks in managing a process is still being done manually by operational teams; responding to abnormal events. The timely identification and diagnosis of the root cause of an abnormality, and the appropriate supervisory action to bring the process back to stability, is almost a daily challenge to operational teams. [4] defined this task as abnormal event management (AEM) which is critical to the stability of a process.

[4] refers to statistics that indicate that 70 % of industrial accidents are caused by human errors. This indicates that methods to improve this task can significantly improve the economic, safety and environmental impact of these industries.

A fault can be described as a deviation of an observable variable or calculated parameter from its acceptable operating range [4]. This can be a process deviation like low steam outlet temperatures from a boiler. A possible cause for this could be a failure on a attemperator valve which can be referred to as the basic event or root cause. The diagnosis task can be seen as a classification problem wherein a

root cause or fault must be placed in a certain group of faults which will have a pe-determined method to solve them.

Disturbances in single control loops are fairly straight forward to solve but disturbances that propagate through a complex system on a plant wide scale present a much bigger challenge. One of the goals of process control is to shift process variability to places like buffer tanks and plant utilities. In modern industrial processes these options have become limited as inventory is usually kept to a minimum and the use of recycle streams for heat integration has introduced extra sources of complex interaction [1].

The time frame over which the disturbance takes place can be used to distinguish between different types of disturbances. [1] classified the types of disturbances in a plant wide setting into three main categories with regards to timescales:

1. slowly-developing, i.e. fouling of a heat exchanger
2. persistent and dynamic
3. abrupt, i.e. a compressor trip

The first item, slowly-developing disturbances are fairly easy to analyze and very predictable in nature. They are often times expected as a result of mechanical degradation over time or inherent characteristics of the process involved. These kinds of disturbances are normally factored into the expected plant availability statistics and can be handled with proper planning and maintenance philosophies.

The second item, persistent and dynamic disturbances that propagate through an entire system, are of interest for this study. These type of disturbances introduce a continuous operational challenge as they can manifest at random and are normally the first signs of instability that leads to the third item listed; abrupt disturbances. There persistent and dynamic disturbances are usually caused by components of a system that are starting to fail but have not yet reached a severity level where they cause the process to move out of acceptable operational bounds.

Abrupt disturbances can also occur sporadically due to component failures that immediately result in a complete system failure. These disturbances are also fairly easy to eliminate after the faulty component is identified and a solution is developed to out engineer the failure.

2.2.1 FDD AS ANALYTICAL REDUNDANCY

Fault detection and diagnosis is normally used to improve the reliability of a system [5]. There are two methods that can be used to accomplish fault diagnosis: hardware redundancy and software or analytical redundancy. Hardware redundancy requires the use of identical components that use the same input signals to generate duplicate output signals. If the output signal then differs between the two redundant systems it is easy to detect a fault and take action, examples of this method is limit checking and majority voting. Hardware redundancy is often times not present due to cost considerations or only present on the most critical sensor elements. This creates the opportunity for faults to occur on any of the other non-redundant elements and then propagate through the whole system. For this reason analytical redundancy has become the main method for doing fault diagnosis.

In this context of analytical redundancy, fault diagnosis includes three components; fault detection, fault isolation and fault identification [5]. Fault diagnosis starts with detecting that a fault has occurred, after detection the fault needs to be isolated by providing information as to where in the system the fault is located. Finally the fault identification step provides information on the types, shapes and size of the fault which basically provides the degree or level of malfunction.

2.2.2 CLASSIFYING FDD METHODS

[5] completed a comprehensive overview of fault diagnosis techniques that have been developed over the last 30 years with specific focus on highlighting new developments over the last 10 years. According to this study by [5], fault detection and diagnosis techniques can be divided into three methods; model-based methods, signal-based methods and knowledge-based methods.

Model based methods use a model of the system derived from physical principles or system identification techniques. The system outputs are then compared to model predicted outputs to identify faults.

Signal based methods do not rely on input- output relationships but rather extract features from the signals measured in the system. Diagnostic decisions are then taken by comparing these key features to prior healthy versions of the same features. These methods can be time-domain based like computing the mean or standard deviation. They can also be frequency-domain based by looking at the spectrum of signals such as the discrete Fourier transform.

[6] discussed the knowledge based methods in detail in the second part of the overview paper. Knowledge based methods are also referred to as data driven methods. These methods can be either qualitative or quantitative in nature. They require large amounts of historical process data and computing power to extract process knowledge. These methods can be further divided into qualitative and quantitative methods.

Qualitative knowledge based fault diagnosis is expert-system methods that rely on sets of rules to evaluate data and detect faults. The rules have to be developed specifically for the system in question but integrating this method with a object-orientated paradigm provided more flexibility by developing more general rules. Another method used with this description is qualitative trend analysis (QTA) which identifies process trends from noisy data and compares them to a database that contains fault trends. The QTA method has been advanced by combining it with integrated signed directed graphs (SDG) to compensate for some of the disadvantages of this method.

Quantitative knowledge based methods are based on solving the pattern recognition problem. This can be done using statistical analysis techniques like principal component analysis (PCA), partial least squares (PLS), independent component analysis (ICA) or support vector machine (SVM). This can also be done using non-statistical analysis based methods which are better suited for non-linear applications, an example of this is a neural network.

A lot of new research is being done in quantitative knowledge based methods due to the vast amount of data that is now available in modern industrial complexes. [7] defines these methods as multivariate statistical process monitoring (MSPM). For linear Gaussian process methods such as PCA, PLS and canonical correlation analysis CCA are used. To cater for non-Gaussian, non-linear and dynamic characteristics numerous extensions have been developed for the above mentioned methods.

2.2.3 EVALUATING FDD METHODS AND STRUCTURE

When discussing FDD methods it is important to look at how they are evaluated. Two very important concepts to understand when evaluating fault FFD methods is completeness and resolution [4]. Completeness describes how well a method is at detecting all the possible faults exhibited by the system. Resolution then relates to how well the method is at classifying the fault into a small subset of the available faults. A diagnostic system has the following desirable characteristics according to [4].

1. Quick detection and diagnosis - creates a trade-off between robustness and performance
2. Isolability - ability to distinguish between different failures
3. Robustness - robust to various noise and uncertainties
4. Novelty identifiability - can detect a new/novel fault in the process
5. Classification error estimate - error measures to project confidence levels
6. Adaptability - be able to develop new scope as new problems/faults arises and more information becomes available
7. Explanation facility - provide explanation of how a fault originated and propagated. Ability to reason cause and effect relationships in the process.
8. Modeling requirements - modeling effort required should be minimal
9. Storage and computational requirements - achieve balance between computational complexity and storage requirements
10. Multiple fault identifiability

In any diagnostic system the measurements and data goes through a transformation process. Two important components to take note of when considering transformations are the a priori process knowledge and the search technique used [4]. In general the transformation of data will go through four steps:

1. Measurement space
2. Feature space
3. Decision space
4. Class space

The measurement space contains the raw input signals to the diagnostic system. The feature space contains variables that are derived from the measurement space using a priori problem knowledge. Useful features are extracted, in this transformation there is often a significant dimension reduction. This is normally done by using feature selection or feature extraction. Its assumed that features cluster better than measurements which will facilitate improved classification or better discrimination. By extracting discriminant features it takes the burden of the next transformation so that it can easily find the features or distinguish between features that match certain decisions.

The mapping from feature to decision space is usually done via some sort of objective function or using simple threshold functions. The class space consist of the list of possible faults and the decision space maps to it via threshold functions, template matching or symbolic reasoning. This could be summarized as using some sort of search or learning algorithm to complete these mappings. It is also the final interpretation of the diagnostic system delivered to the user. The decision space and the class space usually have the same dimensions. This could provide the option of combining these two spaces but the advantage of still keeping them separate is that a certain diagnostic method might not be able to give a crisp solution as to what fault is being detected.

A priori knowledge plays a crucial role through this entire transformation process, it can greatly simplify the problem in every step of the process. Sets of failures and relationships between observations and failures is crucial to the success of any method. This relationship can be inferred, the most basic example being a lookup table or it can be based on domain knowledge. First principle knowledge is referred to as deep, casual and model-based knowledge whereas knowledge gained from past experience with the process can be referred to as shallow, compiled, evidential or process history-based knowledge [4].

2.2.4 NEW APPROACHES TO DATA DRIVEN METHODS

In the following sections of this exploration more focus will be given to quantitative knowledge/ data driven methods. Improvements in computing power and better access to tools used for analyzing data make it possible to significantly improve the contribution of data driven methods to the FDD process.

The most well known statistical methods in this field like PCA, PLS or CCA have the limitation of being very dependent on linear relationships. This works well when a process or system is operating in a region where linear interactions are present. Most control techniques also have this same requirement. It is then not surprising that the factor that challenges control systems is in fact often non-linear interactions that can manifest at some operating points or during some operating conditions. A control valve that operates in the upper or lower ranges of its range can exhibit extreme non-linearity. A worn out component like a actuator can cause big shifts in an input output relationships on direction changes. For this reason it is worthwhile investigating new methods that have shown some potential at solving these difficult problems. One of these methods is the use of the PageRank algorithm [8] employed by google on the ranking of control loops based on the degree of interaction.[3] has developed a

methodology called LoopRank with the aim of prioritizing control loop maintenance by highlighting loops that have a higher correlation with other loops.

2.2.5 PAGERANK DERIVATION

The ranking algorithm developed by [8] started with the idea of ranking web sites depending on how many times they are referenced by other websites.

Consider a group of 4 websites that contain references to each other in the manner indicated in Figure 2.1. References are indicated by arrows.

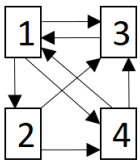

Figure 2.1. web pages with references

An initial idea would be that a page that has more references or back links is more important. Let x_k be the importance of page k. If this method is followed then $x_3 = 3$ would be the most important web page because it has the most references.

This initial ranking method indicates that x_3 is important, but it doesn't take into account the importance that x_3 passes on to x_1 because it is the only page referenced by x_3.

Another approach would be to let back links from important pages count more than back links from less important pages. If a page k has a back link from page j, then the score given to this back link can be calculated as $x_k = \frac{x_j}{n_j}$ where n_j is the total number of references made by page j. The total score for page x_k is then calculated by the sum of all back links to it adjusted for the number of references made. Applying this new method the importance score for each node is calculated:

$x_1 = \frac{x_3}{1} + \frac{x_4}{2}$

$x_2 = \frac{x_1}{3}$

$x_3 = \frac{x_1}{3} + \frac{x_2}{2} + \frac{x_4}{2}$

$x_4 = \frac{x_1}{3} + \frac{x_2}{2}$

These equations can be written in a matrix form, which will be defined as the link matrix A:

$$A = \begin{bmatrix} 0 & 0 & 1 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 \end{bmatrix}$$

Each row in the matrix A represents a page or node while each column represents the importance score from another node. For example row one represents node 1 and it has a back link from node 3 with an importance score of 1 and a back link from node 4 with a importance score of $\frac{1}{2}$.

if $x = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix}^T$ then the problem can be defined as the eigenvector problem:
$Ax = x\lambda$
with x being an eigenvector for A with en eigenvalue of 1
if A is a $n \times n$ matrix, then a nonzero vector x is an eigenvector if $Ax = \lambda x$ for some scalar λ. The eigenvector x that solves this equation is then calculated to be:
$x \approx \begin{bmatrix} 0.387 & 0.129 & 0.290 & 0.194 \end{bmatrix}^T$ which shows that node one is in fact more important than node three.

2.2.6 LOOPRANK ADAPTION

[3] developed a method called LoopRank which is based on the PageRank algorithm to prioritize control loop maintenance. Modern industrial sites have a large number of control loops which becomes challenging to maintain effectively given limited technical resources.

The LoopRank method can be summarized in five steps:

1. Collect time series data of control loop parameters
2. Calculate the partial correlation between all the variables in the data set. This information is used to build a relative weight matrix A or link matrix as referred to in the PageRank algorithm
3. Compute the LoopRank based on the relative weight matrix A using the eigenvector centrality theorem
4. Multiply the result for each loop with a pre-defined weight that is based on the criticality of the loop. This weight is assigned heuristically depending on loop type and profitability
5. Normalize the results to express the score for each loop as a percentage

[3] demonstrated the efficacy of LoopRank by testing it on an industrial case study in which it successfully flagged the most critical control loops.

2.2.7 LOOPRANK DEVELOPMENT

[2] compared LoopRank with a more traditional data driven method that utilizes the integral of absolute or squared error to determine loop interaction. Canonical correlation was used to calculate the relative weight matrix in the LoopRank methodology. It was found that both methods delivered similar results.

The LoopRank methodology was further expanded on by [9] with the goal of improving the relative weight matrix used in step 2 of the procedure as outlined by [3].

[9] demonstrated how transfer entropy provides a more robust estimation of the casual links between nodes in a relative weight matrix. This method was tested using data from the well known Tennessee Eastman case study and was able to successfully highlight the disturbance variables as the top ranking nodes in the matrix.

[10] further developed the LoopRank methodology to calculate importance scores for multiple overlapping time regions. Studying how the importance scores change over time then provided a very useful method for performing incipient fault detection.

2.3 VISUALIZATION

The methodologies involved in fault detection and diagnosis of industrial processes rely heavily on the visual analysis of time series data. Data is continuously generated by industrial processes, this data in its simplest form can be floating point or boolean values on a control system interface screen. This data is used by plant personnel as feedback on the current status of automated processes.

This study will focus on methods and approaches taken to visualize temporal or time series data.

[11] described visualization as the graphical representation of information where data visualization allows for exploration and communication. [12] added that the purpose of visualization is to gain insight into some process represented in the form of quantitative data by means of interactive graphics. Insight is defined by [12] as reference to two types of information; answer questions about data that

describes a particular problem and facts about the problem that the user might not have been aware of.[20] described visual analytics as a process wherein user gain insight into complex data.

Some of the first books to discuss information visualization date back to 1967 with the publication of Jacques Bertin's Semiologie Graphique [13]. A survey by [13] highlighted the number of visualization books have been published each year, displayed in 2.2.

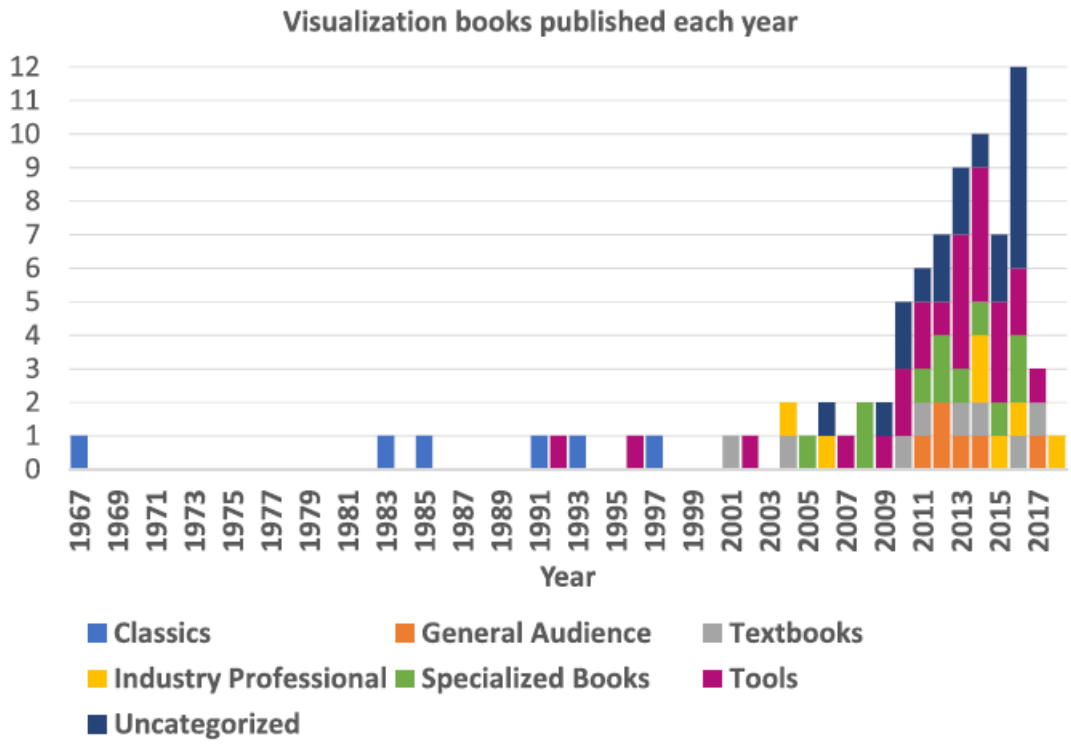

Figure 2.2. Visualization books published each year

The IEEE Visualization conference series aimed at sharing developments in this field started in 1990 and has contributed a large amount of data on how visualization assists other fields to better understand their data. [14] compiled a database of these publications to assist researchers in this field to understand the history of visualization in a field and make the researcher aware of new developments trends. The papers presented through these conferences serve as valuable source to describe previous work related to visualization for the use of fault detection and diagnosis.

2.3.1 VISUAL ANALYTICS

Before designing it is important to understand the mental processes behind data analytics that are used by people to understand and gain insight into data. The work discussed in this section first attempts to

describe the human aspect of of the process and then attempts to integrate the role of the machine or computer into the process.

One of the earliest models for that describe the process of 'sensemaking' was developed by [15]. In this model intelligence analysis or sensemaking consists of:

1. Information gathering
2. Representation of the information in the form of an outline or model that aids the analysis
3. Development of insight through manipulation of the outline or model
4. Creation of a knowledge product or direct action based on the analysis

The process is visually described by Figure 2.3.

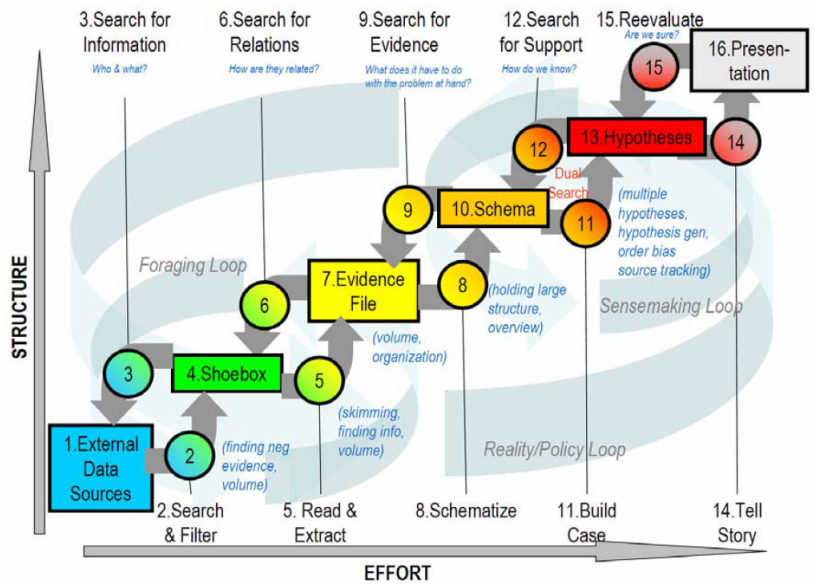

Figure 2.3. [15] sensemaking model

The rectangles represent the flow of data and the circles represent the process flow. What is very clear from this model is that there are a lot of back loops and sets of activities that cycle around finding information and making sense of the information. The end result of the process is the generation of novel information by the analyst.

The external data sources are filtered into a smaller data set called a shoebox. Snippets are then extracted from the shoe box based on observed relationships and placed in the evidence file. A schema or model is then developed and verified using the evidence file. A hypotheses is then formed based on this model with a feedback loop that searches for additional support of this hypotheses. Finally there is a presentation that is first re-evaluated before becoming the product of this work. This model can be used in a top-down (theory to data) or bottom-up (data to theory) approach.

Another approach would be to present a more non-linear model that might be a closer representation to how humans actually process information. [16] presents a model that starts of with the assumption that people always start making sense of something with some perspective, viewpoint or framework. The frame basically functions as hypothesis of how the data fits together.

These frameworks or frames actually define what counts as data and shape the data. Frames also change as we acquire data making it a two way path between a frame and data.

The first cycle that starts from the frame/ data idea is the elaboration cycle. In this cycle the frame is expanded and questioned - there could be doubt about the explanation it provides. If the doubt, possibly caused by troublesome data, is explained away then the frame is preserved.

The second cycle involves a re-framing process. If questioning the frame leads to its rejection, it might lead to its replacement with a better one. Frames could be compared to see which one fits the data best. This model is displayed in Figure2.4. This is a continuous process in which a frame is strengthened or improved at each iteration.

Any method for turning data into knowledge requires the manipulation of the data. Data analysis has traditionally been done manually but with the exponential growth in the amount of data that can form part of an analysis other methods are being developed to make this step more efficient. [17] analyzed this challenge in the emerging field of knowledge discovery in databases (KDD). In simple terms this field is concerned with development of methods and techniques for making sense of data.

One of the most noteworthy fields where these methods are applied is in the area of astronomy. Astronomy contains terabytes of data contained in images. Although industrial application of data

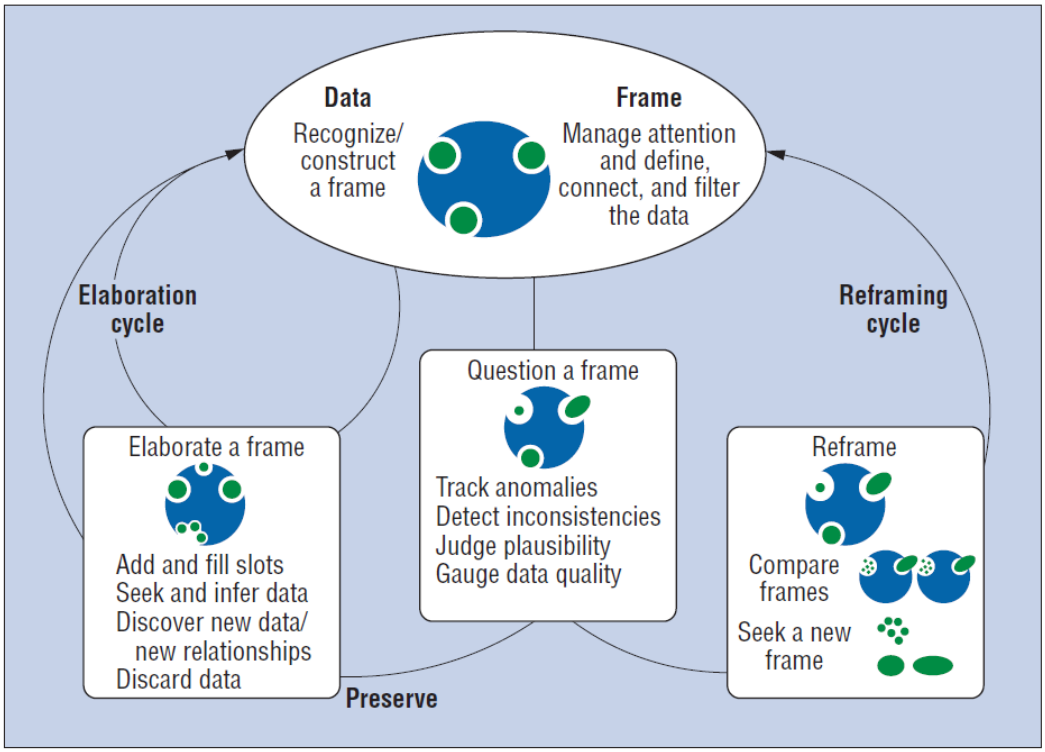

Figure 2.4. [16] frame-data sensemaking model

analysis contains a lot less data than this it could still be worth while studying these techniques to see if they can assist with effectively utilizing large amounts of time series data.

The goal of KDD is extracting high-level knowledge from low-level data in the context of large data sets [17]. The KDD process is visually described by Figure 2.5. The process is described by the following steps:

1. Develop understanding of application domain and use prior knowledge to determine the goal of the KDD
2. Create a data set or subset of the available data that will be used to discover useful information
3. Data cleaning, in this step invalid data like noise of questionable data is removed in order to prevent false deductions
4. Data reduction and projection - finding useful features to represent the data, in this step dimensional reduction can be done to simplify the analysis
5. Match the goals of KDD to a specific data mining method

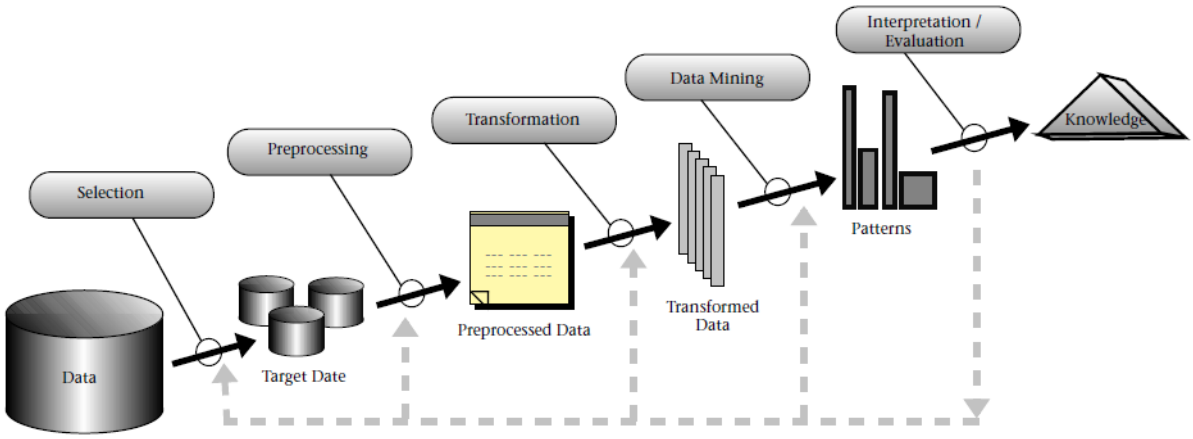

Figure 2.5. [17] steps in the knowledge discovery in databases model

6. Exploratory analysis of the model and hypothesis selection, in this step the data mining method is chosen as well as the methods that will be used to search for patterns
7. Data mining - searching for patterns in the data that would confirm the hypothesis
8. Interpreting the mined patters
9. Acting on the discovered knowledge, using it to solve the problem

These steps describe a linear process but there can be movement back and forth between the steps to refine the process.

[18] goes a step further to describe the process in which knowledge is generated from data by developing a model that defines how the human and computer interact. This model is split into two parts:

1. computer system which involves the data, visualization and analytic models
2. human component that describes the cognitive process associated with an analytical session

What makes visual analytics so effective is the integration of these two parts; the human side provides the perceptive skills, cognitive reasoning and domain knowledge while the computer side provides the computing data storage capability [18]. This visual analytic model also attempts to incorporate some of the other models described earlier like the Sensemaking model [15] and the Knowledge Discovery

Process in Databases (KDD) [17]. The complete knowledge generation model with the reference to other models is displayed in Figure 2.6.

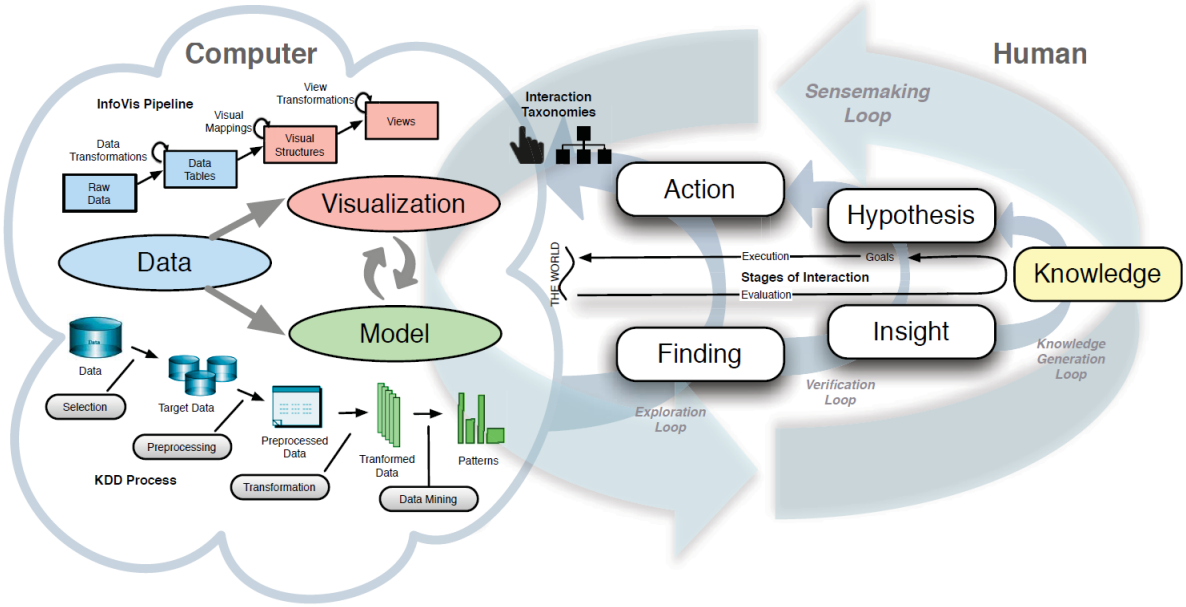

Figure 2.6. [18] computer-human interaction in knowledge generating process

The computer part of the model has three main elements:

1. Data - this includes all the data that describes the analytical problem and includes facts in any structured, semi-structured or unstructured manner. Another element to this is a term called meta data, which could be data that describes the structure of other data or data that summarizes other data.
2. Model - This can be any descriptive measure of the data ranging from statistical properties of the data to complex data mining algorithms. This process can deliver a single number to describe the strength of an hypothesis or a complex pattern.
3. Visualization - In this process the data is transformed into some kind of visual artifact that enables the analyst to detect relationships.

The human element is also described in three main loops: The computer part of the model has three main elements:

1. Exploration loop - findings are made based on observations of the visual analytics system, Actions or individual tasks are then performed that result in tangible and unique responses from the visual analytics system.
2. Verification loop - insight is gained from interpreting findings, often with previous domain knowledge to generate units of information. A hypothesis is then formed based on these insights and tested in the visual analytics system through actions.
3. Knowledge - The final part is to build up confidence in the insights from the data and hypotheses tested on the data. If patterns or stories told by the data makes sense and can be used to answer the questions from the visual analytic process then knowledge is generated. This knowledge is added to the domain knowledge of the analyst to ask more questions and generate more knowledge.

This model proposed by [18] again highlights the iterative process of visual analytics and how feedback loops are used in every element of the model.

2.3.2 VALUE OF RECORD KEEPING IN VISUALIZATION

[19] conducted a experiment to study the visual analytics process and the role of record keeping in this process. This study also elaborated on key elements for the design of a collaborative visual analysis tool. The study required groups to complete tasks given a certain data set using computer aided visual analytics. Actions taken during visual analytics were grouped into four phases:

1. Problem definition - understanding what needs to be solved
2. Visualization - generating visual artifacts
3. Analysis - most complex phase that requires examining of charts and making sense of them combined with other data
4. Dissemination - compiling findings in a report

The findings from the study conducted by [19] confirmed observations by other researchers about the non-linear temporal order of activities. While completing the tasks users would move through the different steps in no specific order. The visualization and analysis phases were strongly interrelated, the users moved back and forth between these two phases much more than the other phases.

[19] generated a very visual model of these four phases with key elements noted in each phase:

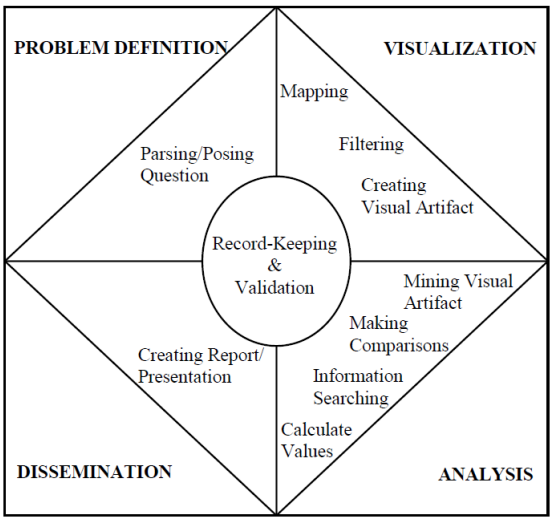

Figure 2.7. Visualization books published each year

Record keeping involves the saving of visual artifacts and notes. In most cases a visual artifact can convey a complex idea more effectively than notes. Visual artifacts were saved for two main reasons; using them in reporting and using them to enhance the analysis process.[19] recommends that visual analytic systems provide functionality to save visual artifacts like charts for later use and comparison. Note taking also plays a critical role in all phases of the process. The notes high level content could be divided into two categories; findings and cues. Finding are a result of mathematical calculations, statistical operations or decisions and outcomes of the analysis process, they could also take the form of saved charts. Cues can be anything that is not directly extracted from the visualizations that could assist the user in framing or understanding the task at hand. A larger proportion of findings were recorded in the analysis phase while the cues were largely taken during the visualization phase. The study concluded that record keeping is used intensively by data analysts.

[20] completed a longitudinal study of a bioinformatics data set analysis to gain understanding of the entire analysis process from raw data to insights. The study highlighted how analysts gather and build insights over long periods of time and connecting the data to extensive domain knowledge and expertise. It is emphasized that analyst spend a significant amount of time on manual manipulation of data and this creates a very important need for the visualization tool to be interactive. Inferior visualizations with interaction were preferred over superior static visualizations.

2.3.3 POWER OF VISUAL ANALYTICS

Thoughts to expand on further: Practical side - how long does each step in analysis take, machine learning, 60 percent of time taken to clean data

Visualization critical for outlier detection - visual anomaly detection system yet to be beaten

2.3.4 VISUALIZATION TOOLS

[21] performed a comparison of visualization tools in the development of a tool that could be used for data analysis at CERN. The goal of this tool was to enable scientist to observe a data set as a whole in order to make better decisions of what variable pairs and time frames to analyze during test campaigns.

JavaScript and Python Libraries were investigated and evaluated. D3 and Dygraphs were both discussed as JavaScript libraries that can be used for plotting. Matplotlib was discussed as a Python library that is based on the plotting functionalities of MATLAB. Two other Python libraries, Bokeh and Mpld3 were discussed with specific attention to their capabilities to generate HTML which enables them to easily present graphs on a web browser. Furthermore Mpld3 can integrate Matplotlob plots using D3 to render visualizations on a web page.

[21] did a detailed comparison of Bokeh and Mpld3 which identified Bokeh as their preferred choice. Bokeh excelled at key aspects like loading time, rendering time and available features and had more actively developed libraries.

[22] did a comparison of data visualization systems with the specific focus on big data applications. Tools were categorized into coding and zero coding tools with graphical user interfaces.

Within the coding category two languages were discussed; Python and R. Within Python multiple libraries were analyzed for visualization, including Bokeh, Altair, Seaborn, ggplot and Pygal. The zero coding category included Tableau, Infogram, Charblocks, Datawrapper, Plotly, RAW and Visual. It's important to note that of these zero coding tools Plotly is also a Python library used by the data science community.

[23] also completed an overview of tools used for big data visualization and the challenges faced with attempting to visualization large amounts of data. The tools were compared using the following criteria:

1. Open Source
2. Integration with popular data sources like Hadoop Hive or Google Analytics
3. Interactive visualization
4. MOOCS; tutorials available for online learning of the tool
5. API; can the services of the tool be embedded

Five tools were analyzed using this criteria:

1. Tableau
2. Power BI
3. Plotly
4. Gephi
5. Excel 2016

A summary of the results is displayed in Table 2.1 shows an example of a table.

Table 2.1. This is a table caption

	Open source	**Integration**	**Interactive**	**MOOCS**	**API**
Tableau	N	Y	Y	Y	Y
Power BI	N	Y	Y	Y	Y
Plotly	Y	N	Y	Y	Y
Gephi	Y	N	Y	Y	Y
Tableau	N	Y	Y	Y	Y

[24] completed a comprehensive survey of tools used in Big Data applications. The concept of Big Data has different dimensions, one model that attempts to describe these dimensions is widely known as 3Vs. The 3Vs stand for Volume, Velocity and Variety. Each of these dimensions poses its own unique challenges in the data management process.

The analysis and visualization of data was represented in a diagram that distinguishes between four different types of software solutions [24]. The diagram displayed in Figure 2.8 depicts the different types according to the Volume and Variety challenges in the 3Vs model.

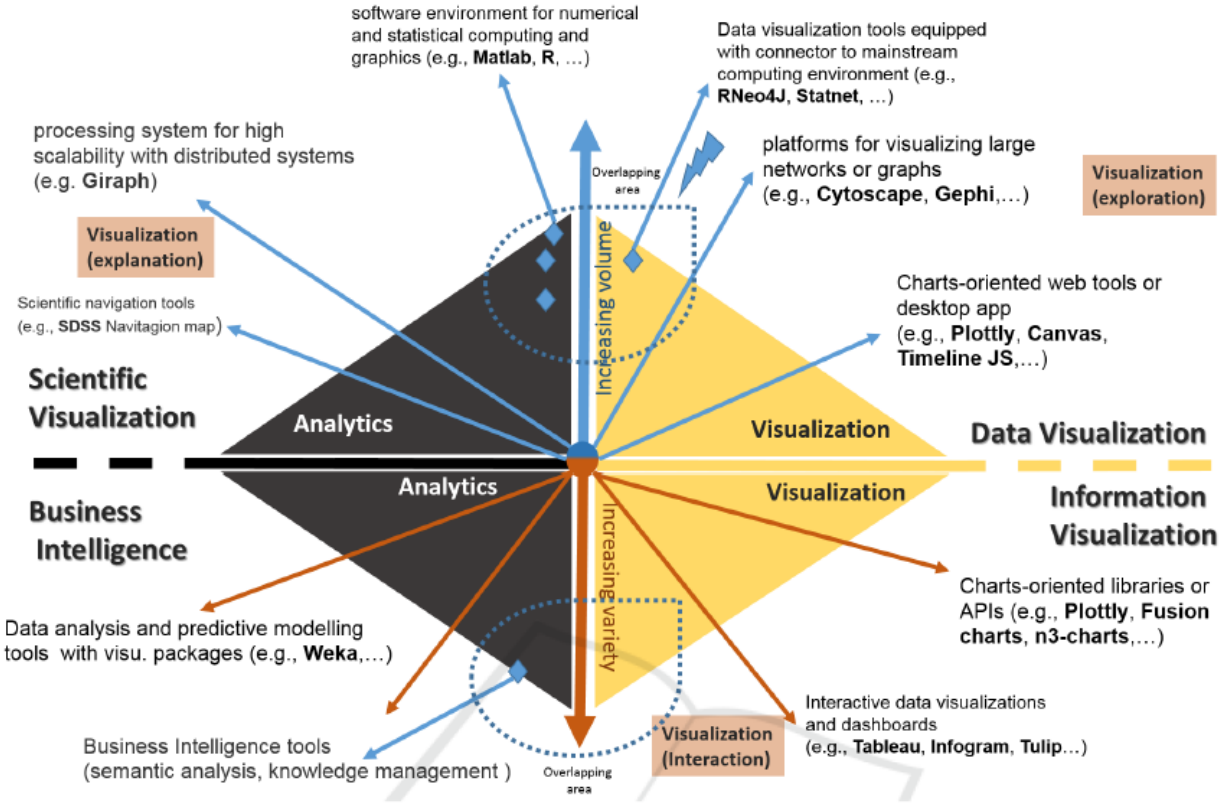

Figure 2.8. Big data visualization solutions

Differentiation is also done according to four main categories that describe the goals of visualization. The first goal would be the clear and effective communication of information to the user with the creation and study of the visual representation of data. The Information Visualization category is created to describe this goal. The second goal diverges from the first one in terms of spatial representation. If spatial representation is given to the visualization because of the intrinsic spatial layout of data like a flow simulation in a 3D space, then this can be seen as Scientific Visualization. These two categories

can be further divided according to the dimension of volume and variety of data. The four categories chosen by [24] divide software as:

1. business intelligent tools for data mining of heterogeneous data
2. scientific visualization tools focused that focus on specific spatial representations using larger data sets
3. data visualization tools used for visual exploration
4. information visualization tools used for interactive visual exploration of data

[24] analyzed 35 different software solutions in these four categories. What is interesting to note is that Plotly is well described in both the data and information visualization categories which implies that it does well at handling big data challenges with variety and volume.

[25] completed a study of information visualization theory and developed a time series analysis tool. The visualization tool developed by [25] was called Evert, it had the following goals:

1. Interactive visualization that enables exploration of time-series data
2. Seamless operation across all major operating systems
3. Python used as central programming language
4. Allow for expansion of the tool through plugins
5. User can efficiently apply the tool in their environments without the need to acquire new knowledge

Three existing visualization environments where also analyzed to determine if they could be used as the platform for this design. The environments investigated were JupyterLab, Glue and Lyra. JupyterLab was designed to be the future of the Jupyter project's notebook interface. The JupyterLab work environment consists of multiple windows of Python interpreters. Glue or GlueViz is a Python-based project designed for multidimensional data analysis. A GUI is launched when the program is activated through which a user would interact with the tool. Lyra is a tool based on the Vega platform that doesn't require any coding to generate data visualizations. Although all three of these tools have their own merits they were not chosen as the basis for the Evert project. The design of JupyterLab was too linear and at the time of the study could not allow for interactive visualizations accross interpreter

windows. Glue could not run on a web based server and could not allow for the customization of its interface. Lyra didn't allow for data manipulation and didn't support plug-ins.

One of the biggest challenges with modern visualization tools is the latency created by processing large data sets [23]. This can be addressed by using pre-computed data, paralyzing data processing and rendering and using predictive middelware.

During the development of Evert one of the key criteria in deciding what interface to use was the rendering of visualization displayed in a browser. Two main approaches are followed in this regards; Canvas and Scalable Vector Graphics. Canvas is a graphics API developed in HTML5 which generates pixels to represent the image. Scalable Vector Graphics (SVG) is an XML-based image format used to define two dimensional vector based graphics. [25] highlighted the flexibility of SVG elements and speed of rendering as the motivation for using this method in the final design. Basing the choice of rendering on SVG would solve the challenge of latency described by [23]. The use of D3 Javascript library as a standard for creating SVG visualizations in a web environment prompted [25] to investigate the following four plotting libraries:

1. Plotly.js
2. C3.js
3. DC.js
4. Mpld3

Plotly was chosen as the preferred visualization library due to its superior features such as pan, zoom and data selection interaction and ability to generate more than 2 Y-axis.

2.4 VISUALIZATION AND FDD

[1] described the exiting potential of combining data driven methods with process information to enhance the diagnosis step. The process information required is a qualitative model of the process topology which captures the logical pathways that information can flow. A standard called Computer Aided Engineering Exchange (CAEX) specifies an XML schema that can capture the process topology and feed it into a data driven tool.

[26] demonstrated a tool that combined plant topology information written in XML with results from a signal analysis tool called Plant-Wide Disturbance Analysis (PDA) to deliver enhanced isolation and diagnosis of root causes. The connectivity information provided by the plant topology allows the tool to verify or reject hypothesis about the propagation path of a fault. The tool described here was built in C# and contains a reasoning engine that can search for physical paths and determine root causes for measured plant-wide disturbances.

Considering qualitative analytics, as described in the section on FDD, a very useful way to visualize process connections is the signed directed graph (SDG) model. This model provides a useful way to visual cause-effect relations in continuous systems [27]. The process variables are denoted as nodes and the casual relations as directed arcs. The sign of the arc indicates the casual flow of the process. [10] used SDG models to aid in the logical reasoning process and display the results obtained from data driven methods like the LoopRank algorithm that was expanded on in earlier sections.

Process monitoring and fault detection of batch processes has received attention in recent years. The extensive use of batch processing in the production of high value products makes it a important to ensure that batch runs are completed with a high level of success. [28] developed a novel approach to process monitoring and fault detection in batch processes by utilizing a technique termed as time-explicit Kiviat diagrams. The core idea behind this technique is based on Multi-way Principle Component Analysis that facilitates batch-wise unfolding of the three dimensional (Time*Variables*Batches) batch data matrix into a two dimensional matrix that captures the variation across batches. The type of data required to warrant the use of this technique is very similar to the data generated by multiple time region analysis in the LoopRank analysis done by [10]. Each batch can be seen as a time region in the LoopRank analysis and the analysis can be seen as the completion of multiple batch runs.

SDG models can be obtained by flow sheets, empirical knowledge and mathematical models but in general they are more often derived from qualitative process knowledge and experience [27]. These models are of particular importance in plant-wide fault detection because they can help explain the fault propagation through a complex system. The directed graph description can also be represented in a matrix form as an adjacency matrix with each element containing a 1 or 0 denoting the direction of the interaction between two nodes.

[10] combined the application of the LoopRank algorithm and techniques to enhance casual relationships into a tool call FaultMap that has the ability to test various different methods used in FDD. FaultMap is predominantly developed in Python with Java dependencies to perform the transfer entropy estimations used in the interaction matrix. FaultMap makes use of various graphical models to display information ranging from basic trending of how importance scores vary across time regions to intricate SDG graphs displaying fault propagation paths.

CHAPTER 3 METHODS

3.1 CHAPTER OVERVIEW

This chapter will describe the process taken to develop a visual analytic system that will support the FDD process.

3.2 CONCEPTUAL DESIGN

The sources referenced in Chapter 2 describes various ways of how to approach FDD and what methods work best to structure the FDD process. A common idea or thread in most of these sources is the need for data manipulation or transformation. The key to obtaining an understanding that there is a problem and what is causing it is to view data in a certain context. This context needs to be created by interacting with the data.

Almost all of the literature sources highlight a need for a structured approach with logical steps that lead to a final goal. [4] describes a four step transformation process that classifies each step as a different data space; measurement, feature, decision and class. [15] proposes a sensemaking model with four key steps: Information gathering, representation of information in a model, development of insight through manipulation of the model and finally creation of knowledge through analysis. [17] describes a five step process that includes data selection, processing, transformation, mining and finally interpretation or evaluation. [19] proposes a structured approach to visual analytics that focuses of problem definition, visualization, analysis and reporting.

Based on the findings of all these sources it is proposed to also divide a new FDD tool into four distinct steps. The first and most important step would be the analysis of raw data. Before any complex transformations can be done it is important to understand what data is available and what the limitations are thereof. The human brain is very good at recognizing patterns without necessarily quantifying the

model that is used to identify the pattern. It is also important to understand the limitations imposed by the frequency of the data used. If the data is of a very low frequency it might not be feasible to identify faults that are caused by high frequency failures which is common when the failure is related to a instrument failure in industrial processes. There might also be differences in the frequency of different parameters in the data set. [10] also highlighted the importance of of a fast Fourier transform (FFT) to identify the active frequency regions, multiple frequency regions of interest could indicate that there are multiple disturbances present with different dynamics.

Process Information Management Systems (PIMS) and Laboratory Information Management Systems (LIMS) collect and store time series data in vastly different frequencies. For example a process data point might be available at minute intervals but a critical process specification like a viscosity analysis might only be available every eight hours. Furthermore LIMS data might also pose a challenge in terms of time shifts. The manual nature of how some lab samples are collected might cause a time shift between when the sample was taken and when the analysis was completed. In practice it can be very difficult to ensure that this time shift is consistent and also how to determine what the exact time shift is.

The raw data should not only be time series data but any other data that can be used to create context. As indicated by [1] and [26] any information regarding plant topology can provide valuable insight into the casual links between variables. This could potentially simplify any models developed and avoid nonsensical conclusions. Although the XML schema discussed by [1] provides an exciting prospect in itself there is no open source version of such a application currently available. For this reason this specific schema will not be incorporated into the tool under development but the idea can still be used. Any visual or diagram of the process that generated the time series data must be included in the first step of the FDD process. Even if the casual links are specified manually this still improves greatly on assuming all parameters are correlated to each other. Another source of additional data that is available on modern control systems is event data. Event data is a valuable source fo meta data that could include alarm data, changes made to control system parameters by operators or any other predefined alerts that would be logged based on the state of a variable, i.e. the changing state of a binary variable to indicate the state of a electrical motor.

Another aspect of the first step in the proposed process is to include the ability to incorporate note taking into the raw data analysis. This ensures that any information gathered by one user can be built

on by others to further the discovery process. As explained by [19] these notes can be in the form of findings and cues, where the findings can be as a result of calculations which might only be related to one of the following steps and cues can be anything that is useful, for example an observation of the casual link identified by using the process topology.

The final thread that must be woven into the entire process is the ability to interact with the data. As highlighted by [20] it is more important to have a highly interactive tool than a very accurate tool because this will encourage the engagement of the user. An engaged user will spend more time to guide the FDD process and provide valuable input into it.

The second step in the FDD process should be where the data is transformed in such a way as to highlight the most important information. This information would be the signs of a fault occurring or that a fault has resulted in some change in how the variables relate to each other. Data transformation and data mining is mentioned as key part of the study done by [17] into knowledge databases. The study by [4] calls this the feature space where certain features of the underlining data set are brought forward where they can be analyzed in more detail. [15] describes this as representing information in a model, a model is something that is created to help in the understanding of how variables in a complex system are related to each other.

In this second step the most customization must be allowed to the user to specify new methods of transforming data. There will always be new and creative ways to digest data and reduce dimensionality. For the purposes of this demonstration the tool called FaultMap that was developed by [10] will be used as the data transformation engine. This will demonstrate how to perform a FDD analysis of a data set by following the LoopRank algorithm to highlight faults and monitor how they develop over time.

The third step in the FDD process will be a space where the model or transformation developed in the second step can be modified or adjusted to test certain hypothesis. In this step we decide if the transformation process has identified a true fault. [4] aptly calls this the decision space. [15] describes this as development of insight through manipulation fo the model. Only when the user can see how the model changes can they truly start to internalize the effect of the model on the raw data and start to gain real understanding of what the effect of the transformation truly was. [19] describes this point as the analysis stage which ties in well with the idea of getting a better understanding of the end result.

The fourth and final step is called the classification space by [4], in this space knowledge is created through the analysis performed in the previous step. Interpretation and evaluation of the model is done and a final report is compiled of the findings of the study. This step can be very intricately intertwined with step three, but it is still important to distinguish this as the step where the final result of the FDD process is generated. This final result could then be documented and tested on a new data set or it could be used to improve any of the previous steps by modifying the transformation technique used in step two. Finally it could be used to allow the user to respond to fault conditions in the process and improve the process stability.

3.3 INTERFACE EXPLORATION

The method for choosing an interface has become a very interesting challenge in itself. In recent times the amount of tools developed for FDD analysis have grown exponentially. Commercially available tools are offered by almost every major player in the process control industry. Table 3.1 displays four such tools available that are specifically marketed to the process control industry.

Table 3.1. Commercial tool examples

Solution	Company	Key theme
Aspen Mtell®	Aspentech	Predict degradation and impending failure
Control Performance Monitor	Honeywell	Condition-based maintenance methodology
SAM GUARD	Precognize	Quickly and accurately identifies equipment failures
AMS	Emerson	Prediction and protection for production assets

Although it is out of the scope of this work to evaluate each of these tools in detail it is clear that a lot of work is being done to use a data driven approach as described by [6] to improve unit operations.

A bigger focus of this study is to define what is really important for someone using an FDD tool and how a tool should be created with the specific goals of the end user in mind. Very often end users experience frustration with existing tools because they are not flexible enough to cater for the unique needs of the individual. This is a common problem in any field but even more so in the process control industry due to the dynamic nature of processes and the infinite number of ways a complex

industrial system can be configured. Every person has his or her own unique perception of how to visualize a process control problem and conduct FDD. In the engineering field a person will very quickly developed their own tool using easily accessible software platforms like Microsoft Excel to accomplish a certain analysis task.

As highlighted by [20] inferior tools in terms of visualization are preferred if they are more interactive. This interactivity can also be linked to the idea of flexibility. A more flexible environment will allow the user more freedom to express his or her own unique approach to a problem. There is also a limit to the amount of flexibility that should be given. Without a pre-defined model and structure the tool will not support the user in solving a problem effectively. An example of structure requirement could be the cells in an spreadsheet, without the cells as a structure to let the user build on each successive calculation or output the user might lose track of what was done to get to the end goal.

This environment that the user of a system predominantly interacts with is also referred to as the front end of the system by [25]. [25] cited the ability to efficiently use the system without requiring new knowledge as one of the key goals of the Evert project. This goal will be analyzed in more detail because it is a key feature of FDD systems that often goes unnoticed but plays a very big role in designing how the system will work.

Three existing platforms were explored by [25] to determine their suitability for a FDD system specifically catered to time series data. These three tools were JupyterLab, Glue and Lyra. JupyterLab and Glue are both Python based platforms while Lyra is based on the Vega platform. The JupyterLab environment had key features that came very close to meeting the needs of Evert, it allowed for interactivity, a pluggable work environment and inline interactive graphs. One of the major drawbacks was the highly programming orientated working environment. It is very interesting to note that since the study of [25] has been completed in 2017 there have been multiple methods developed to get around this challenge which will be discussed in later sections.

A programming orientated environment could be seen as the most flexible environment. This would then allow the user to perform any FDD analysis but it would not become a FDD system if multiple users could not use it in similar ways to identify and diagnose faults. Some sort of structure is still required to guide the user in following the best thought patterns or series of steps described as a sensemaking model by [15].

One of the other key requirements of the Evert project as stated by [25] was to apply Python as the central programming language. A promising alternative platform that is also uses Python as the central programming language is called Plotly Dash. As indicated by [23], [24] and [25] Plotly is a very effective visualization tool. Dash is a point and click interface built on top of Flask, Plotly.js, React and React Js which allows for the creation of interactive dashboards using pure Python. Dash is open source and creates these dashboards in applications to run on a web browser.

The Dash environment was explored as a possible FDD environment and visual analytic system. The end user will certainly not need to code to use the interface and development could be done in Python. A demonstration application was built that attempted to incorporate the basic elements needed to import time series data and start to explore in the data space as defined by [4]. Dash provides very powerful tools that allow for a drag and drop interface to import new data and effortlessly visualize it in tables and figures. It was important that Dash had the functionality to allow the user to display and explore data in its raw tabular form inside a Dash Table. When the nature of the time series data, for example the sampling frequency or the time format, is not yet well understood this ensures that the users selects the correct processing techniques in the data manipulation steps.

Dash also had a very useful Store object that could be used to store data inside the web browser session after it was uploaded. This would make it easier to upload multiple data sets on the same application and navigate through data sets using tools like drop down menus. This also provided the flexibility to manipulate multiple data sets simultaneously, combine them or extract data from them. The Dash application provided all this flexibility in the session created on the users own web browser even if the application was hosted from a remote server. The data would not need to be transported back and forth between two locations which provided a performance benefit and also didn't need to be stored in a fixed location on the users work station.

The Dash Table and Figure objects were found to provide for a highly interactive environment. Based on the findings of [19] which advocates for the ability to incorporate user notes into a workflow an attempt was made to incorporate some sort of note taking functionality into the Dash application. A key element of making notes on time series trends would be the ability to indicate at what point in time the significant event occurred that that a note is referring to. A basic example of this would be a callout object on a spreadsheet graph like a speech bubble that would indicate to a point in time and also allow for text to describe the observation. This type of interactivity was not easily incorporated into a Dash

Figure object. The interactive elements already present on a Dash Figure like the ability to pan or zoom make it difficult to assign a fixed location to a note object as described above. Another option would be to only include the pointer or indicator element in the graph object and add the free text information in a separate input field next to the graph. Although it is possible to include a indicator object on a graph before rendering the functionality to dynamically add these objects after rendering is not yet available. This difference means the user cannot assign markers to a graph while the data is being explored which would be the ideal setup for a note taking interface. Based on these observations the note taking functionality is not yet at a maturity level required for this type of analysis environment.

The development of the demonstration application also allowed for experience with debugging in the development space. Dash automatically creates a new user session every time the source code is modified, this saves a lot of time when making changes or adding new features. Challenges identified were in the application environment when a dynamic object was not behaving as intended or experienced a error that affected its rendering. These errors were not severe enough to prevent the application from running but created pop-up error messages in the application linked to the javascript element on which the platform is built. These errors were difficult to troubleshoot and the Dash environment is not widely adopted yet which limits the amount of online troubleshooting forums to utilize. A method is available to print data to the command prompt window from where the application is opened. This does allow for understanding data structures and dynamic outputs.

The development experience when trying to build a very interactive environment has the potential to become too time consuming which might discourage users from developing their own add ins or make changes to the environment. Together with the lack in interactive note taking functionality indicates that Plotly Dash is not the ideal platform for the intended goal of this project.

The interactive note taking ability is something that is almost built into the way in which Jupyter notebooks are created. Each cell in a notebook could be used for executing code or creating a note. The challenge of creating indicator objects and text notes on a interactive graph would still not be solved but a portion of the graph data that is important for reference to the note could easily be extracted and placed in a cell adjacent or below a note. The interactive Graph objects developed by Plotly could still be used inside this environment. Recent developments have also provided exciting possibilities for creating dashboards from Jupyter notebooks. New solutions like Panel or Voila can easily convert a Jupyter notebook with its interactive elements into a dashboard application. This provides the option

of developing in a notebook environment which is very easy to debug or troubleshoot and then also display a dashboard environment for users that want to focus on only the rendered objects.

CHAPTER 4 RESULTS

4.1 CHAPTER OVERVIEW

CHAPTER 5 DISCUSSION

5.1 CHAPTER OVERVIEW

CHAPTER 6 CONCLUSION

6.1 EXAMPLE HEADING

REFERENCES

[1] N. F. Thornhill and A. Horch, "Advances and new directions in plant-wide disturbance detection and diagnosis," *Control Engineering Practice*, vol. 15, no. 10, pp. 1196–1206, 2007.

[2] A. Rahman and M. S. Choudhury, "New methods for detection of control loop interaction to prioritize control loop maintenance," in *International Conference on Electrical & Computer Engineering (ICECE 2010)*. IEEE, 2010, pp. 263–266.

[3] M. Farenzena and J. O. Trierweiler, "Looprank: A novel tool to evaluate loop connectivity," *IFAC Proceedings Volumes*, vol. 42, no. 11, pp. 964–969, 2009.

[4] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri, "A review of process fault detection and diagnosis: Part i: Quantitative model-based methods," *Computers & chemical engineering*, vol. 27, no. 3, pp. 293–311, 2003.

[5] Z. Gao, C. Cecati, and S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques—part i: Fault diagnosis with model-based and signal-based approaches," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3757–3767, 2015.

[6] C. Cecati, "A survey of fault diagnosis and fault-tolerant techniques—part ii: Fault diagnosis with knowledge-based and hybrid/active approaches," 2015.

[7] Q. Jiang, X. Yan, and B. Huang, "Review and perspectives of data-driven distributed monitoring for industrial plant-wide processes," *Industrial & Engineering Chemistry Research*, vol. 58, no. 29, pp. 12 899–12 912, 2019.

[8] K. Bryan and T. Leise, "The $25,000,000,000 eigenvector: The linear algebra behind google," *SIAM review*, vol. 48, no. 3, pp. 569–581, 2006.

[9] S. J. Streicher, C. Sandrock *et al.*, "Eigenvector analysis for the ranking of control loop importance," in *Computer Aided Chemical Engineering*. Elsevier, 2014, vol. 33, pp. 835–840.

[10] S. Streicher and C. Sandrock, "Plant wide fault and disturbance screening using combined transfer entropy and eigenvector centrality analysis," *arXiv preprint arXiv:1904.04035*, 2019.

[11] S. Few, *Now you see it: simple visualization techniques for quantitative analysis*, 2009, no. Sirsi) i9780970601988.

[12] A. C. Telea, *Data visualization: principles and practice*. CRC Press, 2014.

[13] D. Rees and R. S. Laramee, "A survey of information visualization books," in *Computer Graphics Forum*, vol. 38, no. 1. Wiley Online Library, 2019, pp. 610–646.

[14] P. Isenberg, F. Heimerl, S. Koch, T. Isenberg, P. Xu, C. D. Stolper, M. Sedlmair, J. Chen, h. t. c. d. r. t. b. s. Möller, and J. Stasko, "vispubdata. org: A metadata collection about ieee visualization (vis) publications," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 9, pp. 2199–2206, 2016.

[15] P. Pirolli and S. Card, "The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis," in *Proceedings of international conference on intelligence analysis*, vol. 5. McLean, VA, USA, 2005, pp. 2–4.

[16] G. Klein, B. Moon, and R. R. Hoffman, "Making sense of sensemaking 2: A macrocognitive model," *IEEE Intelligent systems*, vol. 21, no. 5, pp. 88–92, 2006.

[17] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," *AI magazine*, vol. 17, no. 3, pp. 37–37, 1996.

[18] D. Sacha, A. Stoffel, F. Stoffel, B. C. Kwon, G. Ellis, and D. A. Keim, "Knowledge generation model for visual analytics," *IEEE transactions on visualization and computer graphics*, vol. 20, no. 12, pp. 1604–1613, 2014.

[19] N. Mahyar, A. Sarvghad, and M. Tory, "A closer look at note taking in the co-located collaborative visual analytics process," in *2010 IEEE Symposium on Visual Analytics Science and Technology*. IEEE, 2010, pp. 171–178.

[20] P. Saraiya, C. North, V. Lam, and K. A. Duca, "An insight-based longitudinal study of visual analytics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 6, pp. 1511–1522, 2006.

[21] L. Barnard and M. Mertik, "Usability of visualization libraries for web browsers for use in scientific analysis," *International Journal of Computer Applications*, vol. 121, no. 1, 2015.

[22] S. A. Fahad and A. E. Yahya, "Big data visualization: Allotting by r and python with gui tools," in *2018 International Conference on Smart Computing and Electronic Enterprise (ICSCEE)*. IEEE, 2018, pp. 1–8.

[23] S. M. Ali, N. Gupta, G. K. Nayak, and R. K. Lenka, "Big data visualization: Tools and challenges," in *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*. IEEE, 2016, pp. 656–660.

[24] E. G. Caldarola and A. M. Rinaldi, "Big data visualization tools: A survey," in *Proceedings of the 6th International Conference on Data Science, Technology and Applications*. SCITEPRESS-Science and Technology Publications, Lda, 2017, pp. 296–305.

[25] N. Herbst, "Evert: Interactive web-based time-series analysis environment," 2017, unpublished thesis.

[26] S. Y. Yim, H. G. Ananthakumar, L. Benabbas, A. Horch, R. Drath, and N. F. Thornhill, "Using process topology in plant-wide control loop performance assessment," *Computers & Chemical Engineering*, vol. 31, no. 2, pp. 86–99, 2006.

[27] F. Yang, D. Xiao, and S. L. Shah, "Qualitative fault detection and hazard analysis based on signed directed graphs for large-scale complex systems," *Fault detection*, pp. 15–50, 2010.

[28] R. Wang, T. F. Edgar, M. Baldea, M. Nixon, W. Wojsznis, and R. Dunia, "A geometric method for batch data visualization, process monitoring and fault detection," *Journal of Process Control*, vol. 67, pp. 197–205, 2018.

[29] N. Elmqvist and P. Tsigas, "Animated visualization of causal relations through growing 2d geometry," *Information Visualization*, vol. 3, no. 3, pp. 154–172, 2004.

ADDENDUM A DERIVATION OF HIGHER ORDER MODELS

A.1 EXAMPLE HEADING

A.1.1 Subheading example

Filler text to test template. Filler text to test template. Filler text to