WEB BASED PROCESS MINING AND VISUALIZATION TOOL

By

ADITYA SHUKLA

A thesis submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Master of Science

Graduate Program in Electrical and Computer Engineering

Written under the direction of

Prof. Ivan Marsic

And approved by

_____

_____

_____

New Brunswick, New Jersey

January, 2017

ABSTRACT OF THE THESIS

Web Based Process Mining and Visualization Tool

By ADITYA SHUKLA

Thesis Director:

Prof. Ivan Marsic

Process mining has received huge attention by researchers in the last couple of decades. Business processes leave execution logs in various forms which can be examined and analyzed to formalize the process execution. Process mining techniques help analysts to extract knowledge from event logs and traces. Due to ever increasing amount of data and need of mobility and platform independence it is becoming harder to provide a visualization for analysis without overwhelming the user. Informal processes leave behind event logs that cannot be analyzed by plain old algorithms. The information and properties regarding a process highly depends upon the domain and hence a variety of conformance models needs to be analyzed. The existing process mining and visualization tools (Prom, EventFlow, RapidMiner) provide sophisticated visualization at the cost of mobility and platform independence. Therefore, in this thesis we propose a web based mobile and platform independent approach extending Trace Alignment [1][2] based approach to provide a feasible solution to the problem.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Process mining is a technique that allows analysis of various events and processes that occur as a part of essential activities in a business process. It aims at extracting information from the event logs and help analysts to find hidden patterns. The event and activity logs basically consists of data with multiple characteristics and dimensions. Often the data obtained is very huge due to the amount of activities and characteristics the process consists [3]. Analysis of such a huge data often poses a lot of limitations and requires a lot of sophisticated algorithms and analysis techniques to uncover the hidden meanings and significant content [4]. Graphical representation of processes is one of the best way to understand the pattern and deduce hidden important facts. Data visualization is increasingly gaining demand due to the act that the amount of data obtained from such processes is increasing day by day. Often the obtained raw data when plotted in the form of a graph using any visualization technique results in disoriented non uniform graphs, which are difficult to analyze and deduce the meaningful information. The reason for such results are mainly the length of traces in data, multi-dimensional nature and existence of multiple characteristics. There exist a lot of visualization tools and solutions that provide support to visualize process data [5][6]. These solutions often cater to small data traces and provide limited portability. Users are often required to install the application on their computer, customization and configurations are also essential to be performed before using the application. These solutions often eliminate some characteristics from data like time [2], similar activity occurrence in mutually exclusive traces etc., which results in loss of vital information. There are number of limitations that exists in these visualization solutions such as the length of data traces, consideration of multiple dimensions and characteristics, portability and platform independence etc. Our work in this document focuses on solving and eliminating

these limitations and aims at providing a feasible and viable solution to the users to obtain a better visualization and analysis of the process data. In order to develop a technique that ensures to provide a process visualization and analysis solution we have developed a visualization tool based on novel "Time-aware Trace Alignment" algorithm leveraging the web based design architecture. A detailed study and explanation about the "Time-aware Trace Alignment" can be found in [2].The current work is an extension of the previous work by adding portability and platform independence. Here we have overhauled the existing solution by leveraging the Web based architecture and REST services. In our work we take Trauma Resuscitation as our case study. The data obtained from trauma centers fits in all the assumptions made earlier. The length of obtained data is huge and it also comprises of multiple characteristics like time, duration , occurrence etc which are vital to the analysis. Health care institutions and hospitals largely require a sophisticated data visualization technique to monitor the process flow and workforce management. Researchers[3] in the past have proved that various medical professional need frequent context switching in their daily work. Also there are various tests and activities that are performed on a patient that take long duration. Research[2][3] shows that a lot of these tests and activities are mutually exclusive. A patient can go through multiple tests and activities at the same time without potentially affecting the mutual results. The data received as the result of these activities is huge and has multiple dimensions involved which are not possible to stream line without the help of a dedicated and sophisticated visualization tool.

1.1 Thesis Goals

As explained earlier, the existing data visualization and process trace analysis tools have a few limitations such as lack of aggregation of data, loss of data characteristics like time etc which have been solved by the novel "Time-aware Trace Alignment". The current work aims at providing portability, mobility and platform independence to the visualization system. This is achieved by developing a web based solution which could be hosted on a server and later be used just with a help of client typically a web browser. The figure 1.1 shows the

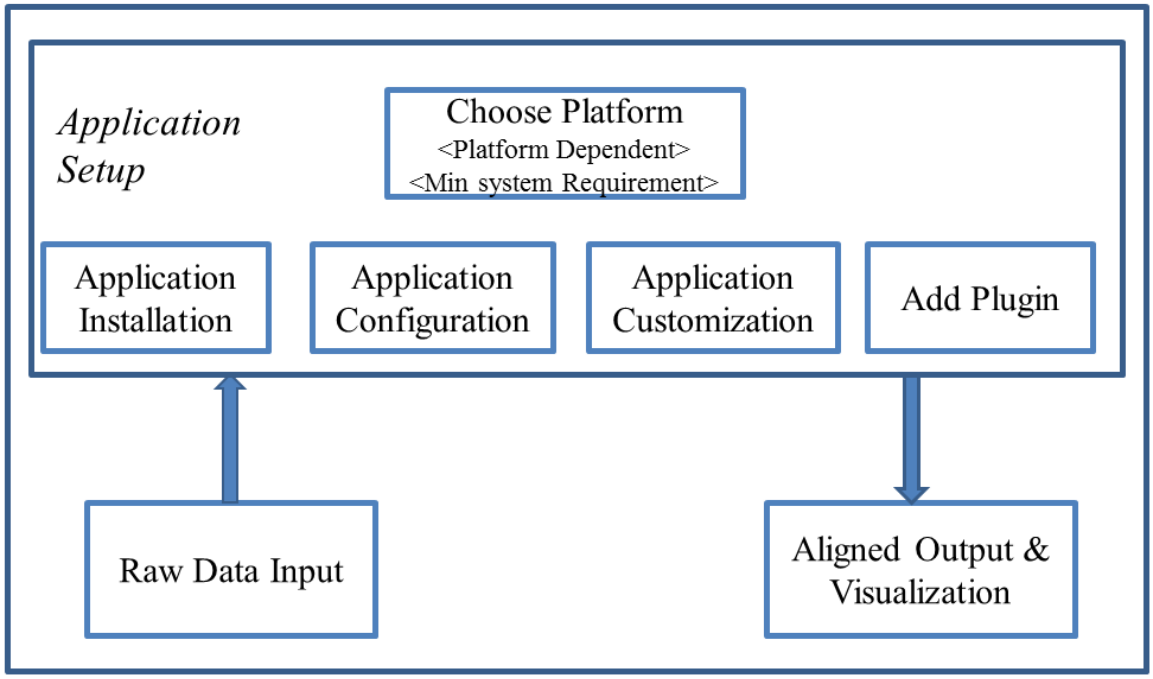

Figure 1.1: Typical Process Mining and Visualization Application Setup Process

typical work flow and sequence of actions an user has to go through in order to obtain the required process trace and visualization using existing application. There are multiple steps involved in setting up the application even before starting the actual usage. It also poses a limitation on being used on any platform and has a minimum specification required for the hosting system.

The main goal is achieved by developing a web based solution which could be hosted on a server and later be used just with a help of client typically a web browser. As shown in figure 1.2, this eliminates the initial configuration and setup steps required by the user and provides portability and accessibility across multiple platforms.

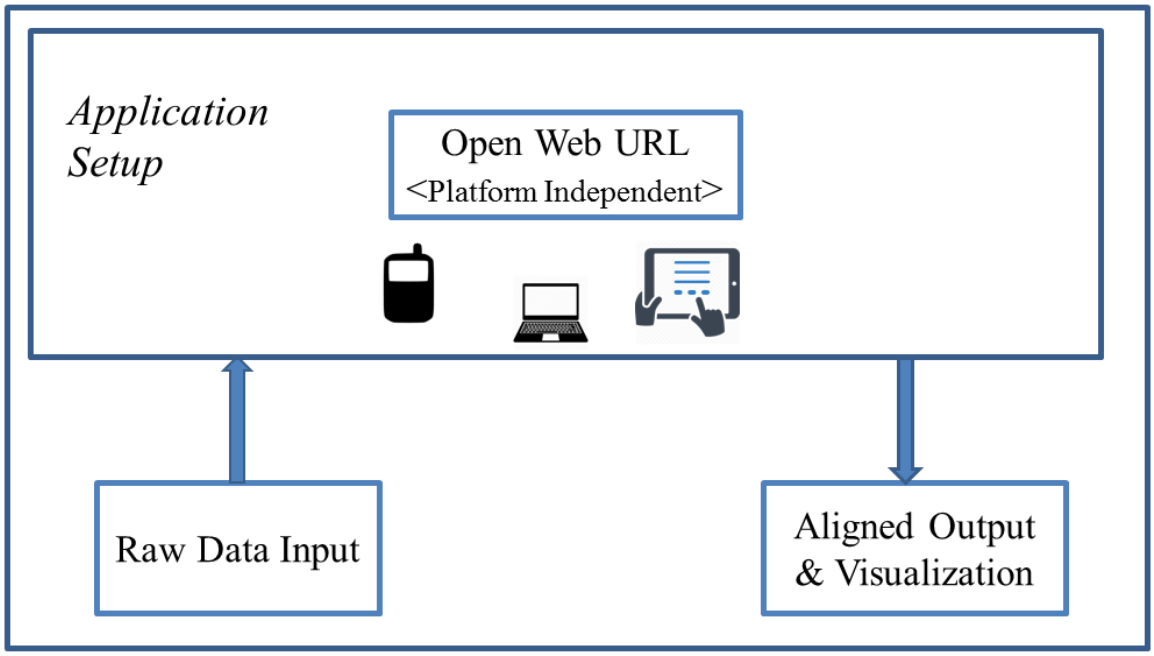

Figure 1.2: Web Based Process Mining and Visualization Application Setup Steps

Chapter 2

Related Work

Data, one of the most crucial and abundant aspects dominate a lot of industries in current scenario. With increasing amount of data, need to understand and analyze it has become imperative. Performance, Reliability and scalability at large revolve around the accurate and precise understanding of process traces obtained from various activities in all the domains. In recent times a lot of research has been conducted to develop and improve techniques to transform, analyze and visualize data obtained from processes. The aim behind all the efforts largely coincide on increasing the accuracy and minimizing human efforts required for the analysis. One of the best methods to analyze the data or process traces is to visualize it in multiple forms verifying based on the inherent characteristics and dimensions of data. Adding the interactive nature to these visualization improves the accuracy and decreases the amount of effort and confusion caused for human interpretation. Researchers have invested a lot of effort in providing a scale-able and reliable tool that could handle Process Trace mining and Visualization. ProM[9] framework is one of the widely used open source project that provides a platform to users to multiple algorithms and visualization for process mining. Continuous development to the project is made and plugins are introduced for the application to cater new demands for algorithms and visualizations. EventFlow[10], provides a sophisticated time-line on which the process events are displayed. The time line provides a unique method of aggregating the flow of events in the process. XESame[11] is a novel approach that describes a relation between the Data sources and Event logs. It provides a means to convert the data source to event logs based on the conversion rules. It is useful in cases where the Event information is not available but can be extracted from the Data source stored within a relational database. RapidMiner[12] provides implementation support for overall process mining projects. Data

mining, Machine learning and statistical operators are available in this framework to start developing and creating applications for specific domain. ProM also provides a Trace Alignment[1][2][3] plugin which helps in aligning the similar traces from a process. This provides an easy method of comparison between two process based on events and activities. On the same line CoCo[13] provides methods of comparing two processes at the same time. It can compare and call out the similarities and differences between two process groups. Despite the availability of sophisticated techniques and framework for interactive visualizations and Statistical analysis there is a lack of available solutions that provide both. Having an interactive visualization tool capable of providing statistical analysis is a must. The tools and frameworks mentioned above, at large don't provide a method of aggregation and sumarization of data. This poses significant limitations on these frameworks when large data sets are to be visualized. In addition, there is limited or no portability and platform independence for these solutions. Users are required to either install the application or add the application as a plugin into an existing application.

Chapter 3

Process Mining using Trace Aligned Visualization

In present time there is an abundance of information flowing around. Every formal or informal process leaves a gigantic amount of data. This data is present in multiple forms like event flow, logs, transactions etc. Due to the size of data log obtained from these processes it if often difficult to identify common patterns and anomalies within the events occurring throughout the process [3]. A simple plot of all the events occurring in the process would help analysts to visualize events, but in order to draw conclusions and compare multiple processes a more sophisticated visualization is required.
Process mining is all about finding the hidden treasure of information within the events. The information can be based on comparison of multiple processes or it can also point out similarities between the processes. To understand more about importance of Trace Alignment in visualization, consider a visualization shown in figures 3.1. In the figure, each row shows different process and each column depicts various events logged during the process execution. The visualization is drawn with Time on x-axis, increasing as we go from left to right. Events are drawn based on their logical time rather than real time. Each event is marked with a different color within a process. Processes may or may not have same events, but for same events the color coding is kept intact to provide ease of readability.
The resulting visualization in the figure 3.1 provides minimal information. It just provides with logical occurrence of events. It is impossible to derive similarities between multiple processes; hence comparison of process is not possible. In order to draw a more meaningful visualization that would provide ease of readability and would allow analysts to draw conclusions can be obtained by using Trace alignment over the event logs obtained from multiple processes. Figure 3.2 shows an updated visualization of same process trace (data), but the only difference is the the visualization is now trace aligned.

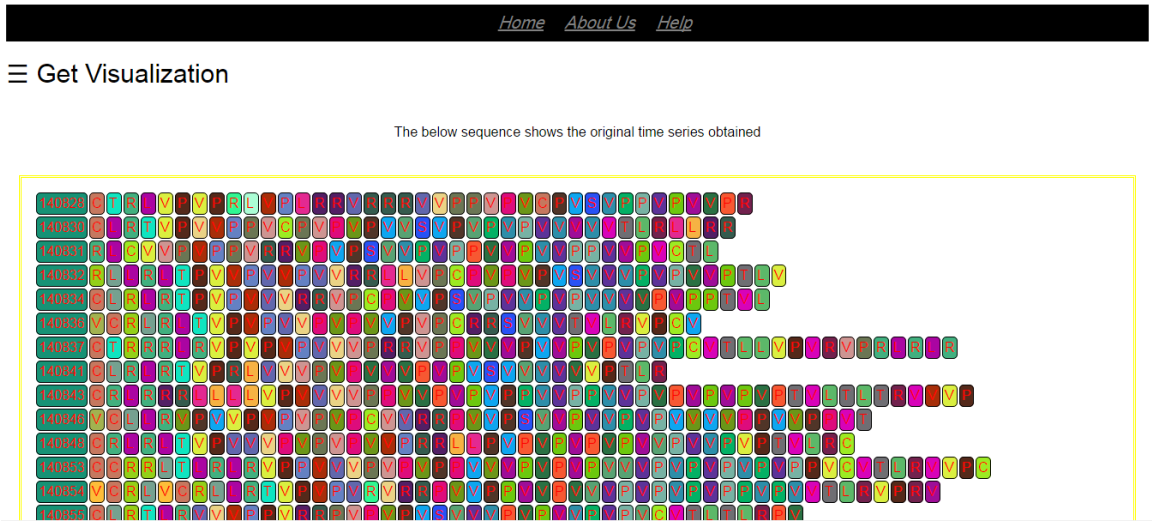

Figure 3.1: Visualization showing all events in one glance

The visualization drawn uses the same dimensions on both axes as previous. The only difference is now the process events are aligned with respect to each other. The resulting visualization is not only easy to be read, but also allows analysts to analyze various processes and compare them in depth. A small analysis that can be drawn from the visualization is that the processes that have similar events occurring at the same time are considered to be equal. Also, if a process has rules and directs events to follow set of principles, can easily be seen in such visualization. This will allow analysts to find events that occur in series (one after the other) or in parallel (no logical occurrence schedule).

Trace aligned visualization is capable of providing a lot of information about process execution. Some of the important conclusions that analysts can draw are

1. Expected or most likely behavior of a process: Trace aligned visualization provides a consensus sequence; the consensus sequence is the parameter which depicts maximum conformation of processes in the visualization. Following the consensus sequence, analysts can obtain the most probable behavior of the process.

2. Listing the deviations: With the help of trace alignment, it is very easy to figure out the deviations within multiple processes.

3. Listing the similarities: Apart from the consensus sequence which provides most likely

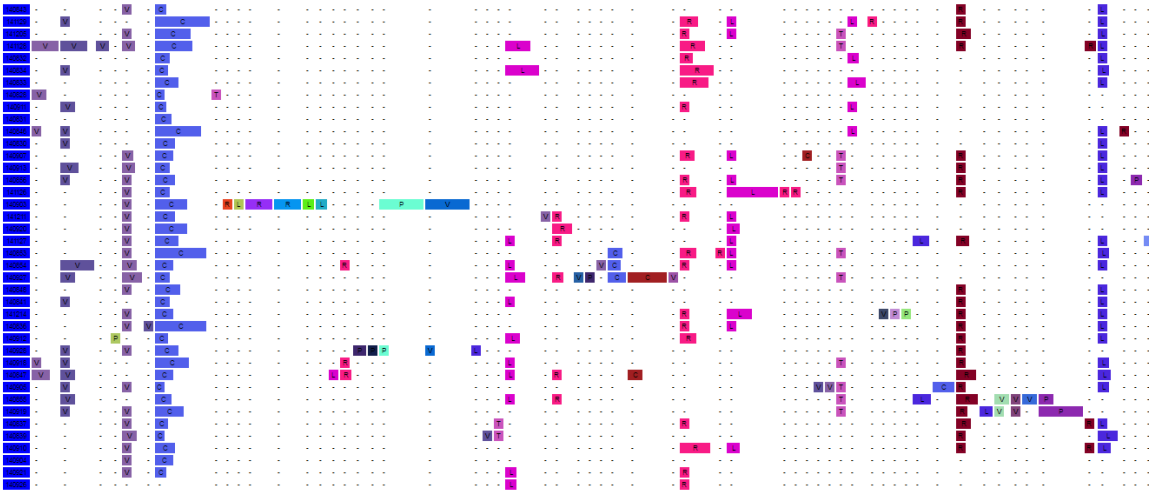

Figure 3.2: Trace Aligned visualization of events

process behavior, Trace alignment can also provide similarities found within chunks of processes.

4. Common Patterns: With in an event log, Trace alignment allows analysts to skim through and obtain most common patterns with in the log.

Chapter 4

System Design Overview

Data visualization for long sequences primarily focuses on visualizing data considering various characteristics of the obtained data. The system emphasizes on visualizations that enables user to identify critical characteristics like time duration, relative occurrence and process alignment in the provided data. Such a system finds extensive application in any data extensive domain. Example of such a domain would be an Emergency room at a trauma center, where a patient goes through various tests and examinations in an EMT room. Doctors perform multiple tests in parallel on the patient to get the accurate condition and suggest a solution for the patient's current condition. Due to high accuracy required in this area it is imperative that the data obtained from such domains are huge and have multiple characteristics. Analysis of such data sequences can be overwhelming unless supported by a sophisticated visualization tool. This system aims at providing an easy and effective way to analyze such data and help streamlining the whole process. The system enables user to view the same data in multiple ways based on the characteristics, for instance user can simply view the raw data or view the data sequences accounting for the time duration. It also features a way in which a user can view all the activities aligned on a time line with the neighboring sequences. To achieve such visualization it is important for the system to be able to extract all the characteristics of data. Once the characteristics are available a number of visualizations are displayed just on a click of a button. The application is platform independent and works on any browser which increases the usability and ease of use. It also eliminates the requirement of application installation and setup which might be cumbersome in many cases. The features like platform independence, multiple visualization screens, and characteristics extraction are few major aspects that make this systems one of its kind and increase its

audience[14]. All these features are achieved by leveraging the latest available technology solutions. To provide the ease of use and platform independence the system utilizes a state of art Service Oriented Architecture working over the HTTP protocol. The whole system in context is mainly data intensive and performance is one of the major aspects, in order to improve the performance of the system, JSON is used for transmitting data over HTTP. Various visualization and characteristic extraction is handled by using D3[15], a cutting edge JavaScript framework which seamlessly works with data intensive applications. The system is available as a web based application which allows user to use it without the hassle of installation and configuration. The system also has a scope to advertise the API which could be used by researchers and programmers in their application to handle visualization.

4.1 Design Decision

The design of the system is a major task, but a task more important than that is deciding how to develop a system and what technologies to be used in the development. These decisions are taken based on what a system is desired to do. In depth analysis of requirements are done in order to understand the actual features of the system. In the current era of technology, where multiple technology options are available it is often difficult to decide what's and where to use. Starting from the programming language, mark-up languages, web technology options, communication protocols, architecture style etc. are few of the examples which are intricately taken care of in the design phase of the development. In this section we will call out various options available to tackle each problem associated with the system design. We will also explain in depth reasons for the choices made in designing this system by explaining the pros and cons of each optional technology at our disposal. We will also draw a conclusion that why it was best to use the technologies we have used in the development of this system. The developed visualization system is essentially a Full Stack application[19]. These are essentially web based applications which have a Front end, a Middle tier and a Database as part of the system. Our system falls into such category as it caters a web based application to solve the issue. As part of the system we have I) Front end often called as the User Interface built in HTML and JavaScript, ii) Middle tier often called as Business logic build in Java and has servlets and iii) Database which is a very thin layer in our system. In

the following section we will describe various requirements for each layer along with the available and chosen technology option to handle the requirements and build the system.

4.1.1 User Interface or the Front End

User interface is essentially the only layer of the system that is exposed to the end user. A system essentially has just this layer as the mode to cater human computer interaction. While developing this layer the emphasis was given on the end user's prospective and the domain rather than the system and developer. To start with making the decisions it is always better to draw basic essential requirement outline that needs to be focused on[20]. We emphasized on the following requirements to draw upon the conclusion of choosing a particular technology used to develop the UI:

1. Ease of use and navigation
2. Minimizing the displayed data
3. Fast turn-around
4. Flexibility and Control
5. Dynamic nature of the rendered pages

The requirements listed above covers the basic ground of how and what should be included in the UI. All of the requirements are essential to be met in the designed system.

Building Web Pages To build web pages, it is very obvious to use HTML. But the plain old html alone will not provide the all of the above listed requirements. It is very important to include dynamic nature, animations, navigation, styles and colors. HTML is easy to develop and also often takes less time to load on the browser, due to the fact it requires no processing from the browser side. But making our application a few micro-seconds faster at the cost of above requirement is not worthwhile. So a vanilla HTML is not suitable for the development in this case. We decided to add dynamic nature and other important features in the web pages. Another available option that is worth considering is using DHTML, but due to rare use of it and advent of HTML 5 poses a lot of

questions on using DHTML. DHTML is also a lot slower to load on the browser side and requires a lot of processing from the browser side further slowing down the application. Considering all the options we decided to use HTML with a support of other technologies to provide us the aspects on which HTML was missing. The HTML pages in the system are supported with JavaScript to add dynamic nature and animations along with CSS to provide better look and feel of the pages. There is very less or no significant slowdown in the performance as Javascript requires minimal browser side processing if developed in a modular way.

Adding dynamic nature to the Web pages Dynamic web pages are developed by inclusion of Javascript in the front end technologies[18]. This provides web pages with the quality to react on any change occurring on the web-page dynamically. It also allows to add or remove parts on the web-page as and when required. With help of JavaScript various animations such as hiding of navigation panel could easily be handled.

Inclusion of Style and Color in the Web Pages Cascading Style Sheets often known as CSS is one of the basic and easiest technologies to be used to provide style and color to HTML web pages[17]. It also allows design and layout of the pages to be kept uniform across the system. One of the added advantages of using css is that it enables the pages to render on to different display devices with variety of display sizes and formats. This provides huge portability and ease of use of the system.

Choosing the Javascript library It is very important to have the dynamic nature in the system. Although every possible scenario can be handled by using plain old Javascript engine yet there are specialized frameworks available to be used. A lot of Javascript frameworks are available as an open source system to be utilized in various applications. The sole purpose of these specialized frameworks is to provide faster and economical web development. Using these frameworks allows developer to use and add fabulous features to the application without having the need to develop the code from scratch. Choosing the best fit framework is a challenge. For our system we examined a few frameworks available which claim to provide easy development solutions. The major aspect of our system to be

kept in mind while choosing the best framework is that our system is Data Intensive. The choice revolved around the fact that the framework should be capable of modifying and changing the data very efficiently. As a choice we closely analyzed multiple frameworks and in the following section a zest of the pros and cons of each framework and our decision is explained.

1. React.js: It is one of the most popular JavaScript frameworks available in current times. Applications like Instagram, Netflix use this framework[21]. The key feature here is the intelligent response to the change in data occurred in the back-end. It is extremely responsive to any kind of change in the incoming data and refreshes the web page automatically. It is capable of handling data extensive applications, Instagram being an evident example of it. But it is not exactly what our system requires. We have static data in our system which is prerecorded and then fed into the system. Also less expertise of this framework in terms of graphical representation of data poses a limitation on its use in our system.

2. Highcharts.js: It is one of the most popular frameworks used by developer in applications which require huge amount of charts and graphs[22]. It is the most prepackaged charting framework available in current times. It can be used to develop various charts and graphs without dealing with even manipulating single HTML DOM attribute. The prepackaged nature of the framework increases the productivity and decreases the development load but at the same time also poses serious limitations with the customization of charts and graphs. It is often difficult to get an off design implementation with such prepackaged frameworks. As our system needs to cater to multiple domains and visualization needs, it might not be the best decision to minimize the capability of customization.

3. 3) D3.js: Data Driven Documents, referred to as D3.js is an efficient JavaScript framework to produce interactive and dynamic visualizations[23]. It deals with the low level HTML DOM attributes to tackle the data. The low level design implementation of this framework gives developer a lot of control over the resulting visualization. It requires a bit of work to initialize the visualization setup but provides a lot of scope

for customization. It also has very convenient methods to read the data from csv files, which in our case is the thin database layer. Few features of D3.js that makes it most favorable framework for our system are:

- Binding: It efficiently binds the data to the HTML DOM elements. For our system this is the most favorable feature. It will enable us to embed the data into HTML pages and display them seamlessly
- Loading: It has various methods which can easily read data from csv files. This provides an added advantage for our system development as the thin data layer of our system is basically csv file. The Ad-hoc development of the system layers will be able to utilize this feature by using csv files as direct database without having the middle tier in place.
- Transformation: It also has a feature that enable data transformation that can be customized based on the on requirements. This feature is very important and can be leveraged to create a balance in the amount of data displayed on the pages. This will provide us with mechanism to maintain and check the overwhelming nature of our data intensive system.
- Transition: This feature provides developer with a tool to have a framework supported animation and alteration to the displayed data as per the user interaction. This will be of high impact in future where our system will start providing such visualization solutions.

Considering all the broad and minute observations from different frameworks, it is considerably evident to use D3.js as our choice of framework in the development.

4.1.2 Business Layer or the Middle Tier

Business layer of the middle tier of the system is responsible for manipulation and pruning of data. It is essentially the layer which performs algorithmic transformation of raw data. The meaningful information in the raw data is pruned and formatted in this layer of the system. Huge chunks of incoming data are subjected to multiple passes of pruning, alignments and reordering. This layer has the maximum workload throughout the system due to the fact

that it has to process each line of incoming data in multiple ways. To start with making the decisions it is always better to draw basic essential requirement outlines that need to be focused on. We emphasized on the following requirements to draw upon the conclusion of choosing a particular technology used to develop the middle tier.

1. Robust and Error tolerant
2. One - one mapping of functionality and modules
3. Fast turn-around
4. Flexibility and Control
5. Efficient memory usage
6. Re-usability and scope for customization

The requirements listed above covers the basic ground of how and what should be included in the middle tier. All of the requirements are essential to be met in the designed system.

Functional Modules These modules are basically the code blocks which essentially provide solution to one of the visualization tasks in hand. It is important to bi-bifurcate various visualizations into multiple modules. These developed modules can then be treated as a black box which will interface to any other domain or classification without requiring much of changing. This provides flexibility and scope for customization. We have used Java as the basic development language for our system. The widespread support and recognition of java makes the development easy and robust on multiple platforms.

Algorithmic Modules These modules are same as functional modules but find usage in multiple visualizations. These are basically at core some interfaces with contracts. These are built in Java due to multiple libraries available in Java that support the development to a very large extent. Prepackaged Java libraries starting from algorithms to data structures are available to be used. Which not only increase the efficiency of development process but also take care and maximize the performance of these algorithms.

Service Oriented Development In order to maximize the efficiency and re-usability service oriented development is leveraged. In this type of development we aimed at providing the result to the User interface over communication protocols like HTTP. The results will be calculated and formed on the high speed servers on which the application is hosted. If we were to process these algorithms on the client side using Javascript it would pose some significant limitations on the client's specification. This would also hamper the performance and efficient memory usage of the system.

Data Interchange Schema It is the method or the notation utilized by a system to transmit data from the middle tier to the client UI. It is very important to choose the best fit notation as the communication takes place over the network. A wrong decision would just eat up the network bandwidth and also cause severe performance issues. It also might cause the client to time-out while waiting for a response. The possible format options are multiple, here we will discuss about the two most popular and widely used data interchange schema in the current scenario.

1. i) XML: Extensible Markup language often called as XML is a method of interchanging data in web based applications. It has various advantages of its own starting with its ability to include meta-data. It is more verbose and is easy to interpret. In our system, the sole purpose of the data interchange module is to carry and transmit data to the client efficiently. XML proves to be a heavy solution in terms of bytes of data transferred from same amount of data. It is because of various reasons like its verbose nature, inclusion of attributes that are not required. In essence XML is a fat method with a lot of attributes and overhead.

2. ii) JSON: Javascript Object Notation often known as JSON, is essentially a fat free option for data interchange for web applications. Its works on basic object notations and can have multiple objects in form of an array. Due to the fact that it has less attributes for the same amount of data it proves to be faster compared to XML. It uses name-value pairs to map data. The size of the files formed on the same amount of data using JSON is much smaller than compared to XML. Although it is less verbose as compared to XML but the significant amount of performance gain is priority in

our system. JSON also uses comparatively less resources in order to be processed compared to XML. JSON also finds seamless compatibility with Javascript which is the main part of our systems front-end.

Considering all the broad and minute observations from both XML and JSON, it is considerably evident to use JSON as our choice of Data Interchange Schema.

Chapter 5

System Architecture

The system is designed keeping in mind various use cases and requirements that it should be capable of providing a fast and reliable data visualization solution. The solution provided here is mainly a Web based application[14]. It is a three tiered architecture that supports the associated use cases as well as provides a robust solution by leveraging the top technology stack available currently.

The application is mainly divide into three parts i.e. I) User Services or Presentation Layer , II) Business Services or Business Logic Layer and III) Data Services or the Data Layer. Each layer is isolated from the other in its working and communicates with each other over HTTP and TCP/IP protocol and provides the required services to other layers. The system architecture here is explained based on above layers. We describe the role of each layer and also explain and call out various technologies used in there to provide a fast and reliable solution to the problem.
As shown in figure 5.1, this system is divide into 3 layers, each layer performs specific task related to the systematic division of the system based on area of expertise. The layers basically work as a consumer and producer manner. Each layer is responsible for generating a specific response based on the request which is then consumed by the requesting layer. The layers mainly communicate over HTTP and TCP/IP protocol. The responses from the Business layer to the Presentation layer are transmitted in JSON encoded format. As far as the Data Layer and Business layer communication is in concern, the data is read by FileStream adapter module and communicated over HTTP protocol. The data file is mounted on network drive or server using SNMP protocol which later is used as a database and read by opening a FileStream to the mounted file.

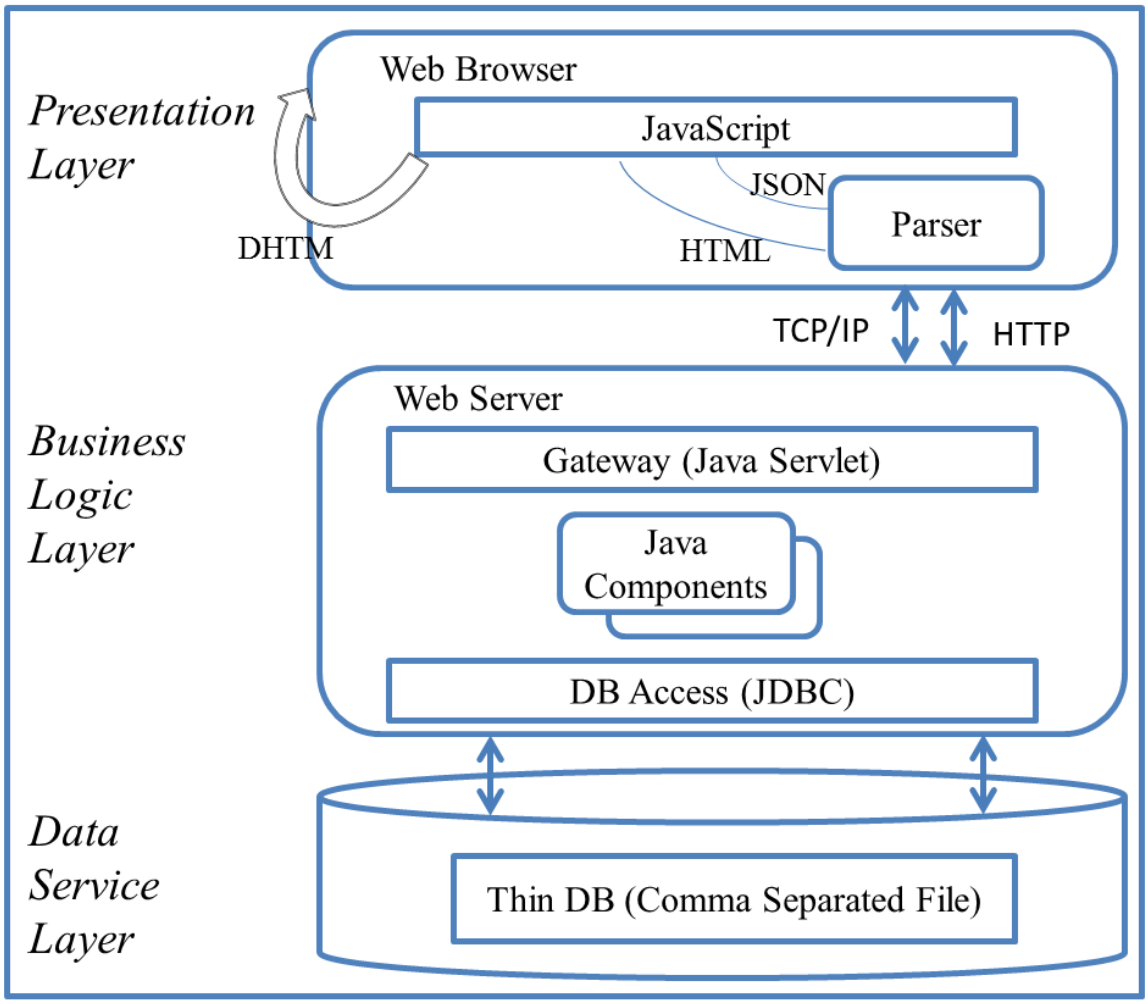

Figure 5.1: Web Based Process Mining and Visualization System Architecture

5.1 User Services or the Presentation Layer

Presentation layer is one the most important part of the application. This layer is responsible to cater human computer interaction services for the system. This layer is often referred to as the Front End of an application. It enables user to use the application and at the same time application can also communicate the required output to the user using this layer. In other words this is the user interface layer of the application. Another reason for the critical nature of this layer can be attributed to the fact that this layer is responsible to produce the required visualization and interact with the user to change the visualization as required. Now as the soul of this application is visualization of data, the importance of this layer is evident.

5.1.1 Design

Presentation layer is the layer responsible to cater all the Human Computer Interaction required for functioning of the system. An in depth user experience research is done to provide best user experience. User Interface of any system is one of the most critical parts of the system. The sole reason for its critical nature is that it has responsibility to cater all the features a system has to offer and at the same time should be very evident and easy to understand[16]. This layer handles all possible visualization the system has to offer and also caters to various inputs and outputs required. The system is aimed to provide services to non-technical business users as well, which further imposes more responsibilities and constraint to the design. The design of this layer was outlined while keeping a few points in consideration. Most of these points directly look upon the issues discussed above.

1. The UI should be easy to understand and self-explanatory
2. The Visualization must not look overwhelming to annoy the user
3. Navigation between pages and visualization screens should make obvious sense

The above guidelines[16] gave a foundation stone towards building of the user interface layer. As the system is not a small one but has multiple features and solutions, the UI

is a mute page application built in HTML. The pages are divided based on the specific functionality of the system they cater. Layout of all the pages is kept same to reflect the sense of connectivity between other pages. Multiple navigation bars are available to navigate through the system based of the specific navigation the user is looking for. As an answer to the requirement of not overwhelming the user we have used a technique to display the data best explained as "Data as Requested", which shows chunks of data as and when requested by the user. These features are explained in depth in the following section.

User Interface Layout The layout of pages is very simple yet provides full feature coverage. The simplicity given to the layout is very favorable towards the ease of navigation and understating of the system for first time users. All the pages in the system necessarily share a common layout which is best to preserve sense of connectivity yet provide with standalone functionality and existence of each page. This also proves to be favorable for users as each page necessarily exhibits the same attributes and layout but caters different functionality of the system.

Functionality Pages The UI has been divided into multiple pages based the functionality and visualization type each page handles. There is a separate page for every type of visualization we handle; this helped us to keep check on the amount of data each page handles. Minimizing the amount of data on each page will never have an overwhelming effect on the user. With this we have also achieved a hidden simplicity in our design which would prove the ease of use of the system for business and non-technical users. Future improvements and functionality addition would necessarily become simplistic. The new functionality page would be in essence a standalone HTML page added to the user interface without affecting any of the existing pages in general. This gives huge advantage in applications like this which has huge customization window based on the domain.

Effective Split of Navigation We have also provided with an easy navigation bar on the left and top which caters different purpose. As shown in figure 5.2 and 5.3, the navigation bar on left helps user to navigate to various available visualizations while the one on top helps user to understand the whole system and gives information about various

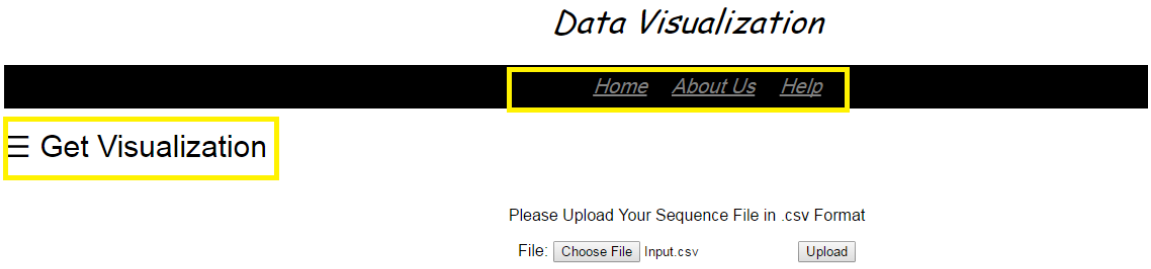

Figure 5.2: Split Navigation to Improve User Experience (1)

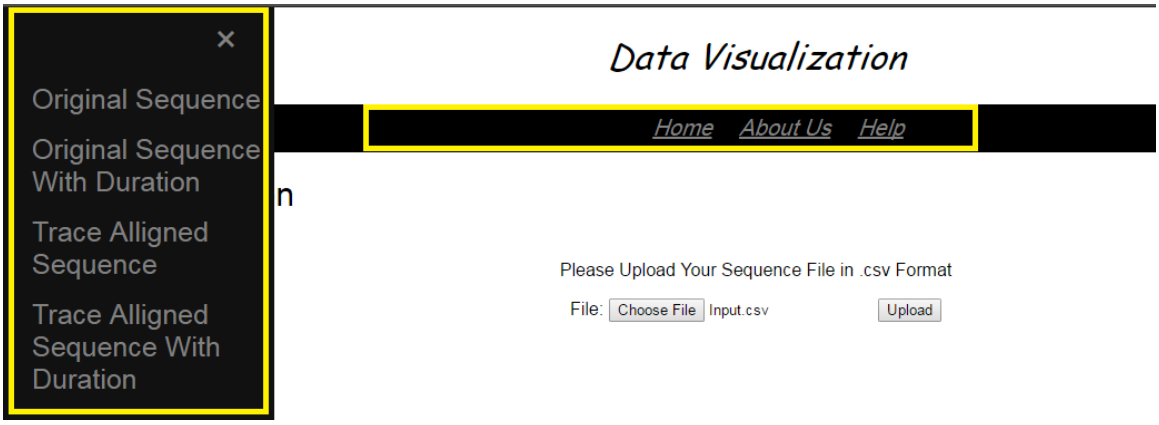

Figure 5.3: Split Navigation to Improve User Experience (2)

options available within the system. A troubleshooting insight is also linked to the top navigation bar. Dividing the navigation bars into two different sets proves to provide much lucid and easy navigation throughout the system. Anytime a user can easily access help or change the type of visualization he is viewing. This also helps in maintaining the dynamic nature if the pages.

Data As Requested An effective measure to check the amount of data displayed at a time on the page is to display the data based on request. In our system, the visualization grids display minimal data that is necessary for understanding the visualization. If user wants to know about the details, a mouse click on the required region opens up a description box. This description box in figure 5.4 provides details about the data in the visualization. It gives out the Activity, Sequence, and Time Duration etc. which later helps the user to draw a meaningful conclusion from the visualization.

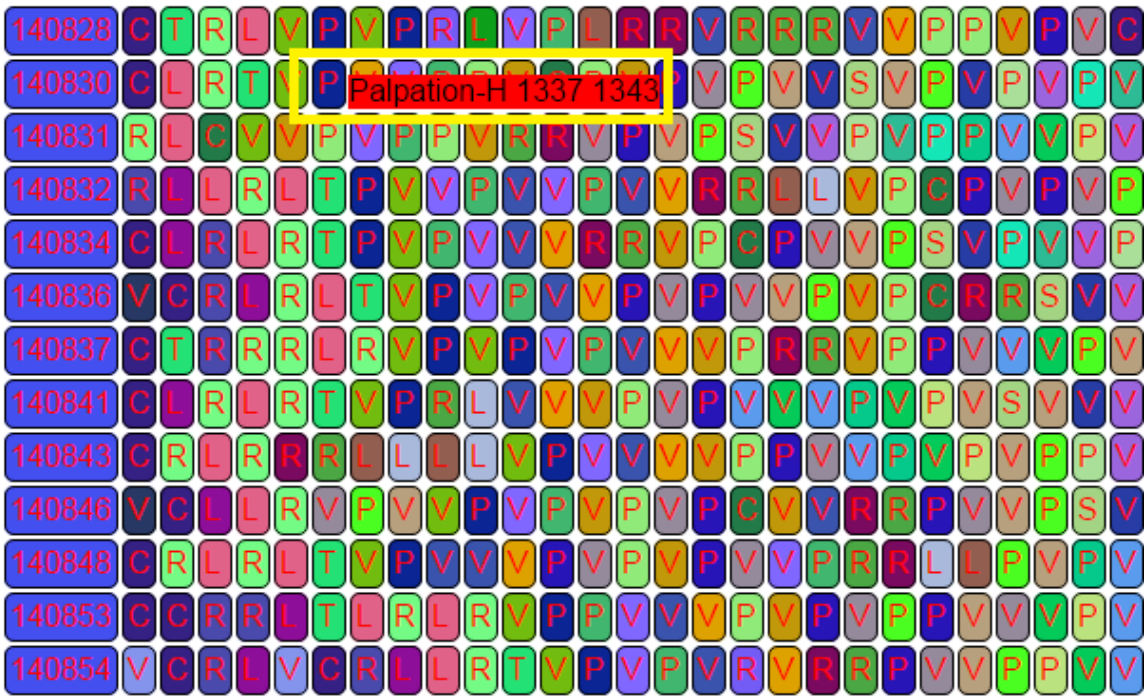

Figure 5.4: Data As Requested On Mouse Click Event

Stateless Pages All the pages in the system are stateless and do not call themselves at any point of time. There are no request parameters associated with the pages. We do not maintain a session which essentially provides us felicity of navigation at the cost of tracking each user activity which in our case is imperative. To provide all the ease of navigation and providing simply what the user needs is achieved by making all the pages stateless. A user can any time just from or to a desired page without caring about the traced. We have carefully designed each page such that the user will not be required to follow particular steps in order to get to the results. For instance a user on basic visualization page can jump to the time duration visualization or any other visualization page of his choice.

Color Coded Sequences Due to the data intensive nature o this application it is possible that a user might get overwhelmed which might lead to unfavorable results. Visualization of multiple unknown and random sequences is very challenging unless provided with some standards and specific regulations. To tackle this issue we have decided to go ahead with color coded sequences shown in figure 5.4. Color coded sequences essentially provide same data but every data chuck is color coded which differs with the other based on the

activity in each sequence. As the aim of the system is to be versatile and provide solutions to multiple domains it is impractical to pre-code the colors for all possible activities. As a result we decide the color schema for each page at the time when user requests the data to be displayed. This on the fly color coding mechanism provides us the flexibility and ability to cater to numerous activities. It also provides ease of domain transition with no customization requirement for color coding as it is automatically taken care of by the data layer.

Design Decisions Well designed system exhibits uniqueness in each part yet provides a sense of belonging-ness and layouts the whole system a single entity. A lot of emphasis was given on the design of the whole system. A minute change in the base layout has ripple effect on each part of the system.

5.1.2 Technology

The User Interface of a system is the only visible part of the system. An UI lacking in features and functionality in the current scenario creates a tiring environment for the end user. It not only degrades the ability of user to efficiently use the system but also causes more human errors and system performance issues. It is very vital to the whole system to build an interface that is fast, efficient and obvious to be used. In order to provide a bundle of feature loaded interface it is imperative to leverage multiple technologies available in current time. In the previous sections we called out the basic requirements of the UI which our system should have. In this section we will describe the technologies we have used to ensure the availability and fulfillment of those requirements.

Web Pages and UI Layout This is the most basic part of any user interface. It has to be simplistic to result better performance and lucrative enough to suite the eyes of a user. To achieve this we have used HTML which keeps it simple and due to no or less requirement for browser side processing it renders on the UI faster.

Improving the look and accessibility To improve the look of the developed HTML pages we have implemented styling and designing techniques. By the use of Cascading style

sheets (CSS) [17], the developed HTML pages render a perfect and similar layout throughout the system. It also enables in creating animations where ever required on the pages.

Dynamic Capability If the pages are static, there is no way to limit data displayed on the page. The concept of displaying data when asked for by the user will not be possible to integrate in static pages. To tackle this issue the pages are made dynamic with the use of JavaScript [18]. We use a JavaScript framework known as D3.js to achieve the goals of providing dynamic nature to the web pages. This allows us to limit the amount of data shown and release more data as and when the user interacts with it.

Minimizing cluttering of Data To maintain the readability of web pages, it is important to display minimal data. We implemented various methods by leveraging the JavaScript frameworks to minimize the amount of data displayed on the web pages. We even hide navigation bars and show only when user requests for it.

5.2 Business Services or Business Logic Layer

This layer of the system is responsible to cater the logical and mathematical processing of data. All the computations and processing on the raw data is handled in thin layer. It is responsible to extract all the essential and meaningful information from raw data. It also collaborates the processed data into a desirable form and keeps it available to be consumed by the user interface layer. This layer provides the user interface layer with all the possible data that it might require. It basically behaves as producer and UI layer as the consumer. It performs various algorithms on data to process it and format it with the norms and rules of the UI layer. The raw data is pruned here based on different features and functionality of the visualization requested. Each data-set required for visualization differs on the basis of attributes required by it. Based on the requirement this layer extracts the attributes and provide with the pruned data. The layer acts as the backbone of the whole system. Logically this layer is the brain of the system and any changes required in feature of the system will directly or indirectly be catered by this layer.

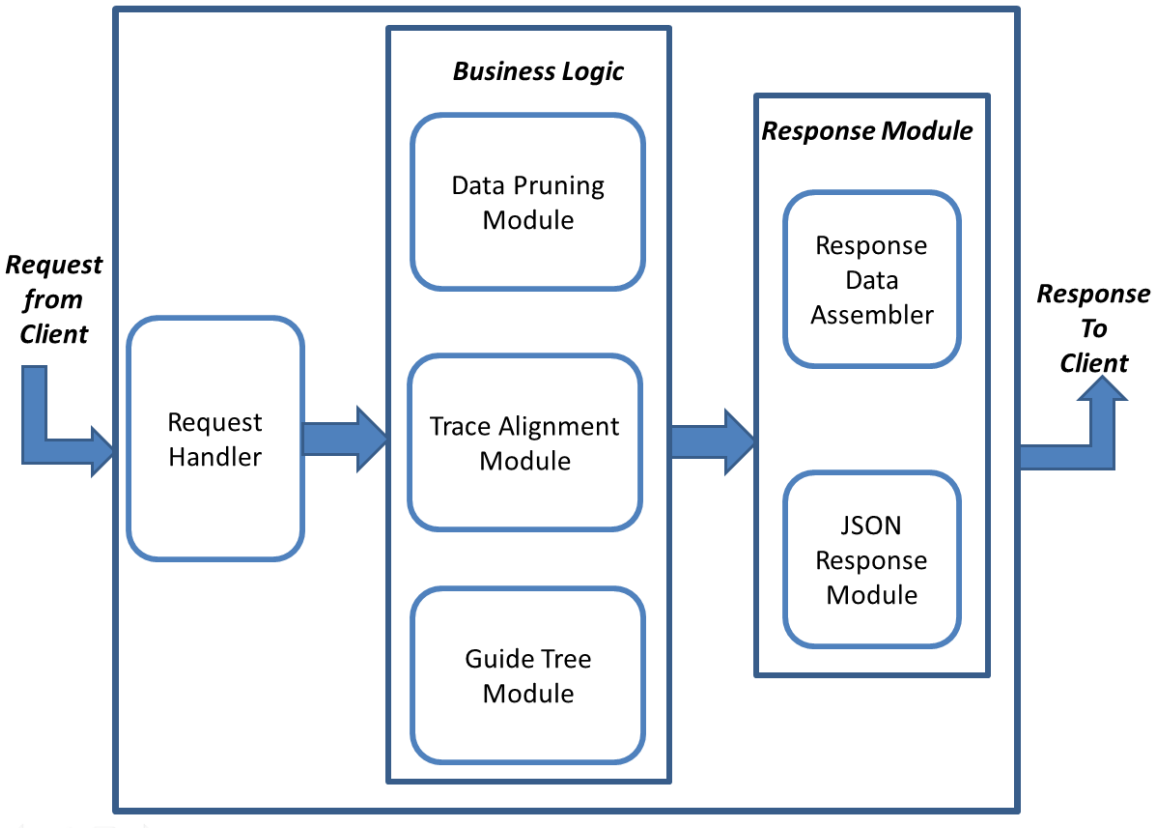

Figure 5.5: Web Based Process Mining and Visualization System - Business Layer Design

5.2.1 Design

Business layer is main computing layer of the system. It conforms to all the business process and refines the data accordingly. All the business and logical models are integrated in this layer serve the purpose of crafting the data that conforms to the system norms. There are various models in the system shown in figure 5.5, which are catered by this layer. Each model is specifically mapped to a visualization view. By the systems virtue of having multiple visualization it is obvious to divide the whole layer based on visualization models and data pruning algorithm associated with each model.
Business logic layer is designed to favor the application in such a way that would enhance the performance and cater to future inclusions and customizations as and when required. This layer can be further divided into sub-parts based on the functional roles each code trace handles. This layer handles a variety of functional roles like algorithmic processing of the raw data, Data pruning based on requests, Collection of processed data and formation of JSON which comprises of the pruned data.
In essence this layer works as a black box which handles all the data requests from the client and supplies with the appropriate response. Input to this layer is raw data along with the requested type of visualization from client and the output is a customized JSON object which is ready to be fed into the UI. As the operations above naturally can be divided into multiple steps, that's exactly how it has been tackled in the system. The systems essentially break up the whole layer into three distinct communicating modules. The aim towards this division is to ensure performance, accuracy, customization, modularity and scope for future expansion. In the following section we will state how the aimed targets are achieved by describing the design of this layer.

Independent business logic To handle various visualization techniques it is essential to separate out each technique to ensure proper functioning. Different visualization mainly differs on an attribute level. Two distinct visualizations will essentially have a different representation of a particular attribute or may have a different attribute all together. Keeping each visualization in a separate code block provides essential integrity to the data. It also ensures easy customization required in future. A new visualization would essentially

be added without any changes in the existing ones. A new code block will essentially be added which will be calling into some of the existing code blocks to ensure re-usability and maintain modularity.

Detached pruning algorithms In order to prune the input data from useless data and meta-data a separate code block is utilized. This design ensures not only modularity and re-usability in the system but also rapid development and customization. The pruning logic is not mixed into any visualization logic but has a separate existence. The caller calls into various code blocks to ensure the data pruning as per the requested visualization. Often it is the case that different visualization might need variety of pruning, but essentially follow a similar baseline. The similarity in the pruning provides us with an elegant way to reuse the code.

Separate request handler The business layer's working differs based on requests. To ensure accuracy in the processing of the request it is necessary to accurately read the requests. The system should be able to differentiate in requests placed by the user. To ensure the accurate functioning of the request handler, the module is separated from the fuzzy business logic and pushed ahead in a separate module. It essentially reads the requests and interprets it accurately. According to the request it later calls into various business logic required to serve the request and hands over the execution to business logic and pruning module. It also ensures that the data is pruned before calling into the business logic modules to check any chances of malfunctioning due to impurity in the data received.

Data Assembler It is really important to ensure data integrity and accuracy of the system. By the virtue of modularity in the system, a great emphasis needs to be given on the part where all the data is gathered after processing in accordance to the request. The processed data from various code blocks and modules is collected and integrated in a separate module. This module ensures the integrity of data and provides confidence in the generated response. It receives the output of various modules and combines them into the final output. This output is now ready to be transformed into response.

Response creation Response creation is a separate module as it has to account for the communication protocols and has to obey and abide to the norms of HTTP. This is a very critical part of the whole system as in this module the created response is transmitted over the web and fed into the user interface. The fact that this system is data intensive, this module has to keep a check on the performance of the system at all times. The best and fastest way to achieve a successful result is to transmit the data in form of JSON objects over HTTP protocol. This module combines all the attributes and various sequences, wraps up everything into one whole object and sends the final object as the response.

5.2.2 Technology

The description of the business layer provided above states the fact that this layer is the backbone of the whole system. Flawless and expeditious working of the layer is essential and has a high impact on the overall performance of the system. The technology stack used to create this layer was handpicked to ensure that the system provides cutting edge solution to the problem.

Modular programming Modular programming is a proven concept which ensures re-usability and easy customization in future, both of which is really important aspects for our system. Modular programming is a methodology which aims at building discrete functional block of code specialized to perform a specific task in the system. The whole functionality of the system is broken down into multiple small functional units rather than building the one whole monolithic system. Each module specifically calls into set of interfaces specially crafted to perform a task. Having code blocks with designated functionality, it becomes easy to manage the flow. It also opens up for easy customization and addition to the feature and functionality of the system. As our system is aimed to solve the visualization issue for multiple domains, it is very necessary for the system to be customizable per domains without compromising on performance and accuracy. In the system, there are four main modules:

1. Data Pruning Module

2. Business Logic module

3. Data Assembler

4. Request and Response handler

The modules are connected to each other by means of method calls but work independently once received the required information from the called module. Request/Response handler is the module that interfaces the user interface and accepts the requests. After receiving the request Data pruning module is called by the request handler. Data pruning necessarily makes sure that the incoming data is good to be processed. After the initial data check, business logic module is called. There are multiple paths available in the business logic module; the called path is decided based on the requested visualization. This module mainly consists of various algorithms that process the data in accordance with the requested visualization type. Once the algorithmic pass over the data finishes, data assembler module is called in to assemble the data in the required output format. This module is necessarily the same for all the requested type of visualization; hence it is reused every time a call is made. Last step before the output is fed to the UI is again calling the Request/Response handler. It is basically a service which handles the response creation. The created output is assembled in the form of JSON object and communicated over HTTP protocol to be fed on the UI.

Java We have used Java 7 as the development language. Java is a platform independent language and is widely accepted throughout. It is also one of the most efficient development platforms. Its seamless integration with HTML and JSON are also one the major factors behind its choice as the development language. A lot of prepacked libraries are available in Java which is required for the development of our system. Availability of such prepackaged libraries significantly minimize the development time.

JSON JavaScript Object Notation often known as JSON is essentially a fat free option for data interchange for web applications[18]. Its works on basic object notations and can have multiple objects in form of an array. Due to the fact that it has less attributes for the same amount of data it proves to be faster compared to XML. It uses name-value pairs to map data. The size of the files formed on the same amount of data using JSON is

much smaller than compared to XML. Although it is less verbose as compared to XML but the significant amount of performance gain is priority in our system. JSON also uses comparatively less resources in order to be processed compared to XML. JSON also finds seamless compatibility with JavaScript which is the main part of our systems front-end.

5.3 Data Services or the Data Layer

This is the thinnest layer of the whole system. The main purpose of this layer is to hold raw data. This raw data is fed to the middle tier for further processing. Due to the nature of our system, this layer has minimalistic intrusion in the processing and display of data. The sole purpose of this layer is to hold data supplied by user. The system is designed in such a way to support portable and platform independent nature. One of the reasons of web based design of our system is to promote portability. The layer is designed in such a way so that it will favor the design.

5.3.1 Design

As explained the data layer is a thin layer in this system. As per the system design the data layer captures data uploaded by the user. Application hosts a web page that interfaces the upload commands. User can use this page to upload the file. As we can see, the user will be uploading the data using web application. The data will be transferred over HTTP protocol. As this data transfer would consume network bandwidth, it is best to minimize the size of this data file. One of the ways to do that is to minimize the overhead data and memory utilized in maintaining the content and preserving the format. Excluding any meta-data that is attached with the file and writing and storing the file in plain text format.

5.3.2 Technology

The various specifications and design constraints described in above section was primarily used to examine and differentiate between various available technologies. To preserve the thin nature of the layer and minimize the overhead of meta-data and formatting, we decided to take Comma Separated File as an option to hold data. The input by the use will be

provided in .CSV file format. The CSV files store data and separate them by using comma, which is the minimislistic way of preserving formatting of the data.

5.4 Web API

Application Programmable Interface referred to as API is a public interface that exposes some functionality of the system in a limited way[24]. The purpose of exposing this functionality is to allow users and programmers to use the specific functionality. These functionalities can be used to build other applications. It is essentially piggybacking the existing specialized functionality and creating a solution that addresses a specific use case. API's can be accessed using specific keys and valid authentication tokens. It acts as a gateway which allows authorized user access and routes the requests based on the keys other parameters. They provide complete set of rules and specifications about how an external program is going to interact with rest of the system. There are various advantages of building an API which often revolve around improving the performance and reusability. Some of the advantages of developing an API can be viewed as following:

1. API's expose the key system features and functionalities that can be leveraged to build a new application around it.

2. API's provide an encapsulated implementation allowing user the ease of implementation by removing the need to import and understand the code base.

3. API improves the reusability of code.

4. API's are often built with a lot of efforts to provide maximum performance, which has a positive impact on application utilizing those API's.

5. API's provide easy means of development and enhancements of Web as well as Desktop based application.

In our system we have developed API and aim towards gaining the above advantages. The choice of API for our system is Rest API. Basing our decision on the fact that primary aim while building our system was to develop a web based application that allows further

mobility and independence to the end user. The beauty of stateless nature and use of HTTP protocol throughout our system provide us with additional advantages of building our Rest API.

5.4.1 Why build an API for Visualization System

Our system primarily targets to solve the issue of mobility and platform independence in the data extensive visualization systems. The goal is to make this system usable across multiple domains. But doing so is actually a huge task which requires a lot of research and work. There are number of ways in which particular data can be viewed, which largely depends upon the application and specific use case. To provide all possible visualization is impossible. To meet the primary goal of our system design, it is best to expose the features and functionality of the system in terms of an API.
The core functionality of the application is based on the various middle tier blocks within the system. These blocks primarily are pruning and the alignment algorithms that work on the incoming data to convert it into meaning full information. That information can be viewed in form of graphical visualization to study patterns and draw various conclusions. In the current system, our domain is specified to be with in medical sciences specifically in the Emergency Medical services. The visualization patterns are developed using specific use cases from the domain. With change in domain the visualization might change, but the data behind the visualization would apparently remain intact. We have developed API that would help users to build their own visualization following a set of rules. That visualization can use the API as the middle tier essentially removing the need to develop and write various algorithmic methods.
The API is aimed to serve the purpose of providing the bare minimum domain centric user developed front end of the visualization with the processed data. This fulfills the primary goal of our system of providing visualization solution to various domains. The entire API is built as Rest API leveraging the HTTP protocol for communication purposes. The API offers one PUT method to upload the data file in .csv format and multiple GET methods to get the data in form of JSON for supporting various visualizations. Below tables provides various valid call list.

URL	/fileUpload?filePath=:path
Method	PUT
URL Parameters	filePath: Path of the data file in .csv format
Success Response	Code: 200 Success
Error Response	Code: 400 Bad Request

Table 5.1: API to Upload the data file to the server

URL	/getVisualization?type=:type
Method	GET
URL Parameters	Type: The type of visualization Requested
Success Response	Code: 200 Success
Error Response	Code: 400 Bad Request

Table 5.2: API to get the visualization data in form of JSON

Chapter 6

Getting a Trace Aligned Visualization - An Application Flow Example

In this section we will describe the working of whole application. It will describe about the various subsystems involved in processing a single user request. We will be taking an example of getting a Trace Aligned visualization. A thorough description of how the request is processed and what actions and activities are carried out in order to respond to the user request is described in figure 7.1. We will also describe the chain of events that occurs during the request processing and response creation.

As we know we are taking an example of getting trace aligned visualization to describe the whole system and its working. It is important to understand that one of the requirements of the system was to enable user to be able to get the visualizations on the locally owned data. In order to achieve that the system needs a data input from the user before it can show the visualizations. For all visualizations, the uploaded data by user is utilized. It is necessary step to obtain the data from user before proceeding to processing and visualization steps.

6.0.1 Navigating to the Application Host

In order to start using the application, it is required to have a web browser that would act as the client in the system. To navigate to the application page, the URL of the application must be typed in the address bar of the browser. As soon as the URL is entered, the browser navigates to the hosted application. The home page of the application shown in figure 7.2 is displayed. On the home page, there are multiple links available that explains about the application and how to use the application. User can easily navigate to these links by using

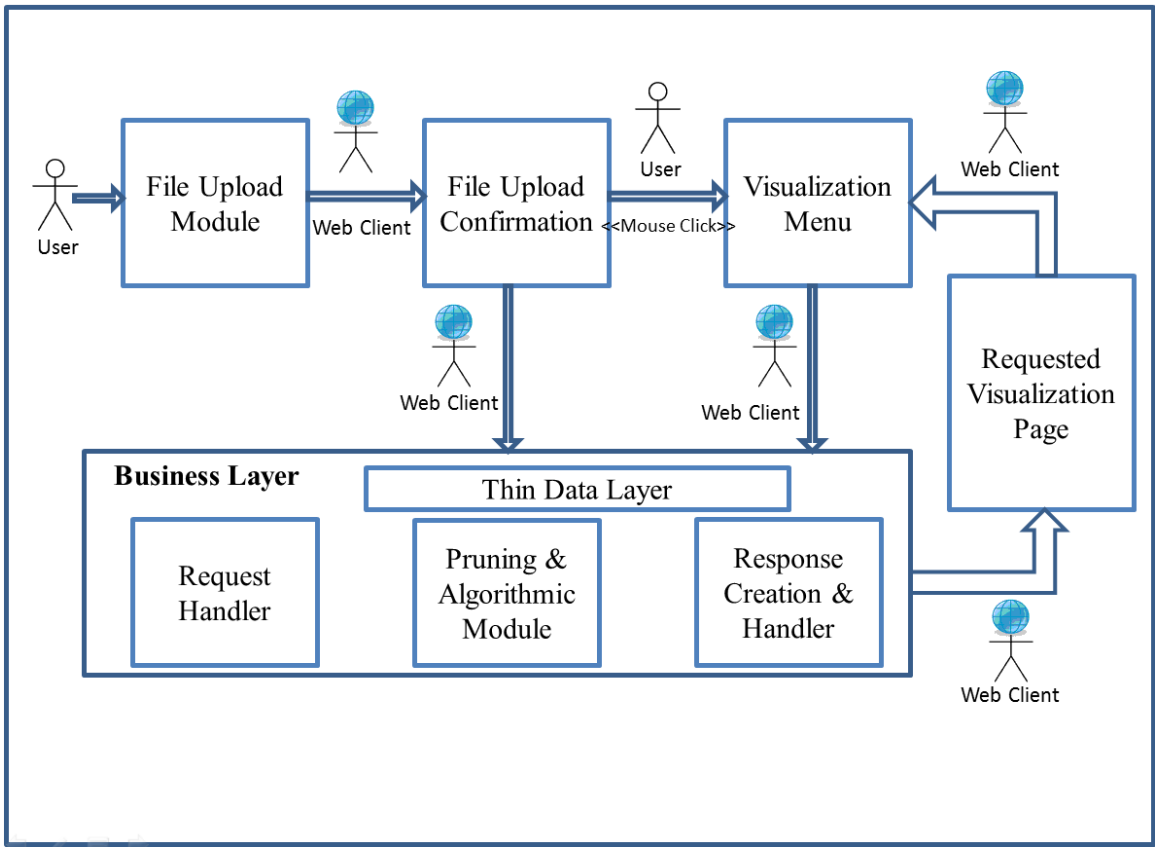

Figure 6.1: Actions Execution for Getting a Visualization

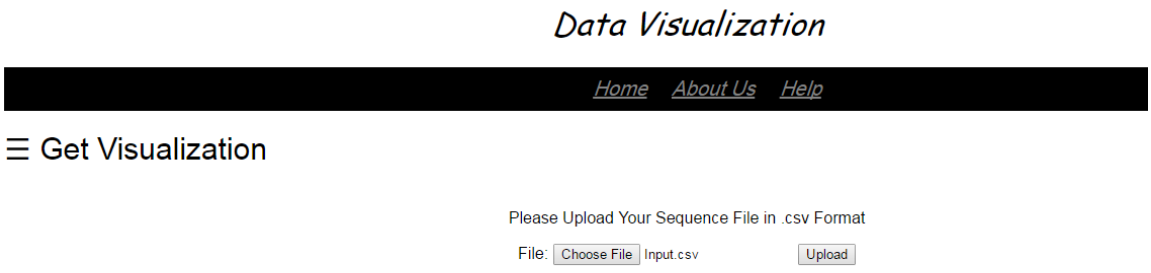

Figure 6.2: Web Based Process Mining and Visualization System - Welcome Page

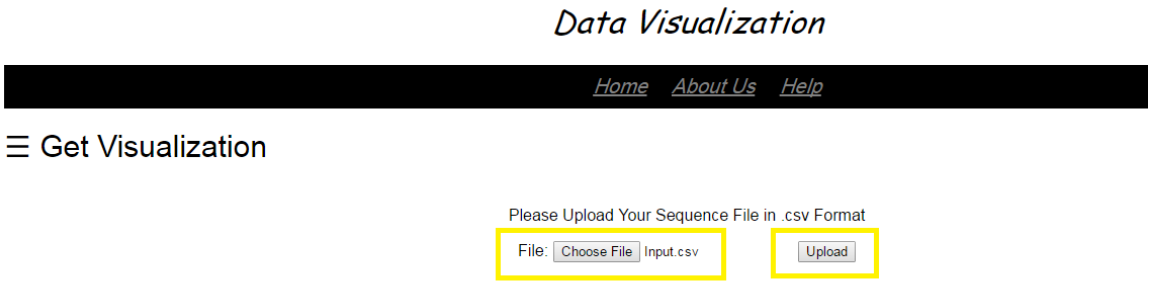

Figure 6.3: Web Based Process Mining and Visualization System - File Upload Page

the navigation bar.

6.0.2 Uploading the Data File

The first task User is required to do before being able to get visualization is to upload the data file. This is the data file which contains records of all the events and traces. This file has to be in .csv format, which is the chosen data file format for reasons explained in previous sections. Figure 7.3 shows the UI page which resembles to the upload file panel. In order to upload the data file, user is required to be on the Home page of application. As soon as user clicks the Browse button on the web page, a file selection wizard opens up. User can select the file of his choosing and press OK. As soon as the OK button is pressed, the file path can be seen and verified in the Address Box next to the browse button. Once selected the file, user should click on the Upload Button. The click on upload button, creates a PUT requests and call the /uploadFile?filePath service to upload the file on to the server. Now the file is available to the system for processing. After this the user can request any time of visualization he wishes to see.

6.0.3 Requesting the Desired Visualization (Trace Alignment)

Now as soon as the required data file is uploaded by the user, he will be able to navigate and view any type of visualization provided. In order to choose the desired visualization type, the user can access the navigation panel on the left of the application page. The navigation bar is self-collapsing in design and will expand only when the user hovers over it. As soon as the mouse is hovered over the navigation bar, it opens up and

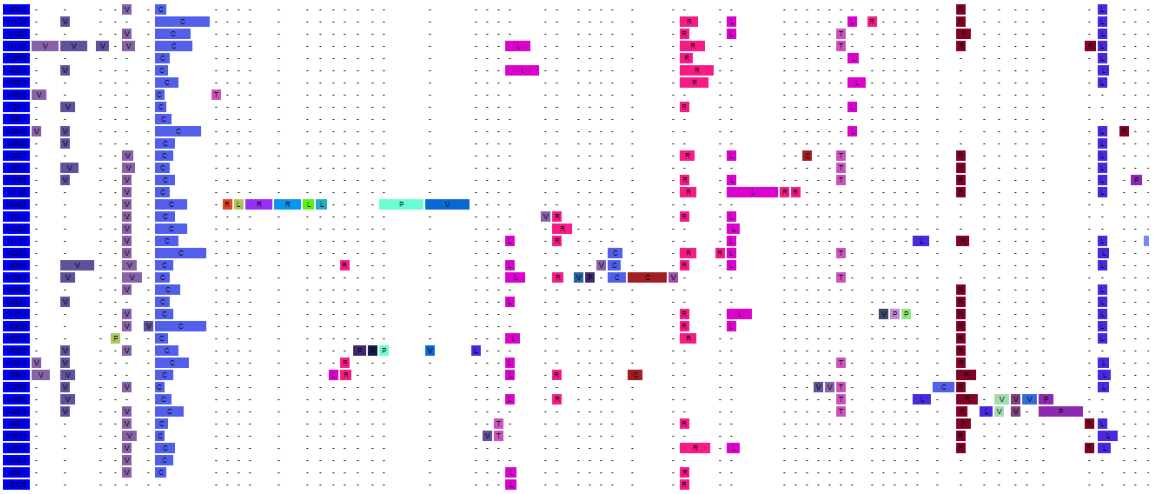

Figure 6.4: Trace Aligned Sequence Obtained from Web Based Process Mining and Visualization System

shows all the available options for visualization type. User can choose any visualization type. In this example we will choose Trace Alignment as the option for visualization.

On clicking the desired visualization (Trace Alignment), the client originates a GET request primarily with the visualization type as the value in the request. The client calls the /getVisualization?type=TraceAlignment. Once the visualization type is selected the application navigates to the new page, which displays the requested visualization (figure 7.4).

6.0.4 Reading and Understanding the Visualization

As soon as the user requests for visualization, the application navigates to a different page based on the request parameters. A multicolored grid is displayed on the page as shown in figure 7.4. This grid is basically the requested visualization showing the data in the required form. The first column shows the "Sequence Number" and the remaining column towards the right of the Sequence number shows various "Activities" in that sequence. Different activity cells are colored differently. The activity cell displays the first character of the Activity Name. In order to get more details about the activity, user can click on desired cell. The click opens up a dialog box which explains the data behind the activity cell.

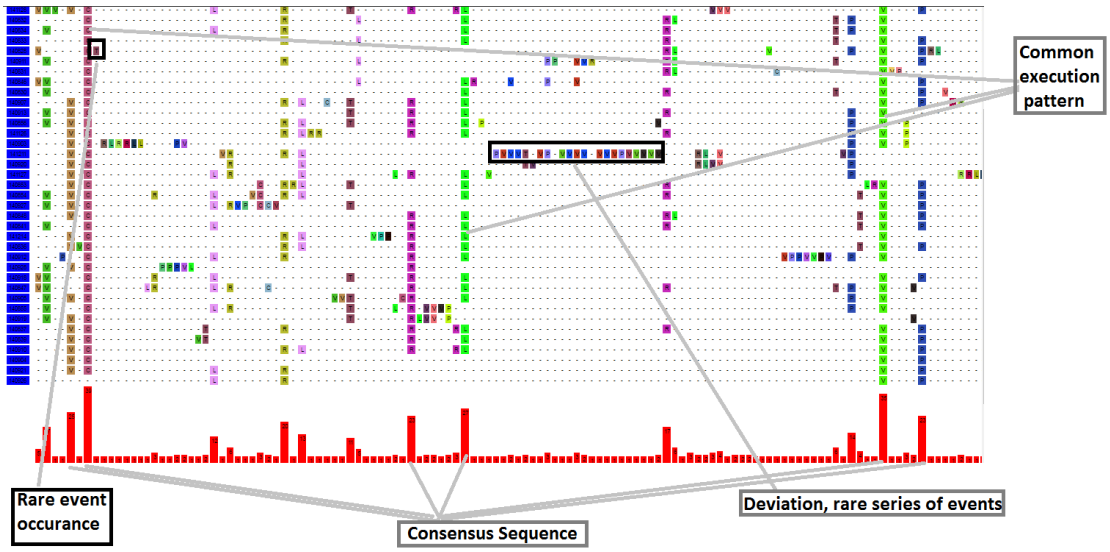

Figure 6.5: Understanding the obtained Trace Aligned visualization

6.1 Case Study: Trauma Resuscitation Process

In this section we present the final visualization obtained on the real time data collected by our partner team. The data-set was collected in collaboration with Children's National Medical Center, Washington DC. The data-set consists of activities and events resulting in trauma resuscitation of injured children. There were 33 children and total of 8 classes of activities were performed. The activities were captured using a video camera and were time coded. Each activity has a start and end time which is used to plot the events in logical order of occurrence.

To analyze the data collected, we ran our algorithm on the data set and obtained the resulting visualization. Figure 7.4 shows the Trace Aligned visualization obtained from the data set. In the figure 7.4, we can easily point out to various important portions of the visualization. Analysts can draw various conclusions based on the occurrence of events as well as their absence. Below are few observations that could be concluded:

1. Consensus sequence: The consensus sequence obtained here is based on the metrics provided by our medical experts. It can vary according to the requirements. The sole purpose of the consensus sequence is to give the common event execution pattern. The

events in consensus sequence are most likely to be noticed in any process execution.

2. Rare event occurrence: In the whole visualization of 33 process traces, there are a few number of events marked in figure 8.1 that are rare. Rare event occurrences are an indication of abnormal process execution. It can be attributed to specific reason related to domain. Experts can later analyze the reason and develop a hypothesis.

3. Rare series of events: In some processes, an anomalous series of events might be noticed. This is evident to prove that the particular process has some deviation with respect to other processes monitored.

In various domains and processes, time duration id often very important parameter to be considered. Trauma resuscitation is of the domains where time duration of an event plays an important role. Figure 7.4 shows the Trace Aligned visualization with event duration.

Chapter 7

Conclusion

This thesis provides a process mining tool that leverages web based design to provide platform independence and mobility. This would be of great help, specially in the area of health and medicine where an on the go methodology is widely used. The inclusion of Trace Alignment algorithm with time duration provides this tool with a sophisticated process mining technique. This could be leveraged in various domain for process mining and visualization. As the system is currently capable of providing four different visualization of the input data, it becomes easier for analysts to unfold the intricate hidden details. The system is developed by leveraging cutting edge technologies like like D3.js and DHTML, which provides faster and reliable processing of data. The modular development of the system favors code reuse to a very large extent. As process mining techniques can be applied to any domain, the modular structure of the system code base is designed to support easy integration and customization based on the domain requirements. The modular development of the system also supports the scope of future improvement and inclusion of more mining algorithms and visualization screens.

The primary area of studies that is Trauma Resuscitation proves that the our system has capability of working with the real time data. As shown, it can be used to find anomalies and conformance in a process run. The availability of conformance sequence, provides system with an ability to measure the deviation of process run with an existing process model. The real power and insights provided by our system would be unleashed when used within multiple health centers to monitor process sequences and analyze them based on a model. This could also be helpful in developing a process model for medical process, which intend to be more of irregular in nature. The availability of a web API will prove to be very useful once publicly made available. Analyzers can easily use the APIs to develop their own

application without having to built their own logic layer.

Chapter 8

References

1. Sen Yang, Xin Dong, Moliang Zhou, Xinyu Li, Shuhong Chen, Rachel Webman, Aleksandra Sarcevic, Ivan Marsic and Randall S. Burd "VIT-PLA: Visual Interactive Tool for Process Log Analysis"

2. Sen Yang, Moliang Zhou, Rachel Webman, JaeWon Yang, Aleksandra Sarcevic, Ivan Marsic and Randall S. Burd, "Duration-Aware Alignment of Process Traces", Accepted for Industrial Conference on Data Mining (ICDM 2016), New York, NY, July 13-17, 2016

3. Bose, RP Jagadeesh Chandra, and Wil MP van der Aalst. "Process diagnostics using trace alignment: opportunities, issues, and challenges." Information Systems 37.2 (2012): 117-141

4. Van Der Aalst, Wil. Process mining: discovery, conformance and enhancement of business processes. Springer Science and Business Media, 2011

5. Perer, Adam, Fei Wang, and Jianying Hu. "Mining and exploring care pathways from electronic medical records with visual analytics." Journal of biomedical informatics 56 (2015): 369-378

6. Perer, Adam, and Fei Wang. "Frequence: interactive mining and visualization of temporal frequent event sequences." Proceedings of the 19th international conference on Intelligent User Interfaces. ACM, 2014.

7. https://github.com/d3/d3/blob/master/API.md

8. Ivan Marsic "Software Engineering"

9. Prom Monroe, Megan, et al. “Temporal event sequence simplification.” Visualization and computer Graphics, IEEE Transactions on 19.12 (2013): 2227-2236

10. http://www.processmining.org/xesame/start

11. J.C.A.M Buijs “Mapping Data Sources to XES in a Generic Way”

12. Markus Hofmann, Ralf Klinkenberg ”RapidMiner: Data Mining Use Cases and Business Analytics Applications”

13. Malik, Sana, et al. “Cohort comparison of event sequences with balanced integration of visual analytics and statistics.” Proceedings of the 20th International Conference on Intelligent User Interfaces. ACM, 2015.

14. Mark Burstein, Christoph Bussler, Michal Zaremba, Tim Finn, Michael N. Huhns ”A Semantic Web Services Architecture”

15. Nurzhan Nurseitov, Michael Paulson, Randall Reynolds, Clemente Izurieta “Comparison of JSON and XML Data Interchange Formats: A Case Study”

16. Ebba Thora Hvannberga, Effie Lai-Chong Lawb, Marta Kristin Larusdottir “Heuristic evaluation: Comparing ways of finding and reporting usability problems”

17. http://www.purelybranded.com/notes/why-use-css-in-website-design/

18. Gregor Richards, Sylvain Lebresne, Brian Burg, Jan Vitek “An Analysis of the Dynamic Behavior of JavaScript Programs”

19. http://daemon.co.za/2014/04/introduction-fullstack-fundamentals/

20. Miriam Clemente, Beatriz Rey, Aina Rodriguez-Pujadas, Alfonso Barros-Loscertales, Rosa M. Banos, Cristina Botella, Mariano Alcaniz and Cesar Avila “An fMRI Study to Analyze Neural Correlates of Presence during Virtual Reality Experiences

21. https://facebook.github.io/react/tutorial/tutorial.html

22. http://www.highcharts.com/docs

23. https://www.dashingd3js.com/why-build-with-d3js

24. S. Clarke, “Measuring API usability”, Dr. Dobb’s Journal Windows/.NET Supplement, May 2004, pp. S6-S9