Politecnico di Milano
Polo di Como

Scuola di Ingegneria dell’Informazione

Corso di Laurea Specialistica in Ingegneria Informatica

Generating Log Files for Business Process Model Mining

Tutor universitario: Prof. Giuseppe Pozzi

Elaborato finale: ARAGHIZADEH Roya
matr.: 877073

Anno Accademico 2018-2019

Table of Contents

Abstract

Over the past decade process mining has emerged as a new analytical discipline able to answer a variety of questions based on event data. Event logs have a very particular structure; events have timestamps, refer to activities and resources, and need to be correlated to form process instances. Moreover, event logs may be huge and may need to be decomposed and distributed for analysis. These aspects make it very cumbersome to analyze event logs manually therefore *Process mining should be repeatable and automated.*

A process model is a representation of a real-world process, where a visual process model is commonly defined as a process diagram. On the other hand, process models can also be non-visual, for example, they might specify process execution semantics.

Process diagrams can form a good basis for business process management activities since they enable process representation, observation and analysis.

Based on the results of the analysis, existing processes can be changed, where the applied changes can be simulated or observed to see if they gain any improvements.

If these activities perform continuously, processes undergo permanent improvements with positive impacts on the efficiency and effectiveness of an organization. This cycle of process improvement activities is commonly known as the PDCA (plan–do–check–adjust), an iterative four-step management method used in business for the control and continuous improvement of processes and products.

After modeling the processes, it is time for the role-playing of Process Mining. Process Mining is a process analysis method that aims to discover, monitor and improve real processes (processes not assumed) by extracting knowledge easily from available event logs in the systems of current information of an organization. It goes beyond the pure presentation of the key data of the process, recognizing the contextual relationships of the processes, presenting them in the form of graphic analysis in order to diagnose problems and suggest improvements in the quality of the process models. With Process Mining it will be possible to detect or diagnose problems based on facts and not on conjectures or intuitions. Process mining seeks the confrontation between event data (observed behavior) and process models (hand-made or automatically discovered). Through the pairing of event data and process models, it will be possible to check compliance, detect deviations, predict delays, support decision making and recommend process redesigns.

This Thesina aims at developing a tool which writes randomly generated log files (execution history) compliant with the user’s requests for a previously defined process Model. Log file well then be used to test process mining algorithms. The document is summarized with discussion, personal experience and challenges faced during the development and implementation process.

Sommario

Negli ultimi dieci anni il processo di “mining” sta iniziando ad essere concepito come una nuova disciplina analitica in grado di rispondere a una varietà di domande basate su dati transazionali. I registri eventi hanno una struttura molto particolare; gli eventi hanno una marca temporale, fanno riferimento ad attività e risorse e devono essere correlati in modo da formare esempi di processo. Inoltre, i registri eventi possono essere vastissimi e devono essere scomposti e distribuiti per l'analisi. Questi aspetti rendono molto complicato analizzare manualmente i registri eventi, pertanto il processo di mining dovrebbe essere ripetibile e automatizzato.

Un modello di processo è una rappresentazione di un processo del mondo reale, in cui un modello di processo visivo è comunemente definito come un diagramma di processo. D'altra parte, i modelli di processo possono anche essere non visivi, ad esempio potrebbero dettagliare la semantica dell'esecuzione del processo. I diagrammi di processo possono costituire una buona base delle attività di gestione dei processi aziendali poiché consentono la rappresentazione, l'osservazione e l'analisi dei processi. Sulla base dei risultati dell'analisi, è possibile modificare i processi esistenti, in cui è possibile simulare o osservare le modifiche applicate per vedere se si ottengono miglioramenti. Se queste attività si svolgono continuamente, i processi subiscono miglioramenti permanenti con impatti positivi sull'efficienza e l'efficacia di un'organizzazione. Questo ciclo di attività di miglioramento dei processi è comunemente noto come PDCA (plan-do-check-adjust), un metodo di gestione ripetitivo in quattro fasi utilizzato nelle attività commerciali per il controllo e il miglioramento continuo di processi e prodotti.

Dopo aver modellato i processi, è il momento di interpretare il processo di mining. Il mining è un metodo di analisi dei processi che mira a scoprire, monitorare e migliorare i processi reali (processi non ipotizzati) estraendo facilmente la conoscenza dai registri eventi disponibili nei sistemi di informazioni correnti di un'organizzazione. Va oltre la pura presentazione dei dati chiave del processo, riconoscendo le relazioni contestuali dei processi, presentandoli sotto forma di analisi grafica al fine di diagnosticare i problemi e suggerire miglioramenti nella qualità dei modelli di processo. Con il processo di mining sarebbe possibile rilevare o diagnosticare problemi basati su fatti e non su congetture o intuizioni. Il mining cerca il confronto tra i dati sugli eventi (comportamento osservato) e i modelli di processo (creati a mano o scoperti automaticamente). Attraverso l'associazione di dati di eventi e modelli di processo, sarà possibile verificare la conformità, rilevare deviazioni, prevedere ritardi, supportare il processo decisionale e raccomandare le riprogettazioni del processo.

Questa tesi mira a sviluppare uno strumento che scrive i file di registro generati casualmente (cronologia di esecuzione) delle attività conforme alle richieste dell'utente per un modello di processo precedentemente definito. Il file di registro può essere utilizzato per testare algoritmi di mining di processo. Il documento è sintetizzato con discussione, esperienza personale e sfide affrontate durante il processo di sviluppo e implementazione.

List of Figures

1. Introduction

In recent years, the amount of available data has been growing in an exponential pace. This is in turn made possible due to the immense growth in number of devices recording data, as well as the connectivity between all these devices through the internet of things.

Big data is a term applied to data sets whose size or type is beyond the ability of traditional relational databases to capture, manage and process the data with low latency. With the emergence of big data and new data sources, a challenge posed to today's organizations consists of identifying how to align their decision-making and organizational processes to data that could help them make better-informed decisions. Big Data analytics can help organizations to better understand the information contained within the data and will also help identify the data that is most important to the business and future business decisions.

Business Process Model and Notation (BPMN) provides a graphical representation of business workflows that anyone, from business analyst to stakeholder, can easily understand; aiding in business process analysis and business process improvements. This existing standards and notations from business modeling (BPMN) could potentially be used in Big Data analytics and business decisions.

Data Mining is known as Knowledge Discovery of Data refers to extracting knowledge from a large amount of data i.e. Big Data. Analyze relationship and patterns in stored transaction data to get information which will help for better business decisions. Data Mining refers to deep drive into the data to extract the key knowledge/Pattern/Information from a small or large amount of data. The main concept in Data Mining is to dig deep into analyzing the patterns and relationships of data that can be used further in Artificial Intelligence, Predictive Analysis etc. Big data only refers to only a large amount of data and all the big data solutions depends on the availability of data. It can be considered as the combination of Business Intelligence and Data Mining. Data mining uses different kinds of tools and software on Big data to return specific results.

Today, data is not the problem – Data is everywhere. Most companies have loads of unused process data that can be used for Process Mining. With Process Mining it will be possible to detect or diagnose problems based on facts and not on conjectures or intuitions. Process mining seeks the confrontation between event data (observed behavior) and process models (hand-made or automatically discovered). Through the pairing of event data and process models, it will be possible to check compliance, detect deviations, predict delays, support decision making and recommend process redesigns.

Considering the existence process mining techniques and tools, the goal of this Thesina is how Process mining exploits the information recorded in event logs to perform an analysis of the real process afterwards. Starting from event logs that are randomly generated according to the user’s requirements and parameters from predefined Business Process Models (BPM) and applying Process mining techniques to create an understandable and analyzable Business process model.

1.1 Thesina outline

The structure of this document is divided in numerous chapters. Chapter 2 describes State of the Art on Big Data definition and its Analytics in healthcare which is related to used healthcare process model as sample, looking at the main concept of Business process Model and Notation and also Process mining which is the main aim of this Thesina. Continued by comparing Process mining tools- ProM, Apromore, Disco and the reason why Disco is selected.

Chapter 3 briefly describes the Goals; development steps and touches the topic of event log. Continuing forward it is Constraints and Requirements which is description of standards and recommendations such as XPDL[1], CFM[2], CSV[3] and Gaussian distribution. Chapter 4 demonstrates three important tools that are used such as programming language, IDE[4], process mining tool and describes the system developed for generating log file.

Chapter 5 contains user experience in executing the log file generator application, running process mining tool on generated log file and validating by comparing initial business process model with finale one. Chapter 6 Summarizes the results that is the process models identified over the selected logs. Finally Chapter 7 contains conclusions and mentions possible future works.

[1] XML Process Definition Language

[2] Control flow complexity

[3] Comma Separated Values

[4] Integrated Development Environment

2. State of the Art

2.1 Big Data History

90% of the available data has been created in the last two years and the term Big Data has been around 2005, when it was launched by O'Reilly Media in 2005. However, the usage of Big Data and the need to understand all available data has been around much longer.
In its true essence, Big Data is not something that is completely new or only of the last two decades. Over the course of centuries, people have been trying to use data analysis and analytics techniques to support their decision-making process.

However, in the last two decades, the volume and speed with which data is generated has changed – beyond measures of human comprehension. The total amount of data in the world was 4.4 zettabytes in 2013. That is set to rise steeply to 44 zettabytes by 2020. Even with the most advanced technologies today, it is impossible to analyze all this data. The need to process these increasingly larger (and unstructured) data sets is how traditional data analysis transformed into 'Big Data' in the last decade.

To illustrate this development over time, the evolution of Big Data can roughly be subdivided into three main phases. Each phase has its own characteristics and capabilities. In order to understand the context of Big Data today, it is important to understand how each phase contributed to the contemporary meaning of Big Data.

- Big Data phase 1.0

Data analysis, data analytics and Big Data originate from the longstanding domain of database management. It relies heavily on the storage, extraction, and optimization techniques that are common in data that is stored in Relational Database Management Systems (RDBMS). Database management and data warehousing are considered the core components of Big Data Phase 1. It provides the foundation of modern data analysis as we know it today, using well-known techniques such as database queries, online analytical processing and standard reporting tools.

- Big Data phase 2.0

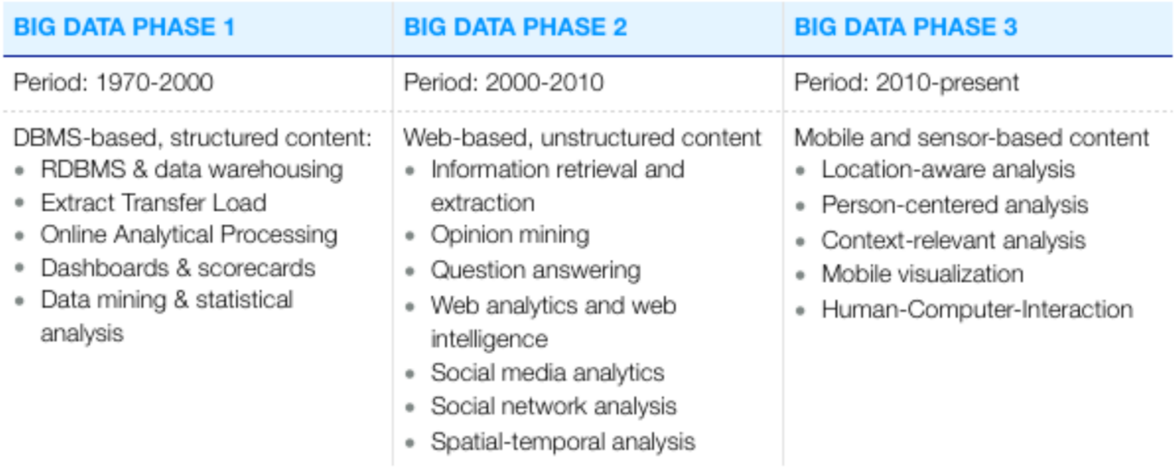
Since the early 2000s, the Internet and the Web began to offer unique data collections and data analysis opportunities. With the expansion of web traffic and online stores, companies such as Yahoo, Amazon and eBay started to analyze customer behavior by analyzing click-rates, IP-specific location data and search logs. This opened a whole new world of possibilities.
From a data analysis, data analytics, and Big Data point of view, HTTP-based web traffic introduced a massive increase in semi-structured and unstructured data. Besides the standard structured data types, organizations now needed to find new approaches and storage solutions to deal with these new data types in order to analyze them effectively. The arrival and growth of social media data greatly aggravated the need for tools, technologies and analytics techniques that were able to extract meaningful information out of this unstructured data.

- Big Data phase 3.0

Although web-based unstructured content is still the main focus for many organizations in data analysis, data analytics, and big data, the current possibilities to retrieve valuable information are emerging out of mobile devices.
Mobile devices not only give the possibility to analyze behavioral data (such as clicks and search queries), but also give the possibility to store and analyze location-based data (GPS-data). With

the advancement of these mobile devices, it is possible to track movement, analyze physical behavior and even health-related data (number of steps you take per day). This data provides a whole new range of opportunities, from transportation, to city design and health care.
Simultaneously, the rise of sensor-based internet-enabled devices is increasing the data generation like never before. Famously coined as the 'Internet of Things' (IoT), millions of TVs, thermostats, wearables and even refrigerators are now generating zettabytes of data every day. And the race to extract meaningful and valuable information out of these new data sources has only just begun.

A summary of the three phases in Big Data is listed in the figure below:



BIG DATA PHASE 1	BIG DATA PHASE 2	BIG DATA PHASE 3
Period: 1970-2000	Period: 2000-2010	Period: 2010-present
DBMS-based, structured content: • RDBMS & data warehousing • Extract Transfer Load • Online Analytical Processing • Dashboards & scorecards • Data mining & statistical analysis	Web-based, unstructured content • Information retrieval and extraction • Opinion mining • Question answering • Web analytics and web intelligence • Social media analytics • Social network analysis • Spatial-temporal analysis	Mobile and sensor-based content • Location-aware analysis • Person-centered analysis • Context-relevant analysis • Mobile visualization • Human-Computer-Interaction

Figure 1 The three major phases in the history of Big Data[5]

2.2 Big Data Analytics for Healthcare

'Big data' is massive amounts of information that can work wonders. It has become a topic of special interest for the past two decades because of a great potential that is hidden in it. Various public and private sector industries generate, store, and analyze big data with an aim to improve the services they provide. In the healthcare industry, various sources for big data include hospital records, medical records of patients, results of medical examinations, and devices that are a part of internet of things. Biomedical research also generates a significant portion of big data relevant to public healthcare. This data requires proper management and analysis in order to derive meaningful information. Otherwise, seeking solution by analyzing big data quickly becomes comparable to finding a needle in the haystack. There are various challenges associated with each step of handling big data which can only be surpassed by using high-end computing solutions for big data analysis. That is why, to provide relevant solutions for improving public health, healthcare providers are required to be fully equipped with appropriate infrastructure to systematically generate and analyze big data. An efficient management, analysis, and interpretation of big data can change the game by opening new avenues for modern healthcare. That is exactly why various industries, including the healthcare industry, are taking vigorous steps to convert this potential into better services and financial advantages. With a strong integration of biomedical and healthcare data, modern healthcare organizations can possibly revolutionize the medical therapies and personalized medicine.

Large amounts of heterogeneous medical data have become available in various healthcare organizations (payers, providers, pharmaceuticals). Those data could be an enabling

[5] *From the Enterprise Big Data Professional Guide*

resource for deriving insights for improving care delivery and reducing waste. The enormity and complexity of these Datasets present great challenges in analyses and subsequent applications to a practical clinical environment. To date, health care industry has not fully grasped the potential benefits to be gained from big data analytics. While the constantly growing body of academic research on big data analytics is mostly technology oriented, a better understanding of the strategic implications of big data is urgently needed.

Big Data analytics can revolutionize the healthcare industry. It can improve operational efficiencies, help predict and plan responses to disease epidemics, improve the quality of monitoring of clinical trials, and optimize healthcare spending at all levels from patients to hospital systems to governments.

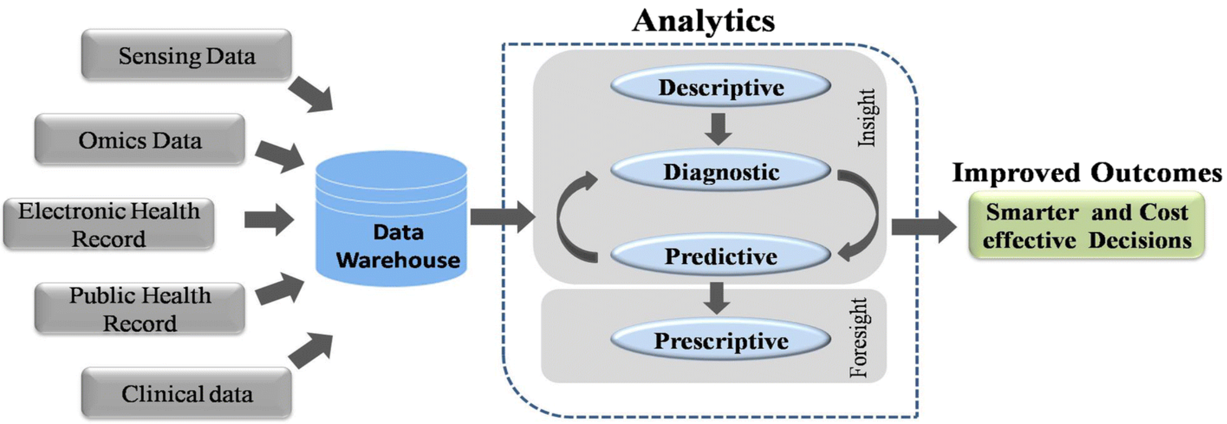


Figure 2 Workflow of Big data Analytics

2.3 Business Process Modeling and Notations

In contrast to an organization's tangible assets, business processes are intangible, and this characteristic makes them more difficult to recognize and observe. So, in order to "work" with processes, we must transform intangible processes into process models. This type of activity is called modeling.

With modeling, the goal is to optimize the process. Although both process modeling and process mapping are techniques that break the process in pieces and allows us to study it, they are not the same. Process mapping is more oriented towards clarifying roles and procedures. On the other hand, process modeling incorporates the work flow and business rules.

To know what is process modeling, we need to understand what is a process and its role inside a company. A defined order of tasks or activities spread through time and space, with a beginning, an end, and clearly defined inputs and outputs. It is a business process. A company has many processes, in all areas, be it marketing, financial, service or production. These processes must be correctly modeled, mapped, optimized, and automatized, to generate value to the customer.

Process modeling is a technique designed to understand and describe the process. It connects and improves the communication between the current and the future state of a process. For example, a diagram that represents how to deliver a product – from the client's order, the entry, communication with shipping, inventory or making to the delivering is a mapping of a process.

A process model is a representation of a real-world process, where a visual process model is commonly defined as a process diagram. On the other hand, process models can also be non-visual, for example, they might specify process execution semantics.

Process diagrams can form a good basis for business process management activities since they enable process representation, observation and analysis.
Based on the results of the analysis, existing processes can be changed, where the applied changes can be simulated or observed to see if they gain any improvements.
If these activities perform continuously, processes undergo permanent improvements with positive impacts on the efficiency and effectiveness of an organization. This cycle of process improvement activities is commonly known as the PDCA (plan–do–check–adjust), an iterative four-step management method used in business for the control and continuous improvement of processes and products.
Business process modelling represents:

- Business activities,
- Information flow and,
- Decision logic in business processes

With the power of visualization, it is used to communicate information regarding a process and the interaction it includes within / between organizations either among the persons reading a model or the persons who create it.

The steps below outline the major steps to take in creating a business process.

- Define the process you are modeling – Define a process in your scope of business operation involved, and what are you trying to achieve
- Identify the starting point of the process
- Identify the different steps in the process.
- Clarify who or what performs each step.
- Decide which type of modeling notation to be use used such as BPMN

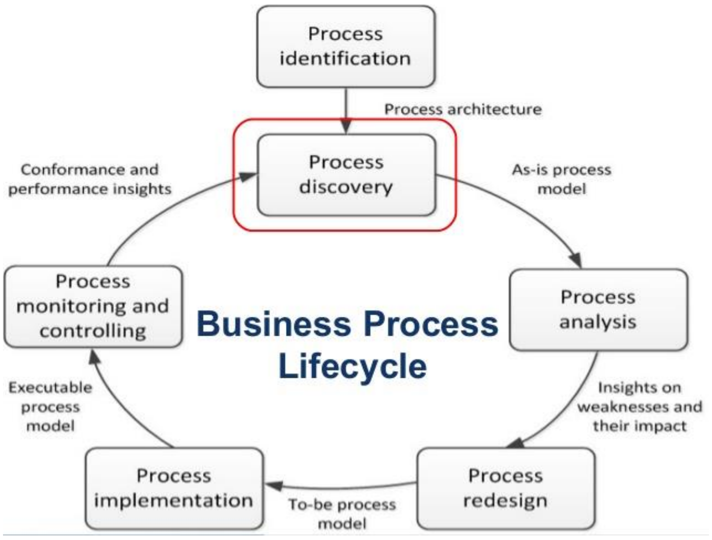


Figure 3 Business process diagram for process improvement

To perform business process improvement, perhaps you could perform the additional gap analysis steps:

- Create an As-is Model (the now state)
- Design the to-be Model (the future state)

- Perform the gap analysis
- Formulate improvement actions

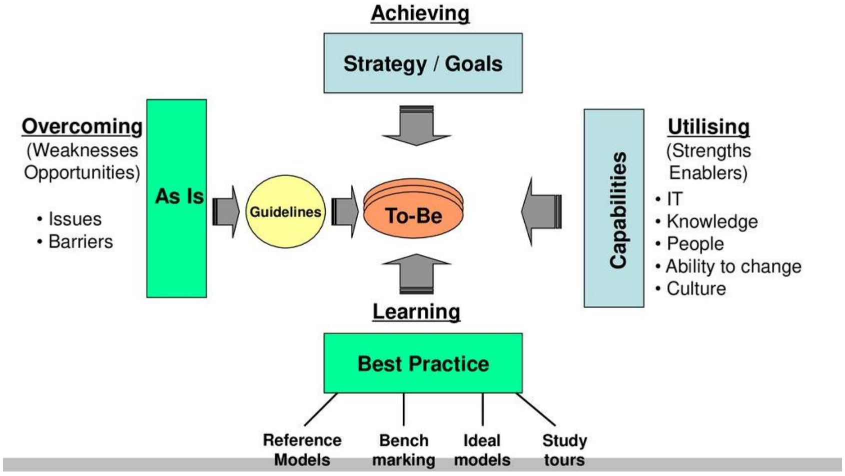


Figure 4 Analysis Steps for Business process improvement

Since process diagrams represent a focal business process management activity, it is important that they reflect actual processes as precisely as possible. Several notations for modeling business processes exist, and the best choice is Business Process Model and Notation.

The Business Process Model and Notation (BPMN) is a standard maintained by the Object Management Group (OMG) and is aimed at business analysts and technical developers. BPMN provides a graphical notation that is widely used for process modelling just like a flow chart method that models the steps of a planned business process from end to end. A key to Business Process Management, it visually depicts a detailed sequence of business activities and information flows needed to complete a process.

At a high level, BPMN is targeted at participants and other stakeholders in a business process to gain understanding through an easy-to-understand visual representation of the steps. At a more involved level, it's targeted at the people who will implement the process, giving sufficient detail to enable precise implementation. It provides a standard, common language for all stakeholders, whether technical or non-technical: business analysts, process participants, managers and technical developers, as well as external teams and consultants. Ideally, it bridges the gap between process intention and implementation by providing sufficient detail and clarity into the sequence of business activities.

The diagramming can be far easier to understand than narrative text would be. It allows for easier communication and collaboration to reach the goal of an efficient process that produces a high-quality result. It also helps with communication leading to XML (Extensible Markup Language) documents needed to execute various processes.

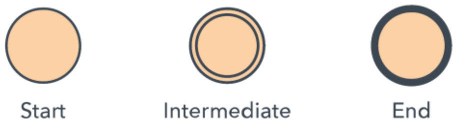
2.3.1 BPMN 2.0 Diagram Elements and Symbols

BPMN depicts these four element types for business process diagrams:

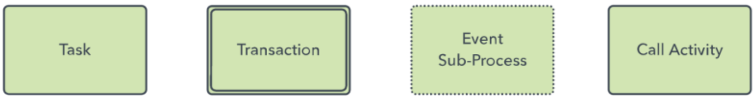
1. Flow objects: events, activities, gateways
2. Connecting objects: sequence flow, message flow, association
3. Swimlanes: pool or lane
4. Artifacts: data object, group, annotation

These are the individual elements and how they are used to define a business process:

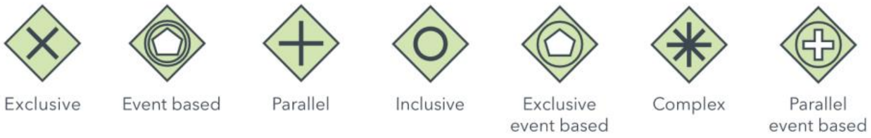
Events
A trigger that starts, modifies or completes a process. Event types include message, timer, error, compensation, signal, cancel, escalation, link and others. They are shown by circles containing other symbols based on event type. They are classified as either "throwing" or "catching," depending on their function.



Activity
A particular activity or task performed by a person or system. It's shown by a rectangle with rounded corners. They can become more detailed with sub-processes, loops, compensations and multiple instances.



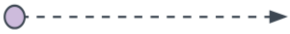
Gateway
Decision point that can adjust the path based on conditions or events. They are shown as diamonds. They can be exclusive or inclusive, parallel, complex, or based on data or events.



Sequence flow
Shows the order of activities to be performed. It is shown as a straight line with an arrow. It might show a conditional flow, or a default flow.

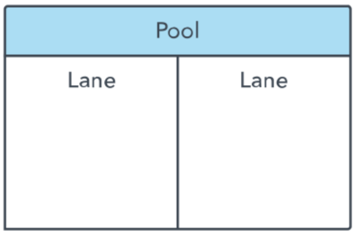
Message flow
Depicts messages that flow across "pools," or organization boundaries such as departments. It shouldn't connect events or activities within a pool. It is represented by a dashed line with a circle

at the start and an arrow at the end.

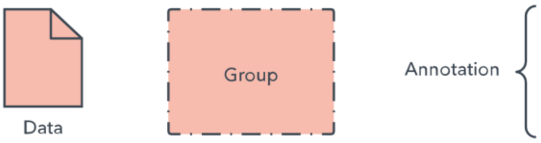


Association
Shown with a dotted line, it associates an artifact or text to an event, activity or gateway.

Pool and swimlane
A pool represents major participants in a process. A different pool may be in a different company or department but still involved in the process. Swimlanes within a pool show the activities and flow for a certain role or participant, defining who is accountable for what parts of the process.



Artifact
Additional information that developers add to bring a necessary level of detail to the diagram. There are three types of artifacts: data object, group or annotation. A data object shows what data is necessary for an activity. A group shows a logical grouping of activities but doesn't change the diagram's flow. An annotation provides further explanation to a part of the diagram.



2.4 Business Process Mining

Over the past decade process mining has emerged as a new analytical discipline able to answer a variety of questions based on event data. Conventional Business Process Management (BPM) and Workflow Management (WfM) approaches and tools are mostly model-driven with little consideration for event data. Data Mining (DM), Business Intelligence (BI), and Machine Learning (ML) focus on data without considering end-to-end process models. Process mining aims to bridge the gap between BPM and WfM on the one hand and DM, BI, and ML on the other hand. Event logs have a very particular structure; events have timestamps, refer to activities and resources, and need to be correlated to form process instances. Process mining results tend to be very different from classical data mining results, e.g., process discovery may yield end-to-end process models capturing different perspectives rather than decision trees or frequent patterns. A process-mining tool like ProM provides hundreds of different process mining techniques ranging from discovery and conformance checking to filtering and prediction. Typically, a combination of techniques is needed, and, for every step, there are different techniques that may be very sensitive to parameter settings. Moreover, event logs may be huge and may need to be decomposed and distributed for analysis. These aspects make it very cumbersome to analyze event logs manually. Process mining should be repeatable and automated.

The goal of Process Mining is to turn event data into insights and actions. Process mining is an integral part of data science fueled by the availability of data and the desire to improve processes.

Process Mining focuses on extracting knowledge from data generated and stored in corporate information systems in order to analyze executed processes. In the healthcare domain, process mining has been used in different case studies, with promising results.

Performing business process analysis in healthcare organizations is particularly difficult due to the highly dynamic, complex, ad hoc, and multi-disciplinary nature of healthcare processes. Process mining is a promising approach to obtain a better understanding about those processes by analyzing event data recorded in healthcare information systems.

2.4.1 Business Process Mining Tools

Process mining tools should support the filtering data in order the avoid malfunction of software. Obviously, these programs must implement the process discovery and provide some models. Therefore, models that are generated by software can be investigated for conformance checking. Conformance checking is a plug-in, which controls the process model from two point of view. First point is the exact model in real situation and second point is how to improve this model in further application.

Using process mining provides several benefits to business owners and organizations.

In the list hereunder the most significant ones:

- Understanding how a process is actually performed.

Most of the time, business owners know very well their processes from a theoretical perspective: what is supposed to happen, when, who is supposed to do what, under which condition.

However, they usually do not have a way to investigate what is really happening throughout the process lifecycle. Traditional reporting, Business Intelligence (BI), statistical tools have difficulties in revealing both the big and the very detailed picture. Such a back and forth navigation from the big picture and the detail is what is the most efficient way to comprehend the real-life situation. Process mining solutions such as Disco focuses on making it easy to digest and exploit.

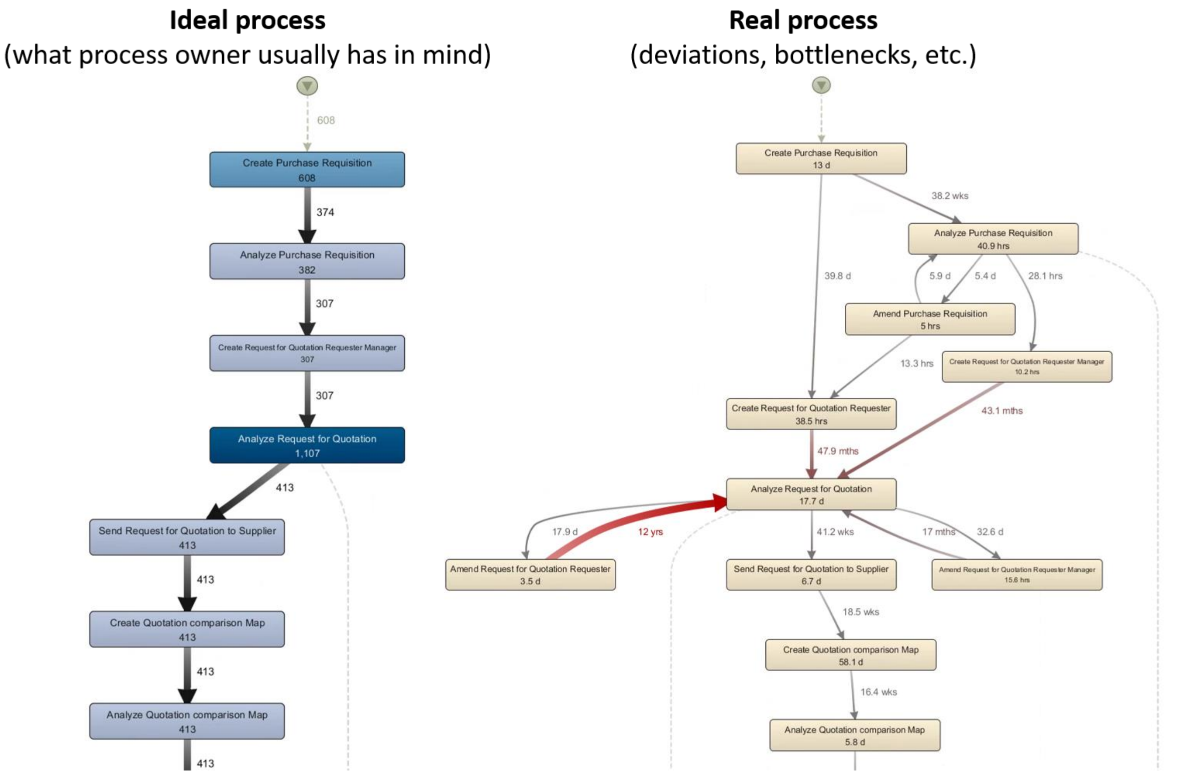

Ideal process
(what process owner usually has in mind)
608
Create Purchase Requisition
608
374
Analyze Purchase Requisition
382
307
Create Request for Quotation Requester Manager
307
307
Analyze Request for Quotation
1,107
413
Send Request for Quotation to Supplier
413
413
Create Quotation comparison Map
413
413
Analyze Quotation comparison Map
413
413
Real process
(deviations, bottlenecks, etc.)
Create Purchase Requisition
13 d
38.2 wks
Analyze Purchase Requisition
40.9 hrs
39.8 d
5.9 d
5.4 d
28.1 hrs
Amend Purchase Requisition
5 hrs
13.3 hrs
Create Request for Quotation Requester Manager
10.2 hrs
Create Request for Quotation Requester
38.5 hrs
43.1 mths
47.9 mths
Analyze Request for Quotation
17.7 d
17.9 d
12 yrs
41.2 wks
17 mths
32.6 d
Amend Request for Quotation Requester
3.5 d
Send Request for Quotation to Supplier
6.7 d
Amend Request for Quotation Requester Manager
15.6 hrs
18.5 wks
Create Quotation comparison Map
58.1 d
16.4 wks
Analyze Quotation comparison Map
5.8 d

Figure 5 Comparison between Ideal and Real processes

- Improving the process flow by knowing the actual flows, frequencies, and delays in the process.

IT system logs track great volumes of useful information, making it feasible to compute paths and lead times to switch from one step in the process to another. One can leverage this valuable information to detect bottlenecks, highlight sub efficiencies, reveal most frequent paths, etc.

- Cost savings by improving the productivity of employees.

Speed up investigation time, and free up time of your analysts to improve processes rather than to analyze the as-is. Reduce overall process time by focusing on the pain points.

- Improving the quality and increasing the efficiency of auditing.
- Verifying that implemented process changes have had the expected effect by comparing the old process to the new process.

Top three Business Process Mining Tools:

- **ProM**
  ProM Tools offers a range of process mining applications on an open source license. These tools run on Java so can be easily integrated with other platforms. The community maintain the tools and develop plug-ins to help us make use of the ProM selection.

- **Apromore**[6]
  Apromore stands for Advanced Process Model Repository is designed as a business process analytics platform. It’s a robust platform with a wide range of features and functionality. In fact, Apromore boasts about its ability to lead the pack: For example, Apromore was the first tool to provide automated discovery of rich BPMN models, the

[6] *Advanced Process Analytics Platform*

comparison of BPMN models and event logs for conformance checking, the detection and characterization of process drifts from event logs, the replaying of event logs on top of BPMN models, the visual analysis of stage-based process performance, and the predictive monitoring of process performance.

- **Disco**
  Developed by academics, Disco is a comprehensive tool which has a broad array of features and a smoother user experience/user interface than the other open source tools, from my perspective. It seems a little more intuitive and resembles slightly some of the paid enterprise platforms in that regard.
  Disco advertises itself using a very catchy phrase: an x-ray for your processes. It sums up very nicely what process mining does and what its benefits are It also feels like a more honest description than the constant references to machine learning and AI which cover the websites of the paid tools.

In comparison, ProM has the wide variety of features and it is the best tools for process mining. Although ProM has the best solution, it has user interface problem. It is not easy to use as much as any other tools. That is why to use a software rather than ProM firstly and export the model into the ProM is the best way of the process mining. For example, Disco has very easy to use user interface and anybody easily understand how the process mining is evaluated.

3. Goals

As mentioned in the previous section, the main purpose of this Thesina is to become familiar with Process Mining and its power in Data Analysis and especially in Business Processes Analysis. There are many prerequisites for achieving this goal. After getting acquainted with concepts like Business Process Modeling, Business Process Modeling and Notation, Business Process Mining, and considering the importance of each in relation to Data Analysis, Big Data, etc., it's time to take advantage of this knowledge in practice Let's take a look at its ability in today's technology world.

To achieve this, I have created a tool (Log File Generator) that can randomly generate Log Files according to the requirements and parameters chosen by user by receiving a PM described in BPMN and parsing it. Then by importing this Log File into one of the Process Mining tools (Disco) we can obtain a new PM with using the information received from the user. By comparing the initial PM and the PM obtained by using the Process Mining software and tools we realize the power of utilizing the Process Mining to reach the business analysis objectives.

In the following, I explain the process of getting the job done step by step and get to know the basic concepts needed to create this application.

The steps are done in the following order:
First, I need to develop a tool to create the Event logs that are randomly generated pursuant to user's requirements, I named it "Log File Generator" and I have used Java as programming language for that which will explain the reasons and benefits of its usage.

3.1 Event Log

An Event log records business events from process-aware information systems (PAIS) such as WFM (Workflow Management), ERP (Enterprise Resource Planning), SCM (Supply Chain Management) and CRM (Customer Relationship Management) systems. Typically, Event logs contain information about start and completion of activities, their ordering, resources which executed them and the process instance they belong to.

The data columns determine the analysis possibilities that we have later on and here is where the real process mining requirements come into play. we need to identify at least the following three elements: Case ID, Activity, and Timestamp.

- **Case ID**

A case is a specific instance of the process. What precisely the meaning of a case is depends on the domain of the process. For every event, we must know which case it refers to, so that the process mining tool can compare several executions of the process to one another. the case ID influences the process scope.in other words It determines where the process starts and where it ends.

- **Activity**

An activity forms one step in the process. There should be names for different process steps or status changes that were performed in the process.

- **Timestamp**

The third important prerequisite for process mining is to have a timestamp column that indicates when the activity took place. This is not only important for analyzing the timing behavior of the process but also to establish the order of the activities in the event log.

The first task of this tool LFG[7] is to receive a BPMN as an input that must be in XPDL format[8]

The LFG will then receive a surplus information from the user that will be used in subsequent steps to create a Log File. This information includes:

- For every activity (Task):
  - The ExpectedTaskDuration;
  - The MaxTaskDuration;
  - The Average duration;
  - The StDev of the duration with Gaussian distribution;

- For every OR/XOR split
  - The percentage of each arch (The most important point in this section is considering that the sum of all these percentages must be equal to 100).

In the next step, one of the important tasks of this application, which is to identify the Independent Execution Paths, will be revealed. As we know, each WF can have several different paths, which vary depending on the number and variety of activities and decisions. In other words, in this section we come across the general concept of workflow complexity as described in Requirement section.

The LFG is able to identify the independent execution paths in the BPM used and display them to the user with information about each of them (from start to end and activities in between) and then compute the probability of each independent execution path with considering the percentage of every IEP (we take this percentage from user in previous step) .

After getting acquainted with the concept of Workflow Complexity and using its techniques to identify and calculate the independent paths available in the desired PM, it is time to continue the process of creating a Log File.

At this point the desired number of tuples will be received and the log file will be created. The format used when saving the Log File is The Comma Separated Values Format (CSV).

After creating and saving the Log File in the format mentioned, it is time to use Disco for Process Mining.

And final step after getting the log file created as a Disco tool input is specifying the elements required for Process Mining then the final PM is created and we can get some interesting results by comparing the input PM and the PM created after the Process mining operation.

[7] *Log File Generator*

[8] *which is used as a file format for the Business process model*

3.2 Constraints

At first glance, perhaps the first limitation to consider in analyzing a PM is facing loop paths. In these cases, it would be much more difficult to identify independent routes. In this Thesina, the default is that none of the paths in the PM will have a Loop and each path will run only once.

The second limitation that is very important in analyzing a PM is the accuracy of a PM syntactically and semantically.

Traditionally, it is believed that the quality of the model has to be evaluated relative to the purpose of the model.

We define the process of process modeling (PPM) as the sequence of steps a modeler performs in order to translate his mental image of the process into a formal, explicit and mostly graphical process specification: the process model. Here there are a few characteristics of good Business Process Models:

- Relevant: Subdivide and describe the system in ways that support implementation
- Complete: Capture all the requirements
- Precise: Define components specifically enough to support implementation
- Sequential: Depict the order of system events
- Rich: Account for enough complexity

3.3 Requirements

3.3.1 XPDL

The Workflow Management Coalition, a global organization of individuals and groups engaged in managing business workflow, has been developing workflow specifications for many years. These specifications are designed for the developers of workflow software products to implement solutions that are consistent, complete, and interoperable with other systems. One of the most significant workflow specifications is XPDL. XPDL specification defines the process activities, how they are performed, and the sequence in which they occur. An XPDL package corresponds to a collection of Process and Collaboration Diagrams in BPMN and consists of a set of Process Definitions, package-level declarations and other package level constructs such as artifacts and message flow between processes.

As mentioned before, the need for the interoperation of business processes at the human level, in addition to the software level, can be solved with standardization of the Business Process Modeling Notation (BPMN) being developed by the Business Process Management Initiative (BPMI).

In 2002, BPMI and WfMC agreed to work together. As part of this agreement, the WfMC has accepted BPMN as a notation for XPDL. Structurally, BPMN and XPDL are very similar; both being flow-chart structures.

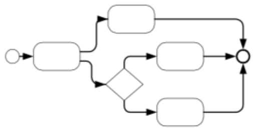
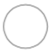
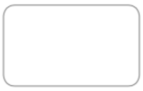
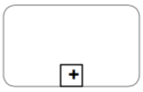
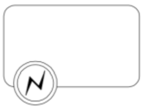
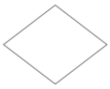
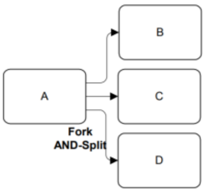
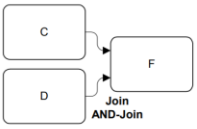
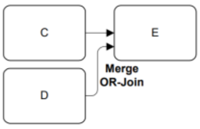
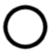
BPMN Graphical Object	***Mapping to XPDL***
 The details of a Pool or an Expanded Sub-Process	<WorkflowProcess/>
 Start Event	<Activity> <Route/> </Activity>
	<Transition/>
Sequence Flow	
 Task	<Activity> <Implementation> <Tool/> <Performer/> </Implementation> </Activities>
 Sub-Process	<Activity> <Implementation> <SubFlow/> </Implementation> </Activities>
 Intermediate Event attached to activity boundary	<Activity> <Implementation/> <TransitionRestriction> <Split Type="XOR"/> </TransitionRestriction> </Activities> Combined with a: <Transition> <Condition Type="EXCEPTION"/> <Transition>
 Decision	<Activity> <Route/> <TransitionRestriction> <Split Type="XOR"/> </TransitionRestriction> </Activities> Combined with a: <Transition> <Condition/> <Transition>
	<Activity> <Implementation/> <TransitionRestriction> <Split Type="AND"/> </TransitionRestriction> </Activities>
	<Activity> <Implementation/> <TransitionRestriction> <Join Type="AND"/> </TransitionRestriction> </Activities>
	<Activity> <Implementation/> <TransitionRestriction> <Join Type="XOR"/> </TransitionRestriction> </Activities>
 End Event	<Activity> <Route/> </Activity>

Figure 6 BPMN Objects and their Mappings to XPDL

3.3.2 Workflow complexity

To study complex workflow and their complexity, one has first answer to the question “What is a complex system?” Several definitions and explanations as to the formal definition of complexity exist, but they all have some aspects in common. Several researchers have centered their efforts on characterizing and quantifying the difficulty associated with complexity. Indeed, in some cases complexity is defined as the degree of disorder, while in other cases, it is the minimum length of the description of a system or the amount of resource (i.e., time or memory) needed to a system to solve a certain problem.

In the context of Business Process management (BPM) and Workflow Management Systems (WFM), applications and tools have moved from stand-alone office automation systems which augmented individuals, to networked and shared applications. The architectures based on workflow to support this new way of distributed working gave place to a rising complexity. To manage and control the complexity of workflows and kept it within an acceptable range, it is necessary to develop methods, algorithms and tools to measure their complexity.

the complexity of a system is strongly connected to its number of possible states and the information needed to describe the system. This definition is attractive, but when applied to workflows it is too restrictive. There are several complexity metrics can be devised based on the structure, parts and organization of workflows.

Workflow specifications can be understood from a number of different perspectives. four main complexity perspectives can be identified: activity complexity, control-flow complexity, data-flow complexity, and resource complexity.

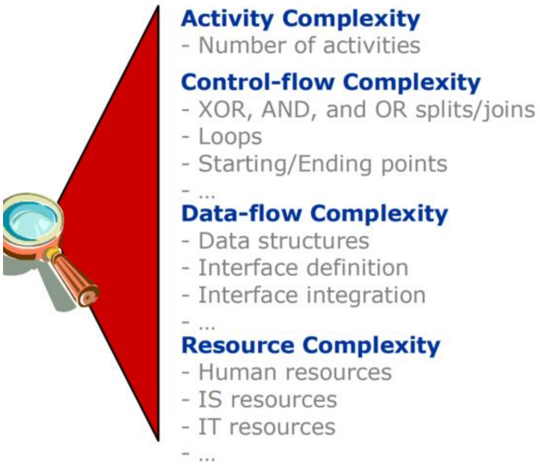


Figure 7 Perspective top workflow complexity

Since the main focus of this Thesin is on the second perspective, we will skip the rest of the discussion and continue to explore Control Flow Complexity.

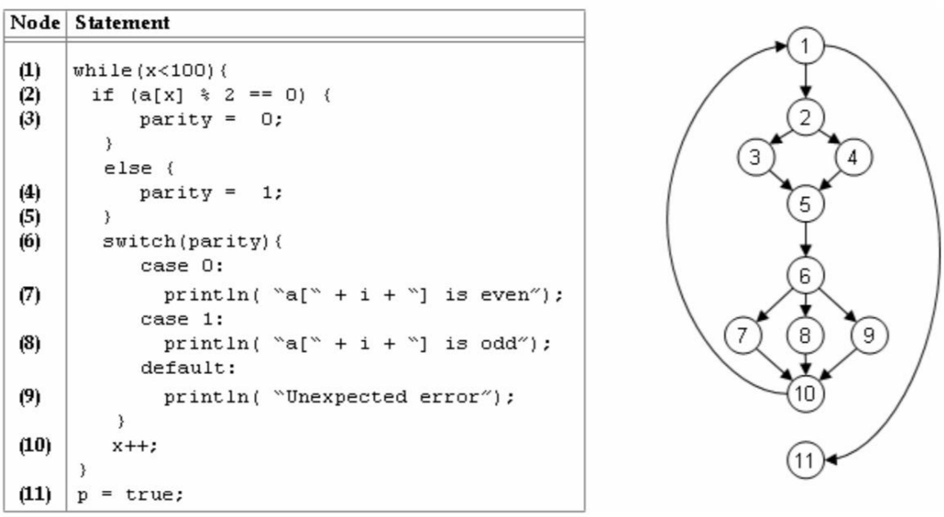
The control-flow perspective describes activities and their execution ordering through different constructors, which permit flow of execution control. Constructors include sequence, choice, parallelism, splits, joins, loops, and ending and starting points (Cardoso 2005). Splits allow defining the possible control paths that exist in a process. Joins have a different role; they express the type of synchronization that should be made at a specific point in the process. A control-flow complexity model needs to take into account the existence of XOR-split/join, OR-split/join, AND-split/join, loops, etc.

The overall goal of workflow complexity analysis is to improve the comprehensibility of workflows. Thus, it is important to develop methods and measurements to automatically identify complex workflows and complex areas of workflows and also it is necessary to developed multi-dimensional metrics of complexity and then validated the metrics through objective and subjective measurements.

There are several complexity considerations and metrics from several fields such as information complexity, cyclomatic complexity, Kolmogorov complexity, cognitive complexity, and computational complexity. I learnt about Cyclomatic complexity during Business Process Modeling Lessons by Prof.G.Pozzi, so I used it as my knowledge in this Thesina as well.

- **Cyclomatic complexity**:

The main idea of this matric comes from McCabe's cyclomatic complexity that its objective was to evaluate processes' complexity. One of the first important observations that can be made from the MCC control flow graph, shown in Figure bellow, is that this graph is extremely similar to a process. One major difference is that the nodes of a MCC control flow graph have identical semantics, while process nodes (i.e., activities) can have different semantics (e.g., AND-splits, XOR-splits, OR-joins, etc.).



```
Node  Statement

(1)   while(x<100){
(2)     if (a[x] % 2 == 0) {
(3)         parity =  0;
        }
        else {
(4)         parity =  1;
(5)     }
(6)     switch(parity){
            case 0:
(7)           println( "a[" + i + "] is even");
            case 1:
(8)           println( "a[" + i + "] is odd");
            default:
(9)           println( "Unexpected error");
        }
(10)    x++;
      }
(11)  p = true;
```

Figure 8 MCC Control Flow Graph

The metric called Control-flow Complexity (CFC) metric, is based on the analysis of XOR-splits, OR-splits, and AND splits control-flow elements. The main idea behind the metric is to evaluate the number of mental states that have to be considered when a designer is developing a process. Splits introduce the notion of mental states in processes. When a split (XOR, OR, or AND) is introduced in a process, the business process designer has to mentally create a map or structure that accounts for the number of states that can be reached from the split. Mathematically, the control-flow complexity metric is additive, thus it is very easy to calculate the complexity of a process, by simply adding the CFC of all split constructs. The control-flow complexity was calculated as follows, where "P" is a process and "a" an activity.

$$CFC(P) = \sum_{a \in P, a \text{ isa } xor-split} CFC_{XOR}(a) + \sum_{a \in P, a \text{ isa } or-split} CFC_{OR}(a) + \sum_{a \in P, a \text{ isa } and-split} CFC_{AND}(a)$$

Figure 9 Control Flow Complexity Calculation

The higher the value of CFCXOR(a), CFCOR(a), and CFCAND(a), the more complex is a process design, since developer has to handle all the states between control-flow constructs (splits) and their associated outgoing transitions and activities. Each formula to calculate the complexity of a split construct is based on the number of states that follow the construct. CFC analysis seeks to evaluate complexity without direct execution of processes. The advantages of the CFC metric are that it can be used as a maintenance and quality metric, it gives the relative complexity of process designs, and it is easy to apply. Disadvantages of the CFC metric include the inability to measure data complexity, only control-flow complexity is measured.

3.3.3 Comma Separated Values format (CSV)

A comma separated values (CSV) file contains different values separated by a delimiter, which acts as a database table or an intermediate form of a database table. In other words, a CSV file is a set of database rows and columns stored in a text file such that the rows are separated by a new line while the columns are separated by a semicolon or a comma. A CSV file is primarily used to transport data between two databases of different formats through a computer program.
The advantage of using CSV file format for data exchange is that the CSV file is relatively easy to process by any application and data extraction can be achieved with the help of a simple program. In the earlier years when database technologies were still in their infancy, the CSV was the most standard portable format.
And last but not least CSV file format is the one that can be used in Process Mining tools and in this Thesina this tool is Disco.

3.3.4 Gaussian Distribution

The Gaussian distribution, normal distribution, or bell curve is a probability distribution which accurately models a large number of phenomena in the world.
Basically, it is the mathematical representation of how a large number of items follow the Central Limit Theory (CLT). The CLT says that, under mild conditions, the (normalized) sum of random values will tend to a gaussian distribution as the number of values in the sum increases. A Gaussian distribution can describe many examples of real-world data such as the ground state of a quantum harmonic oscillator or the distribution of demographic characteristics in populations.
Data can be "distributed" (spread out) in different ways.

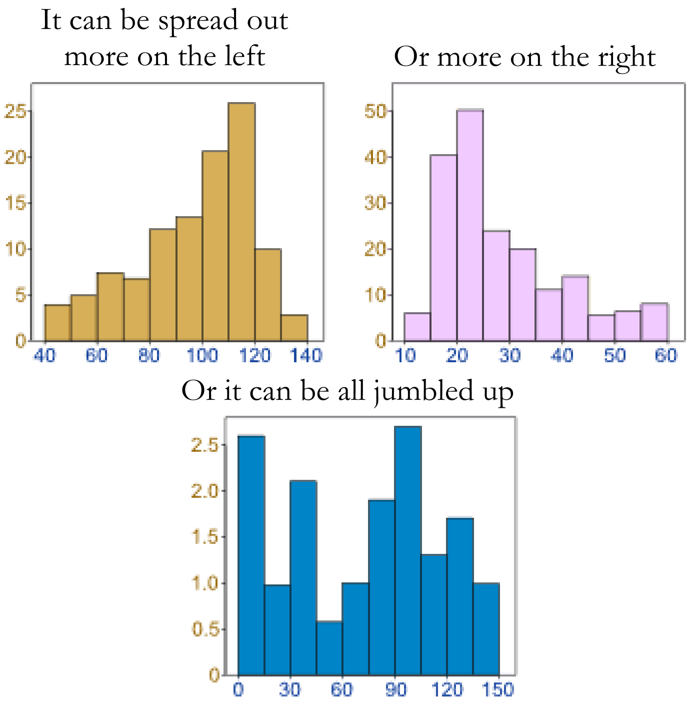


Figure 10 Data Distribution Types

But there are many cases where the data tends to be around a central value with no bias left or right, and it gets close to a "Normal Distribution" like this:

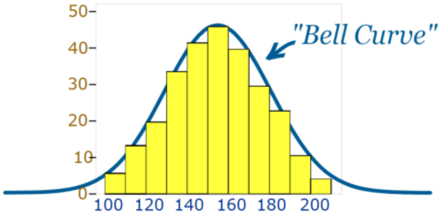


Figure 11 Bell Curve Histogram

The "Bell Curve" is a Normal Distribution And the yellow histogram shows some data that follows it closely, but not perfectly (which is usual). In Bell Curve We say the data is normally distributed.

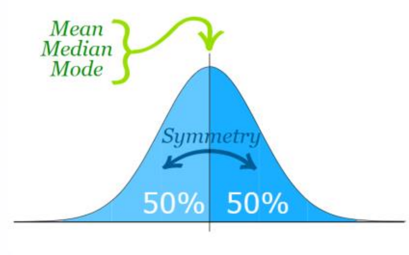


Figure 12 Normal Distribution

The Normal Distribution has:

- mean = median = mode
- symmetry about the center
- 50% of values less than the mean
- and 50% greater than the mean

Standard deviation is a widely used measurement of variability or diversity used in statistics and probability theory (99% of cases within average +-3*StDev). It shows how much variation or "dispersion" there is from the "average" (mean or expected value). A low standard deviation indicates that the data points tend to be very close to the mean, whereas high standard deviation indicates that the data is spread out over a large range of values.

The graph of the Standard distribution has the shape of a bell curve, as shown in Figure below:

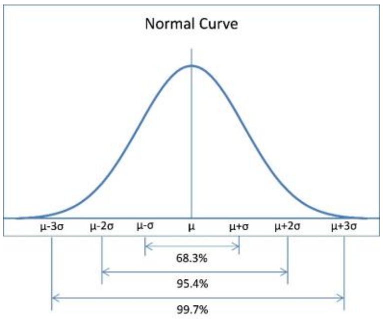


Figure 13 Standard Distribution Graph

Some characteristics of the Normal distribution are as follows:

- Mean = median = mode = μ
- Standard deviation = σ
- Skewness = kurtosis = 0

4. System Description

4.1 Tools selection

4.1.1 Java

In this Thesina I have used Java as programming language for developing my application (Log File Generator) because during my Bachelor studying, I became familiar to C++ and Java has the look and feel of the C++ programming language, but it is simpler to use than C++ and enforces an object-oriented programming model.

Java is a programming language, designed to be concurrent, class-based and object-oriented, as well as a computing platform first released by Sun Microsystems in 1995.

An important reason for using Java in this Thesina is its platform independence or multiplatform support. Java programs are able to execute on different machines as long as there is a JRE (Java Runtime Environment) in place. Be it mobile phones, PCs running Linux, macOS, or Windows, and even large mainframe computers, JRE is compatible with all of them.

A big part of the Java ecosystem is the large variety of open source and community built projects, software platforms and APIs.

Java has been tested, refined, extended, and proven by a dedicated community of Java developers, architects and enthusiasts. Despite origins dating back almost two decades, Java has consistently evolved over the years.

Java is designed to enable development of portable, high-performance applications for the widest range of computing platforms possible, hence enabling the fundamental tenets of overarching accessibility as well as cross-platform interaction.

Java has become invaluable to developers by enabling them to:

- Write software on one platform and run it on virtually any other platform.
- Create programs that can run within a web browser and access available web services.
- Develop server-side applications for online forums, stores, polls, HTML forms processing, and more.
- Combine applications or services using the Java language to create highly customized applications or services.
- Write powerful and efficient applications for mobile phones, remote processors, microcontrollers, wireless modules, sensors, gateways, consumer products, and practically any other electronic device.

4.1.2 NetBeans

NetBeans is an open-source integrated development environment (IDE) for developing with Java, PHP, C++, and other programming languages. NetBeans is also referred to as a platform of modular components used for developing Java desktop applications.

NetBeans uses components, also known as modules, to enable software development. NetBeans dynamically installs modules and allows users to download updated features and digitally authenticated upgrades.

NetBeans IDE modules include NetBeans Profiler, a Graphical User Interface (GUI) design tool, and NetBeans JavaScript Editor.

NetBeans framework reusability simplifies Java Swing desktop application development, which provides platform extension capabilities to third-party developers.

4.1.3 Together Workflow Editor

Together Workflow Editor is a professional and open source graphical editor developed to create, edit, manage and review WfMC (Workflow Management Coalition) XPDL (XML Process Definition Language) process definition files.

The program provides quick access to a graph overview, external package relations, transient package pool components, as well as XPDL and LDAP components.

we can view the whole graph of the selected process or activity set, graphical tree overview of the current package and all of its referenced external packages, and XPDL text view of the selected elements.

In this Thesina Together Workflow Editor is used to show the XPDL file as a graph that is import to Log File Generator. By that we can compare the Process model before and after using the Application. (LFG)

4.1.4 Disco

The core functionality of process mining is the automated discovery of process maps by interpreting the sequences of activities in the imported log file. Disco is a complete process mining toolkit from Fluxicon that makes process mining fast, simple and easy to understand.

Every process mining project starts with the data that should be analyzed. Disco has been designed to make the data import really easy by automatically detecting timestamps, remembering your configuration settings, and by loading data sets with high speed.

One simply opens a CSV or Excel file and configures which columns hold the case ID, timestamps, activity names, which other attributes should be included in the analysis, and the import can be started.

The other advantage of Disco is that it supports project work through the management of multiple data sets in one project view. In a typical process mining project, one will import log files in different ways, filter them, and make copies to save intermediate results. This results in many different versions and views of the data sets and can easily get out of hand.

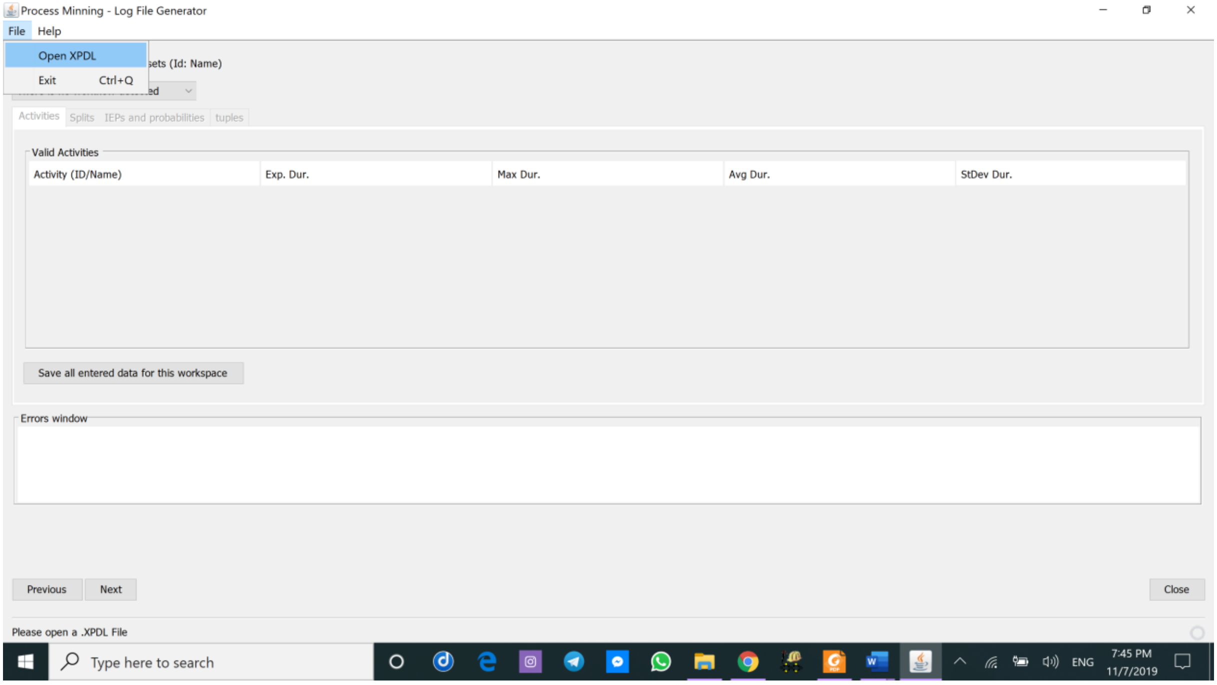
After generating Log file by Log File Generator App in .CSV format, Using Disco as a powerful Process mining tool was my best decision to achieve the best results for this Thesina.

4.2 The system

Here this is the Log File Generator Application. For the first step after running the App it is necessary to open a XPDL from File Menu. Considering that a XPDL file is an entry of an XML file, in order to recognize all the available workflows, we must search for the tags with the title of Workflow Process or Activity Set and we save them in a set naming workflow. Then for each workflow we can extract other tags with such names as Split, Activity and Transition.

It's worth mentioning for each Transition, the information related to "To, From, ID" and the field naming "Percentage" with the initial value of zero (in order to show the Transition probability) must be saved. Later on, the value will be given to this field by the user.

Hereinafter, for each activity in workflow the information for "ID, Name" must be generated and saved from the XPDL file.

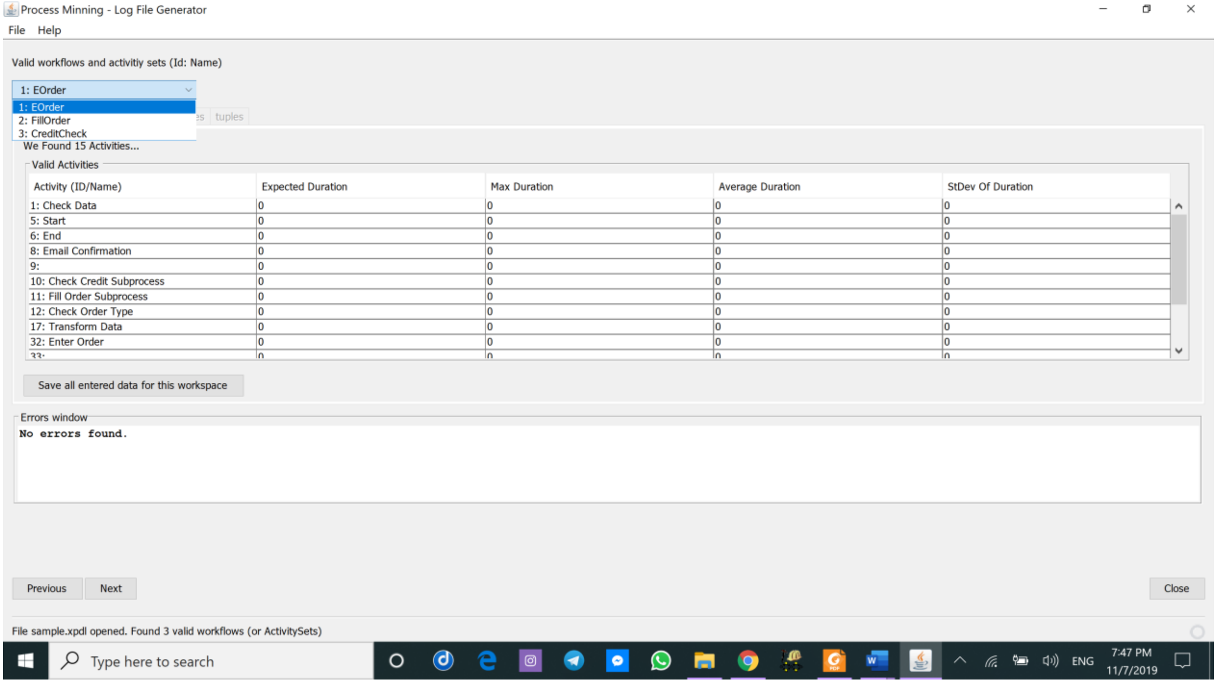


Activity (ID/Name)	Exp. Dur.	Max Dur.	Avg Dur.	StDev Dur.

Please open a .XPDL File

Figure 14 Log File Generator Application – Opening XPDL

After opening the XPDL, we can see all the valid Workflows and Activities that exist in our Business Process Model. All the recognized Workflows after executing the required levels will be seen in a Combo Box which by selecting each Activity that is available in that Workflow will be demonstrated to the user as the table below.



We Found 15 Activities...

Valid Activities

Activity (ID/Name)	Expected Duration	Max Duration	Average Duration	StDev Of Duration
1: Check Data	0	0	0	0
5: Start	0	0	0	0
6: End	0	0	0	0
8: Email Confirmation	0	0	0	0
9:	0	0	0	0
10: Check Credit Subprocess	0	0	0	0
11: Fill Order Subprocess	0	0	0	0
12: Check Order Type	0	0	0	0
17: Transform Data	0	0	0	0
32: Enter Order	0	0	0	0

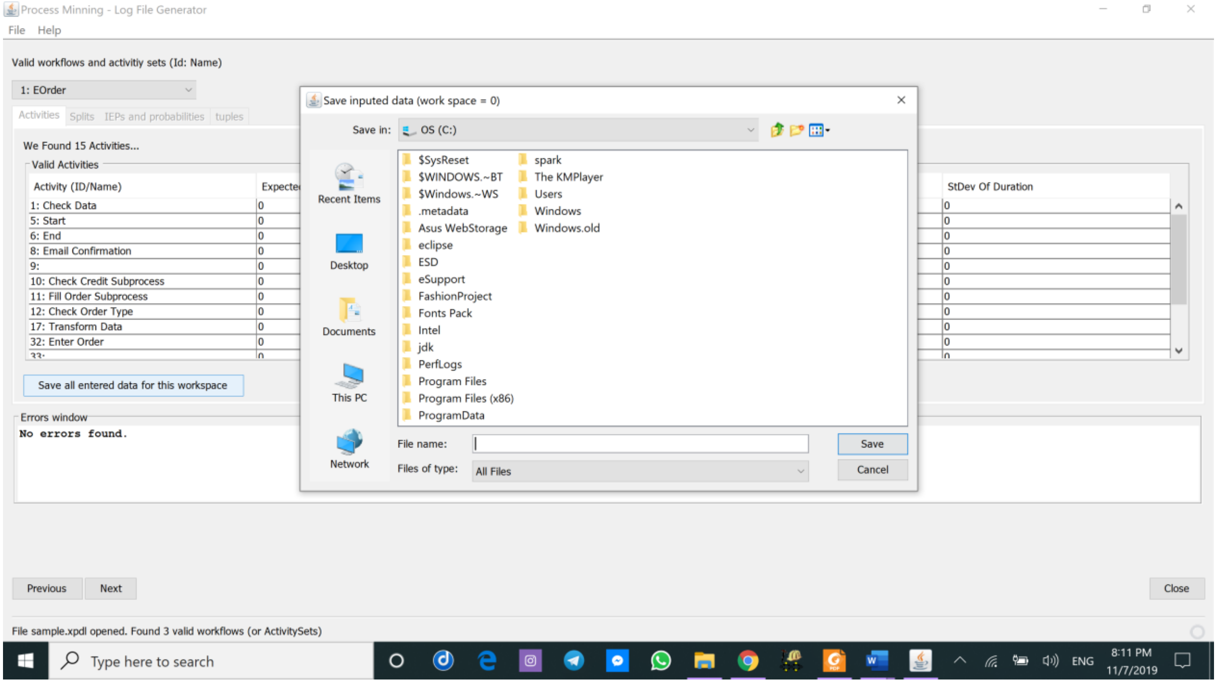
```
No errors found.
```

File sample.xpdl opened. Found 3 valid workflows (or ActivitySets)

Figure 15 Log File Generator Application – Selecting Workflow

In the beginning, the "Expected Duration, Max Duration, Average Duration, Standard Dev" have default value of zero and in the next levels the value must be given to them by the user. At this phase as it is shown in the figure, the user is able to enter data for each Activity and in the case of not entering the data by the user, the zero value will be considered.

In addition, it's possible to save the entered values by the user in a CSV file which gives the user the possibility of having the access to the input data at any time (even with the closing of the application/program). The "save all entered Data for this workspace" button in intended for this purpose.



We Found 15 Activities...

Valid Activities

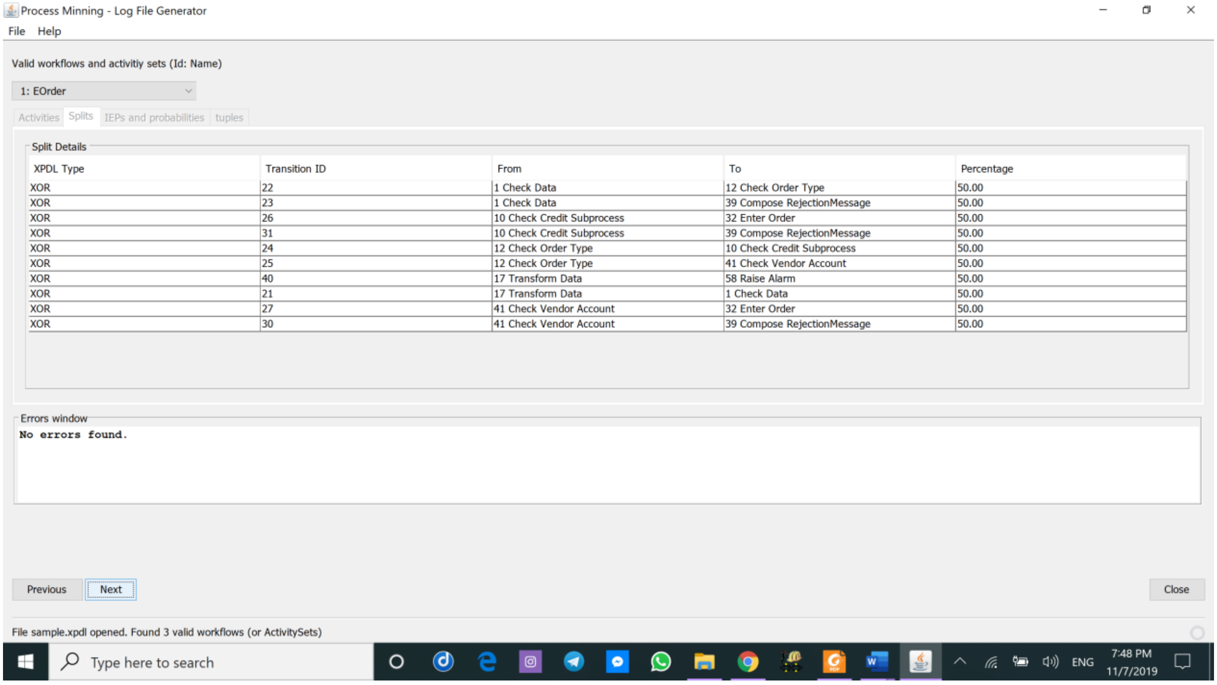
Activity (ID/Name)	Expecte	StDev Of Duration
1: Check Data	0	0
5: Start	0	0
6: End	0	0
8: Email Confirmation	0	0
9:	0	0
10: Check Credit Subprocess	0	0
11: Fill Order Subprocess	0	0
12: Check Order Type	0	0
17: Transform Data	0	0
32: Enter Order	0	0

```
No errors found.
```

File sample.xpdl opened. Found 3 valid workflows (or ActivitySets)

Figure 16 Log File Generator Application – Activities

At this point, among all the known "Splits" from the previous stage, only the OR and XOR splits will be filtered, which for each of the two Branch there will be an equal possibility of (%50) and this means their probability will be identical. Only under one condition it is possible to change the probability of a Branch: The sum of the probability of Branches of each Split must be 100%, otherwise, the "the summation of percentage is invalid" message will be shown and the user is not able to go forward to the next level.



Valid workflows and activitiy sets (Id: Name)

1: EOrder

Split Details

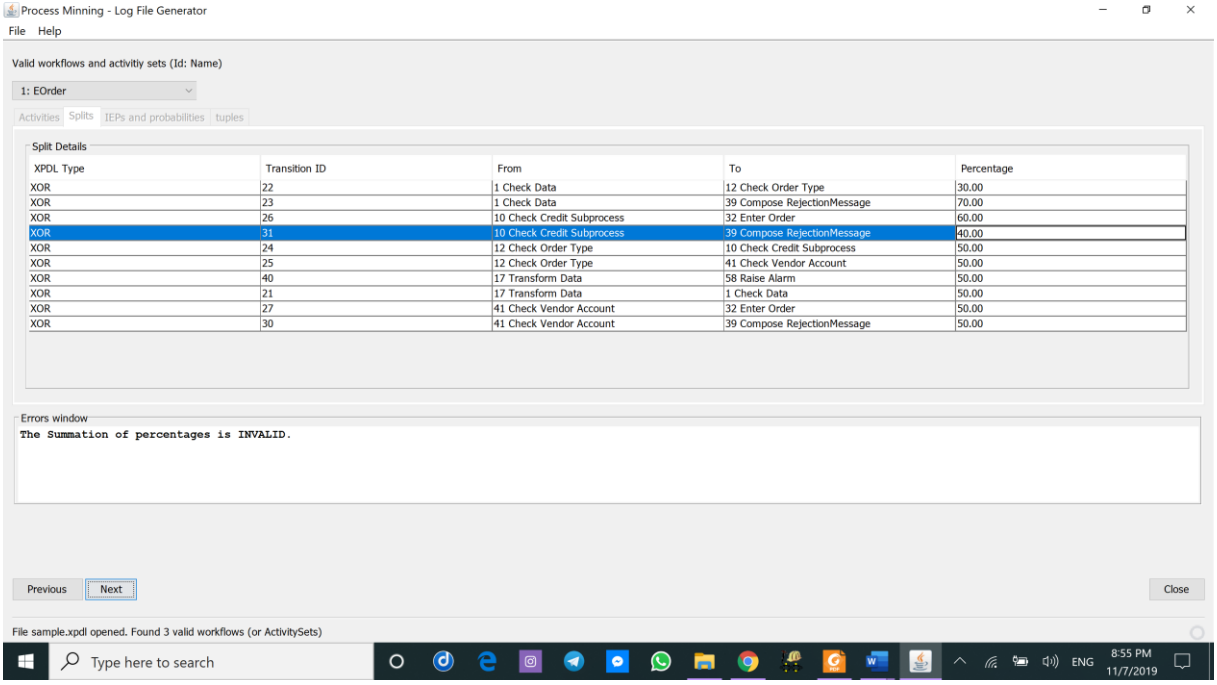
XPDL Type	Transition ID	From	To	Percentage
XOR	22	1 Check Data	12 Check Order Type	50.00
XOR	23	1 Check Data	39 Compose RejectionMessage	50.00
XOR	26	10 Check Credit Subprocess	32 Enter Order	50.00
XOR	31	10 Check Credit Subprocess	39 Compose RejectionMessage	50.00
XOR	24	12 Check Order Type	10 Check Credit Subprocess	50.00
XOR	25	12 Check Order Type	41 Check Vendor Account	50.00
XOR	40	17 Transform Data	58 Raise Alarm	50.00
XOR	21	17 Transform Data	1 Check Data	50.00
XOR	27	41 Check Vendor Account	32 Enter Order	50.00
XOR	30	41 Check Vendor Account	39 Compose RejectionMessage	50.00

Errors window

```
No errors found.
```

File sample.xpdl opened. Found 3 valid workflows (or ActivitySets)

Figure 17 Log File Generator Application – Splits



Valid workflows and activitiy sets (Id: Name)

1: EOrder

Split Details

XPDL Type	Transition ID	From	To	Percentage
XOR	22	1 Check Data	12 Check Order Type	30.00
XOR	23	1 Check Data	39 Compose RejectionMessage	70.00
XOR	26	10 Check Credit Subprocess	32 Enter Order	60.00
XOR	31	10 Check Credit Subprocess	39 Compose RejectionMessage	40.00
XOR	24	12 Check Order Type	10 Check Credit Subprocess	50.00
XOR	25	12 Check Order Type	41 Check Vendor Account	50.00
XOR	40	17 Transform Data	58 Raise Alarm	50.00
XOR	21	17 Transform Data	1 Check Data	50.00
XOR	27	41 Check Vendor Account	32 Enter Order	50.00
XOR	30	41 Check Vendor Account	39 Compose RejectionMessage	50.00

Errors window

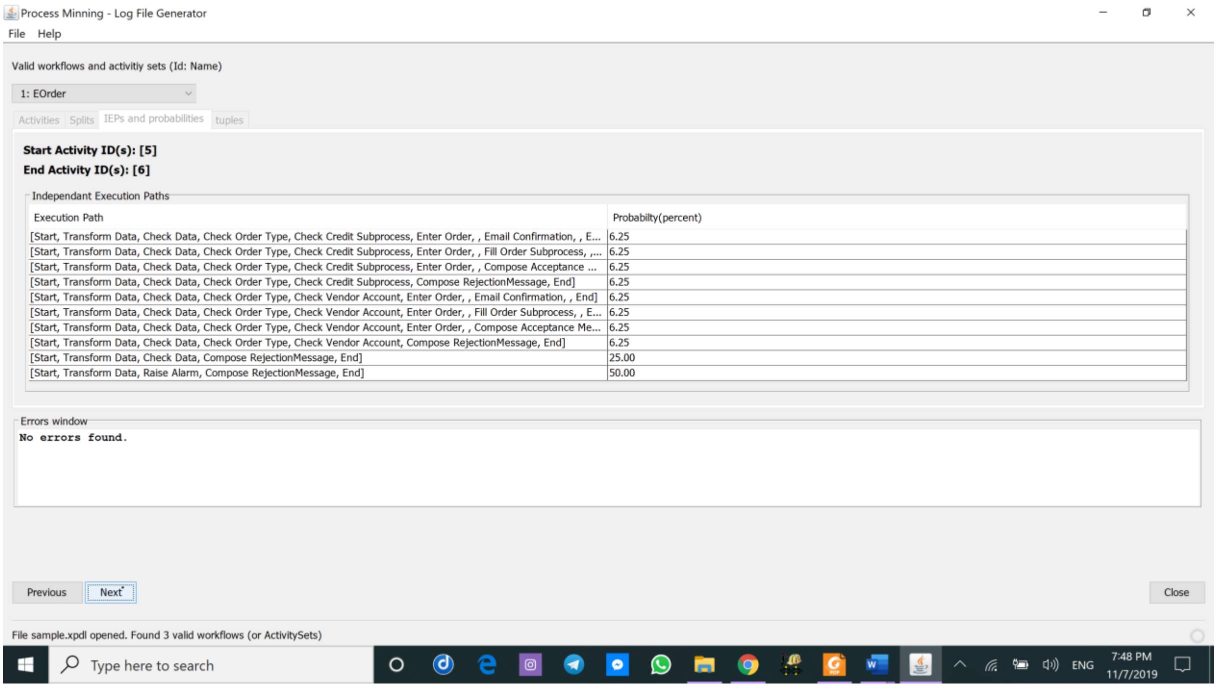
```
The Summation of percentages is INVALID.
```

File sample.xpdl opened. Found 3 valid workflows (or ActivitySets)

Figure 18 Log File Generator Application – Splits/Percentages

In this stage all the possible paths from the starting Activity to the end Activity will be identified. It must be mentioned that all those Activity without inputs, will be considered as a starting Activity. In another word, in none of the Transition the ID of the Activity must not be in the To feature of that Transition, also for the ending Activity the ID of that Activity must not be shown in the From feature.

In order to calculate the probability of each path (the default probability of 100 is considered) from the start Activity to the end Activity, we go forward Transition by Transition. If in the path of a Transition, its percentage value is not zero, it means that it is a Transition of one of the Branches of the Split and its probability must be multiplied by the current value. At the end when the Activity is finished, the probability of this path is either %100 which means in this path, there were no Split, or if the value is lower than %100 it means in this path there were several Split and each of the probabilities will be calculated on the basis of the user's input.



Valid workflows and activitiy sets (Id: Name)

1: EOrder

Start Activity ID(s): [5]

End Activity ID(s): [6]

Independant Execution Paths

Execution Path	Probabilty(percent)
[Start, Transform Data, Check Data, Check Order Type, Check Credit Subprocess, Enter Order, , Email Confirmation, , E...	6.25
[Start, Transform Data, Check Data, Check Order Type, Check Credit Subprocess, Enter Order, , Fill Order Subprocess, ,...	6.25
[Start, Transform Data, Check Data, Check Order Type, Check Credit Subprocess, Enter Order, , Compose Acceptance ...	6.25
[Start, Transform Data, Check Data, Check Order Type, Check Credit Subprocess, Compose RejectionMessage, End]	6.25
[Start, Transform Data, Check Data, Check Order Type, Check Vendor Account, Enter Order, , Email Confirmation, , End]	6.25
[Start, Transform Data, Check Data, Check Order Type, Check Vendor Account, Enter Order, , Fill Order Subprocess, , E...	6.25
[Start, Transform Data, Check Data, Check Order Type, Check Vendor Account, Enter Order, , Compose Acceptance Me...	6.25
[Start, Transform Data, Check Data, Check Order Type, Check Vendor Account, Compose RejectionMessage, End]	6.25
[Start, Transform Data, Check Data, Compose RejectionMessage, End]	25.00
[Start, Transform Data, Raise Alarm, Compose RejectionMessage, End]	50.00

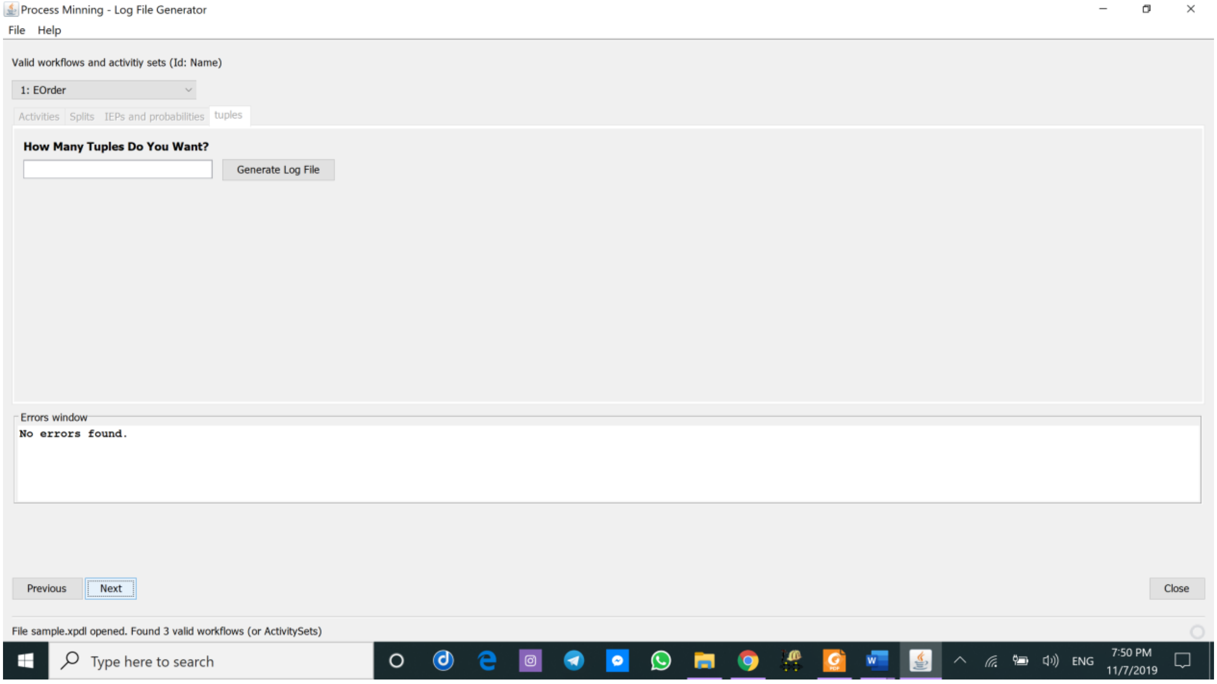
Errors window

```
No errors found.
```

File sample.xpdl opened. Found 3 valid workflows (or ActivitySets)

Figure 19 Log File Generator Application – Independent Execution Paths

In the last step of the App, the user will enter a number as number of Tuples in Log File. Log File will be created as a CSV file and it will include the fields such as "Case ID, Activity, Start Time, End Time". The start and finish time are calculated based on the Average Duration that is entered in the first stage.


Process Minning - Log File Generator

Valid workflows and activitiy sets (Id: Name)

1: EOrder

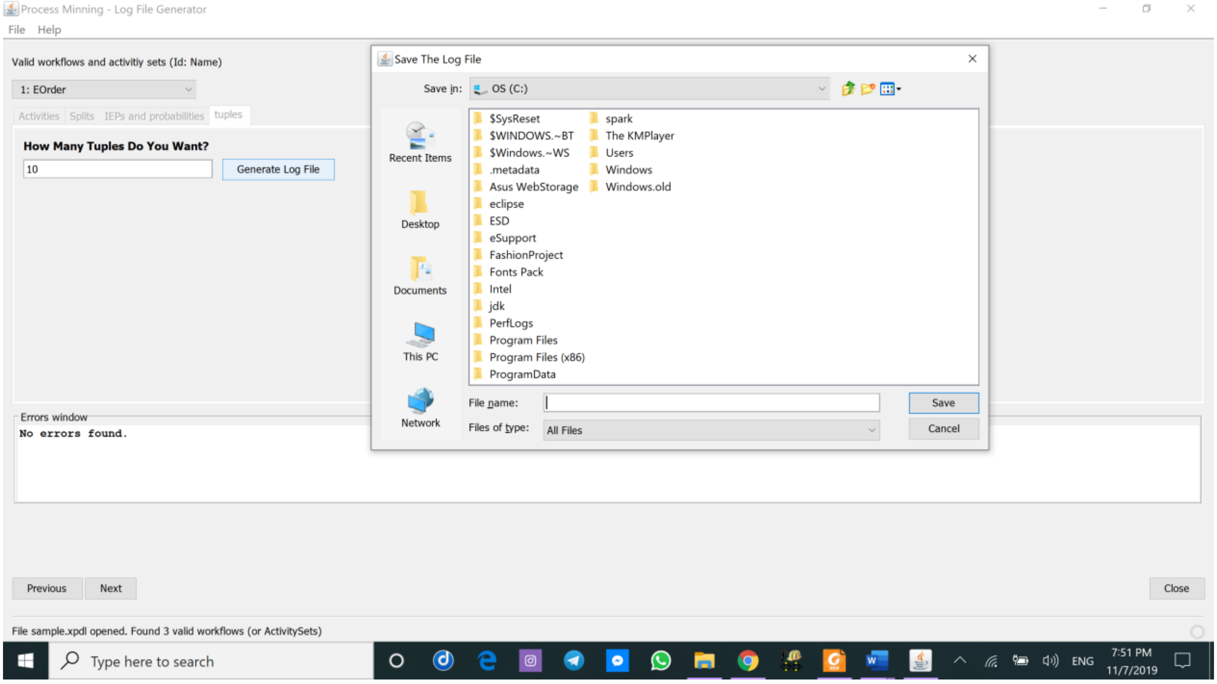
How Many Tuples Do You Want?

Errors window

```
No errors found.
```

File sample.xpdl opened. Found 3 valid workflows (or ActivitySets)

Figure 20 Log File Generator Application – Tuples


Process Minning - Log File Generator

Valid workflows and activitiy sets (Id: Name)

1: EOrder

How Many Tuples Do You Want?

10

Save The Log File

Save in: OS (C:)

$SysReset
$WINDOWS.~BT
$Windows.~WS
.metadata
Asus WebStorage
eclipse
ESD
eSupport
FashionProject
Fonts Pack
Intel
jdk
PerfLogs
Program Files
Program Files (x86)
ProgramData
spark
The KMPlayer
Users
Windows
Windows.old

File name:

Files of type: All Files

Errors window

```
No errors found.
```

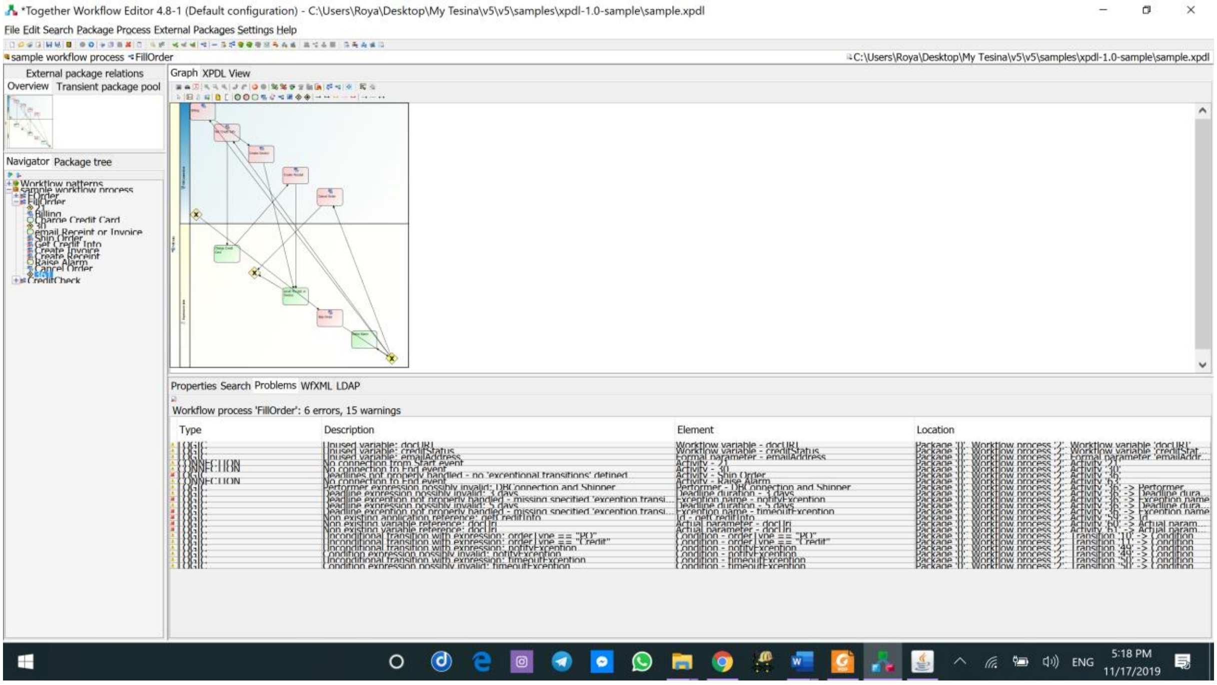
File sample.xpdl opened. Found 3 valid workflows (or ActivitySets)

Figure 21 Log File Generator Application – Generating Log File in Chosen Path

5. Experiments

5.1 Running an Experiment

For using the App LFG, we should begin with a sample BPMN in XPDL format. For having a good comparison at the end, it is better to see the PM graph. For having this graph, I have used together workflow editor tool which expressed before. Using TWE is so simple. It needs to open a XPDL file from File Menu as shown in figure below.



*Together Workflow Editor 4.8-1 (Default configuration) - C:\Users\Roya\Desktop\My Tesina\v5\v5\samples\xpdl-1.0-sample\sample.xpdl

Workflow process 'FillOrder': 6 errors, 15 warnings

Type	Description	Element	Location
LOGIC	Unused variable: docURL	Workflow variable - docURL	Package '1', Workflow process '2', Workflow variable 'docURL'
LOGIC	Unused variable: creditStatus	Workflow variable - creditStatus	Package '1', Workflow process '2', Workflow variable 'creditStat...
LOGIC	Unused variable: emailAddress	Formal parameter - emailAddress	Package '1', Workflow process '2', Formal parameter 'emailAddr...
CONNECTION	No connection from Start event	Activity - 21	Package '1', Workflow process '2', Activity '21'
CONNECTION	No connection to End event	Activity - 30	Package '1', Workflow process '2', Activity '30'
LOGIC	Deadlines not properly handled - no 'exceptional transitions' defined	Activity - Ship Order	Package '1', Workflow process '2', Activity '36'
CONNECTION	No connection to End event	Activity - Raise Alarm	Package '1', Workflow process '2', Activity '63'
LOGIC	Performer expression possibly invalid: DBConnection and Shipper	Performer - DBConnection and Shipper	Package '1', Workflow process '2', Activity '36' -> Performer
LOGIC	Deadline expression possibly invalid: 3 days	Deadline duration - 3 days	Package '1', Workflow process '2', Activity '36' -> Deadline dura...
LOGIC	Deadline exception not properly handled - missing specified 'exception transi...	Exception name - notifyException	Package '1', Workflow process '2', Activity '36' -> Exception name
LOGIC	Deadline expression possibly invalid: 5 days	Deadline duration - 5 days	Package '1', Workflow process '2', Activity '36' -> Deadline dura...
LOGIC	Deadline exception not properly handled - missing specified 'exception transi...	Exception name - timeoutException	Package '1', Workflow process '2', Activity '36' -> Exception name
LOGIC	Non existing application reference: getCreditInfo	Id - getCreditInfo	Package '1', Workflow process '2', Activity '59' -> Id
LOGIC	Non existing variable reference: docUrl	Actual parameter - docUrl	Package '1', Workflow process '2', Activity '60' -> Actual param...
LOGIC	Non existing variable reference: docUrl	Actual parameter - docUrl	Package '1', Workflow process '2', Activity '61' -> Actual param...
LOGIC	Unconditional transition with expression: orderType == "PO"	Condition - orderType == "PO"	Package '1', Workflow process '2', Transition '10' -> Condition
LOGIC	Unconditional transition with expression: orderType == "Credit"	Condition - orderType == "Credit"	Package '1', Workflow process '2', Transition '11' -> Condition
LOGIC	Unconditional transition with expression: notifyException	Condition - notifyException	Package '1', Workflow process '2', Transition '49' -> Condition
LOGIC	Condition expression possibly invalid: notifyException	Condition - notifyException	Package '1', Workflow process '2', Transition '49' -> Condition
LOGIC	Unconditional transition with expression: timeoutException	Condition - timeoutException	Package '1', Workflow process '2', Transition '50' -> Condition
LOGIC	Condition expression possibly invalid: timeoutException	Condition - timeoutException	Package '1', Workflow process '2', Transition '50' -> Condition

Figure 22 Together Workflow Editor

As we can see in this sample, we have three workflow processes: “EOrder, FillOrder and Creditcheck”. By selecting each of them we can see its graph with other related properties for each. I have selected “FillOrder” workflow in this figure.

Recognizing Activities, Paths, ... is very difficult here and this is the main point of Process mining.

In the figure below I have saved the “FillOrder” workflow’s graph in PDF format for having a better view on details.

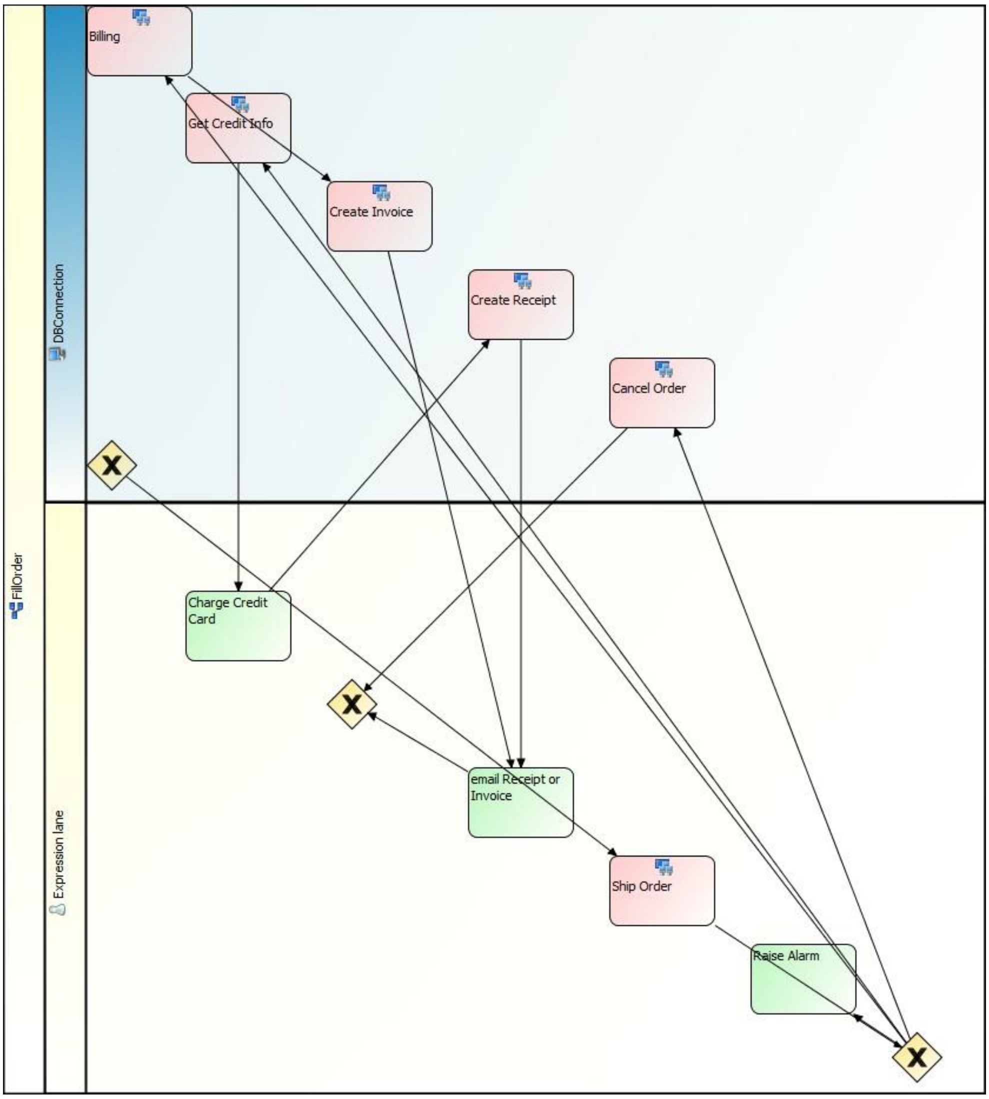


Figure 23 Together Workflow Editor – FillOrder

And here is the List of Activities, their Names and IDs that TWE shows.

ID 21
Name Start

Incoming transitions
← 361 orderType == "PO"
Outgoing transitions
→ 60

ID 22
Name Billing

Incoming transitions
← 59
Outgoing transitions
→ 61
WebService

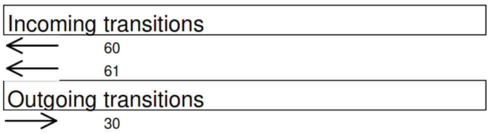
ID 23
Name Charge Credit Card

Incoming transitions
← 31
← 64

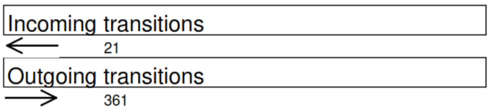
ID 30
Name End

Incoming transitions
← 60
← 61
Outgoing transitions
→ 30

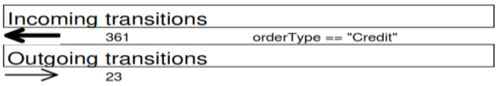
ID	31
Name	Email Receipt or Invoice



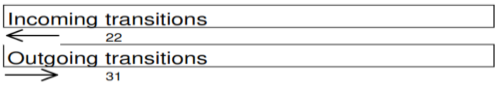
ID	36
Name	Ship Order



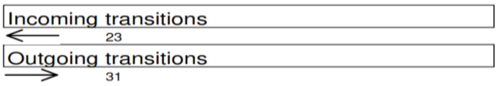
ID	59
Name	Get Credit Info



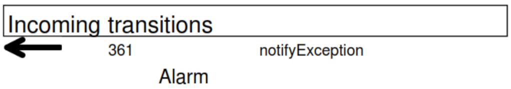
ID	60
Name	Create Invoice



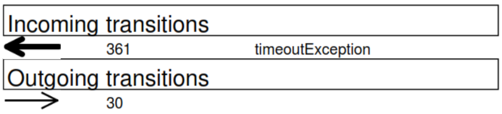
ID	61
Name	Create Receipt



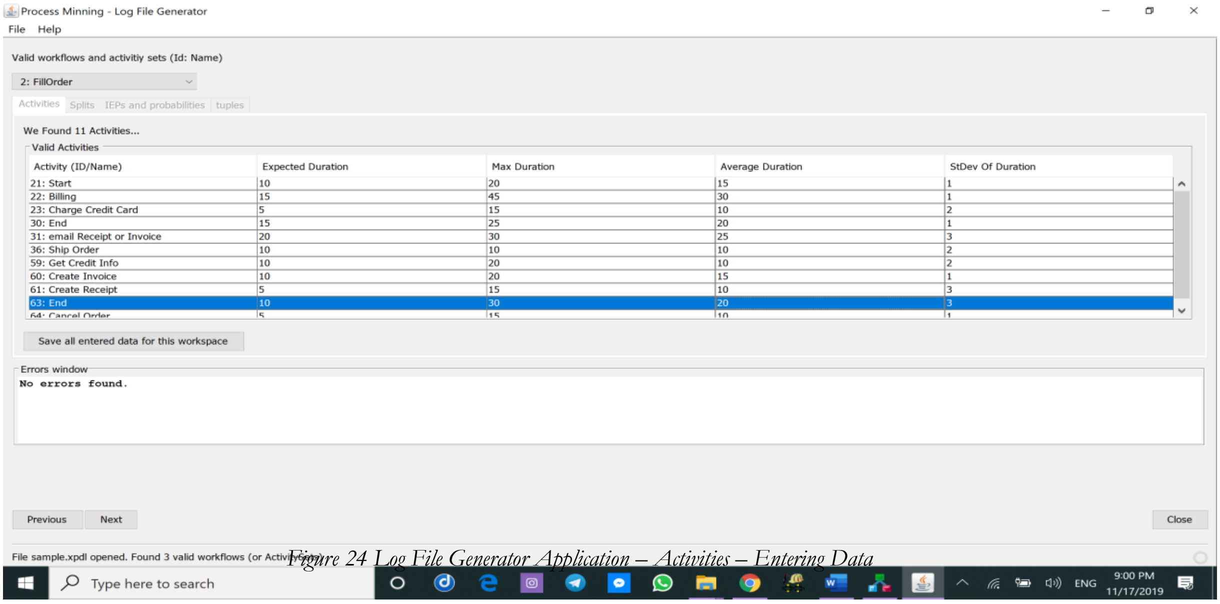
ID	63
Name	End



ID	64
Name	Cancel Order



Based on this info we should see these Activities in the App after opening XPDL file. As mentioned before in section 4 (System) as soon as running LFG App and opening a XPDL file from File Menu, LFG App recognises all the Activities and their Names and IDs and show it to user.



Valid workflows and activitiy sets (Id: Name)

2: FillOrder

We Found 11 Activities...

Valid Activities

Activity (ID/Name)	Expected Duration	Max Duration	Average Duration	StDev Of Duration
21: Start	10	20	15	1
22: Billing	15	45	30	1
23: Charge Credit Card	5	15	10	2
30: End	15	25	20	1
31: email Receipt or Invoice	20	30	25	3
36: Ship Order	10	10	10	2
59: Get Credit Info	10	20	10	2
60: Create Invoice	10	20	15	1
61: Create Receipt	5	15	10	3
63: End	10	30	20	3
64: Cancel Order	5	15	10	1

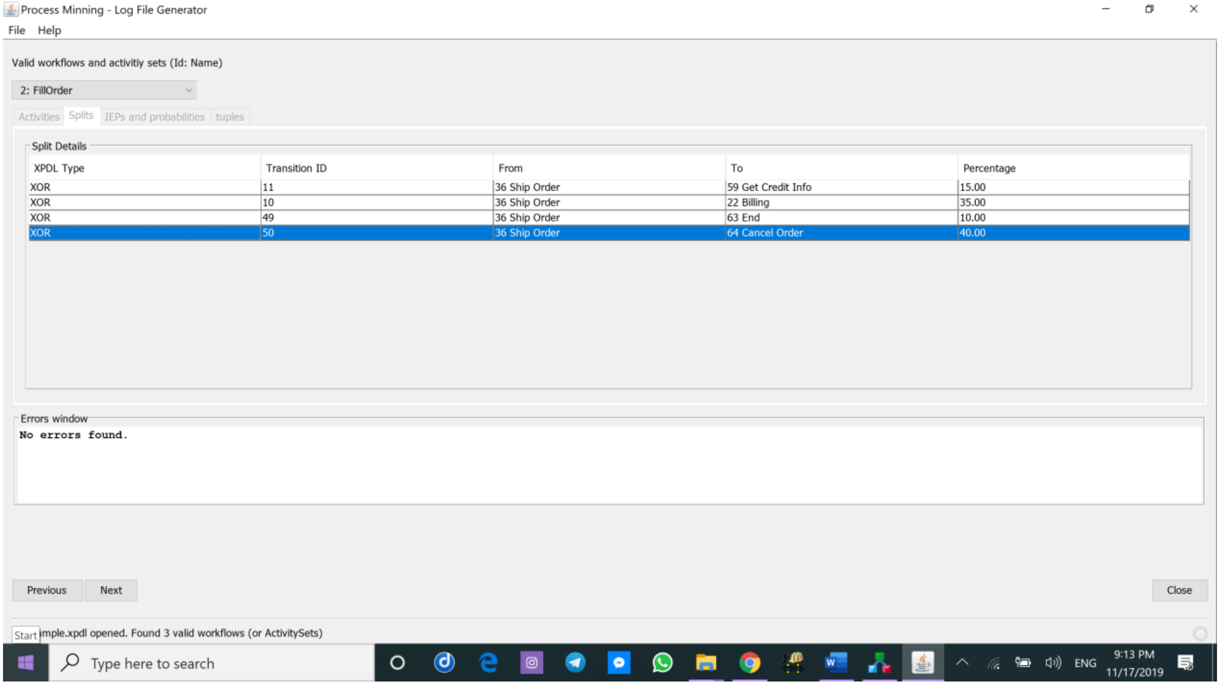
Errors window

```
No errors found.
```

File sample.xpdl opened. Found 3 valid workflows (or Activi

Figure 24 Log File Generator Application – Activities – Entering Data

Then user should input all the required fields and save them as CSV file in preferred path on the system and Click on Next to step forward.
In the next tab user sees every existed Splits in the PM and should give the percentage for each (consider the summation of that must be equal to 100 otherwise user will face to an Error and cannot go to next step)



Valid workflows and activitiy sets (Id: Name)

2: FillOrder

Split Details

XPDL Type	Transition ID	From	To	Percentage
XOR	11	36 Ship Order	59 Get Credit Info	15.00
XOR	10	36 Ship Order	22 Billing	35.00
XOR	49	36 Ship Order	63 End	10.00
XOR	50	36 Ship Order	64 Cancel Order	40.00

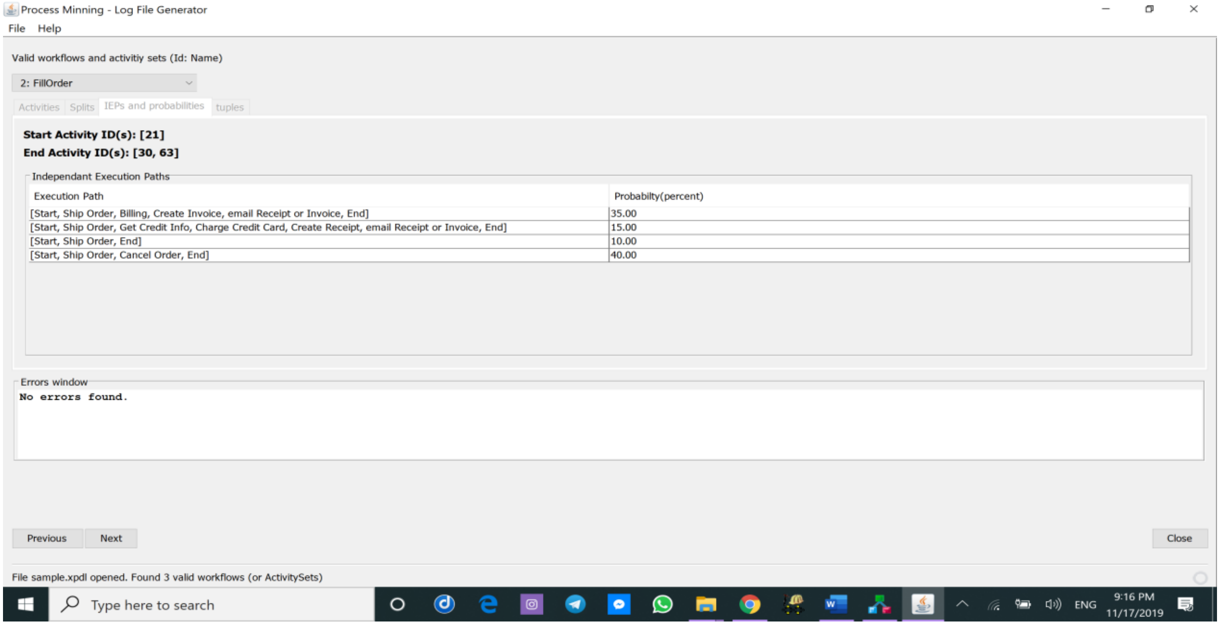
Errors window

```
No errors found.
```

mple.xpdl opened. Found 3 valid workflows (or ActivitySets)

Figure 25 Log File Generator Application – Splits – Entering Data

Here this is IEP within the PM, we have four IEP based on the info below. The probability of each is computed by App based on the percentage of each Splits that user is entered in pervious step and it means that the probability of running the first path is 35% and so on so forth. As mention before, last step is all about the number of tuples in Log File and saving it in user's preferred path.



Valid workflows and activitiy sets (Id: Name)

2: FillOrder

Start Activity ID(s): [21]
End Activity ID(s): [30, 63]

Independant Execution Paths

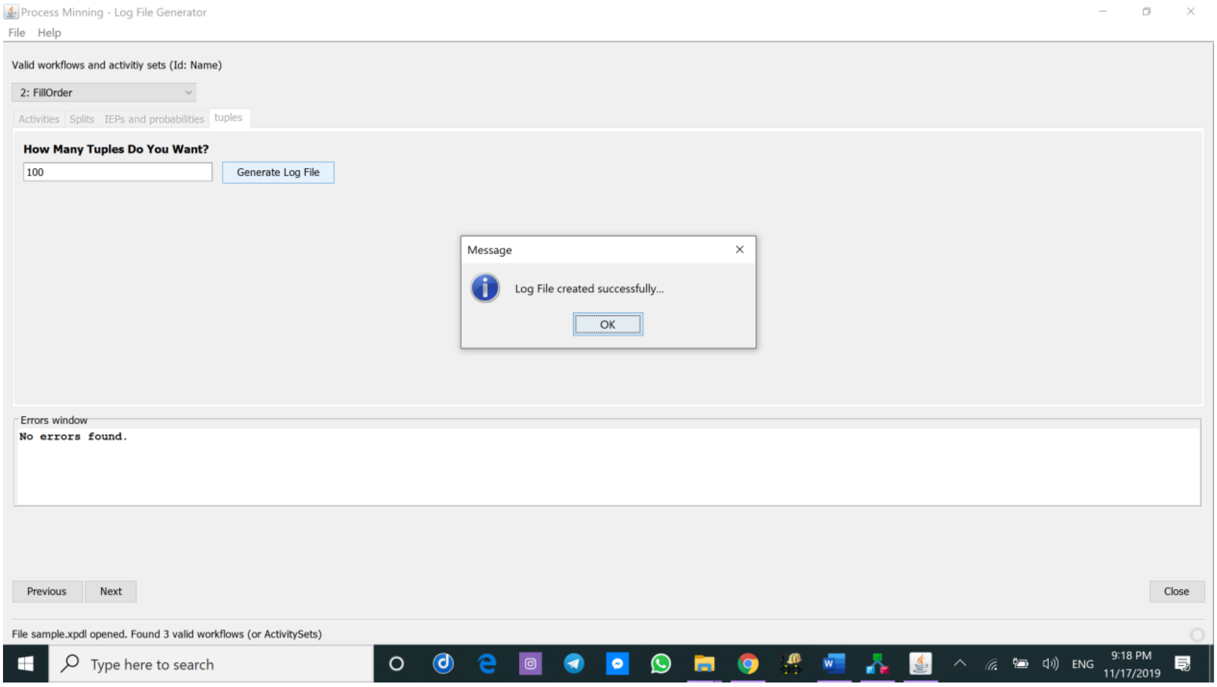
Execution Path	Probabilty(percent)
[Start, Ship Order, Billing, Create Invoice, email Receipt or Invoice, End]	35.00
[Start, Ship Order, Get Credit Info, Charge Credit Card, Create Receipt, email Receipt or Invoice, End]	15.00
[Start, Ship Order, End]	10.00
[Start, Ship Order, Cancel Order, End]	40.00

Errors window

```
No errors found.
```

File sample.xpdl opened. Found 3 valid workflows (or ActivitySets)

Figure 26 Log File Generator Application – Independent Execution Paths - Probabilities



Process Minning - Log File Generator

File Help

Valid workflows and activitiy sets (Id: Name)

2: FillOrder

How Many Tuples Do You Want?

100

Message

Log File created successfully...

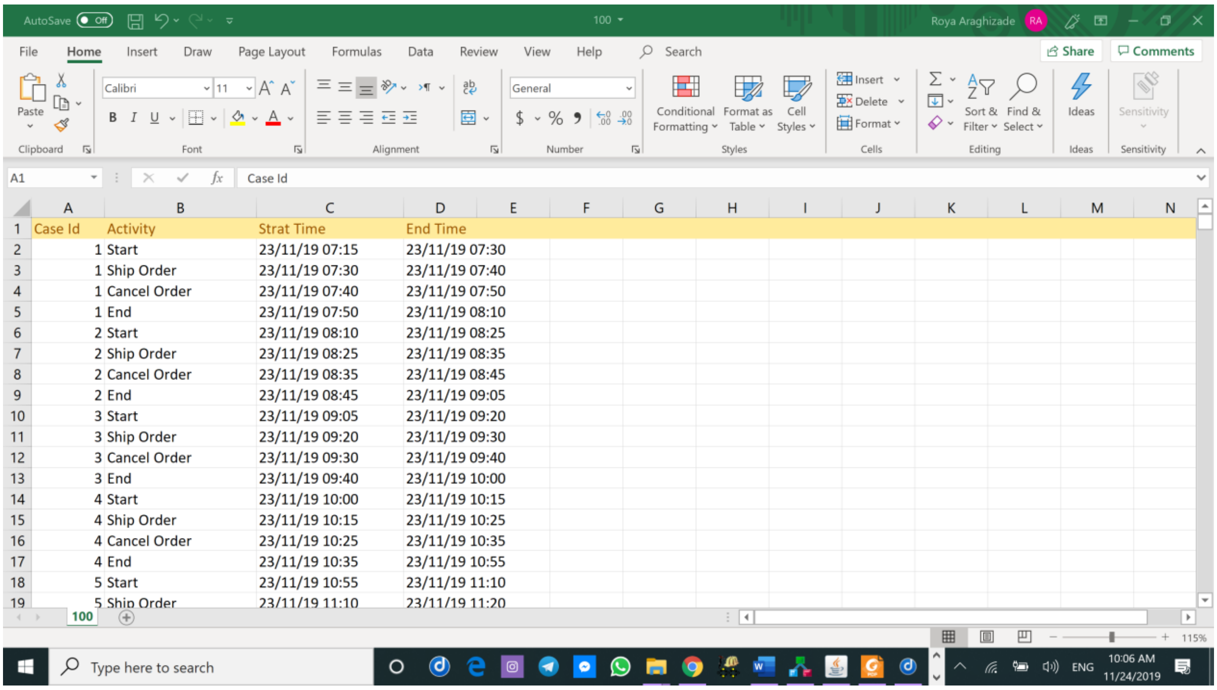
Errors window

```
No errors found.
```

File sample.xpdl opened. Found 3 valid workflows (or ActivitySets)

Figure 27 Log File Generator Application – Tuples – Generating Log File

The result of using LFG is in figure below. A Log File with all required data based on what user entered before. As we can see here, we have the basic data that a process mining tool needs to have for creating PM and analyzing it.



Case Id	Activity	Strat Time	End Time
1	Start	23/11/19 07:15	23/11/19 07:30
1	Ship Order	23/11/19 07:30	23/11/19 07:40
1	Cancel Order	23/11/19 07:40	23/11/19 07:50
1	End	23/11/19 07:50	23/11/19 08:10
2	Start	23/11/19 08:10	23/11/19 08:25
2	Ship Order	23/11/19 08:25	23/11/19 08:35
2	Cancel Order	23/11/19 08:35	23/11/19 08:45
2	End	23/11/19 08:45	23/11/19 09:05
3	Start	23/11/19 09:05	23/11/19 09:20
3	Ship Order	23/11/19 09:20	23/11/19 09:30
3	Cancel Order	23/11/19 09:30	23/11/19 09:40
3	End	23/11/19 09:40	23/11/19 10:00
4	Start	23/11/19 10:00	23/11/19 10:15
4	Ship Order	23/11/19 10:15	23/11/19 10:25
4	Cancel Order	23/11/19 10:25	23/11/19 10:35
4	End	23/11/19 10:35	23/11/19 10:55
5	Start	23/11/19 10:55	23/11/19 11:10
5	Ship Order	23/11/19 11:10	23/11/19 11:20

Figure 28 Log File in CSV Format

Every Activities are matched with all Activities we saw in the list of TWE and the first step of LFG App. Case ID determines where the process starts and where it ends. The case IDs start with the default value of one and are added to each path accordingly.

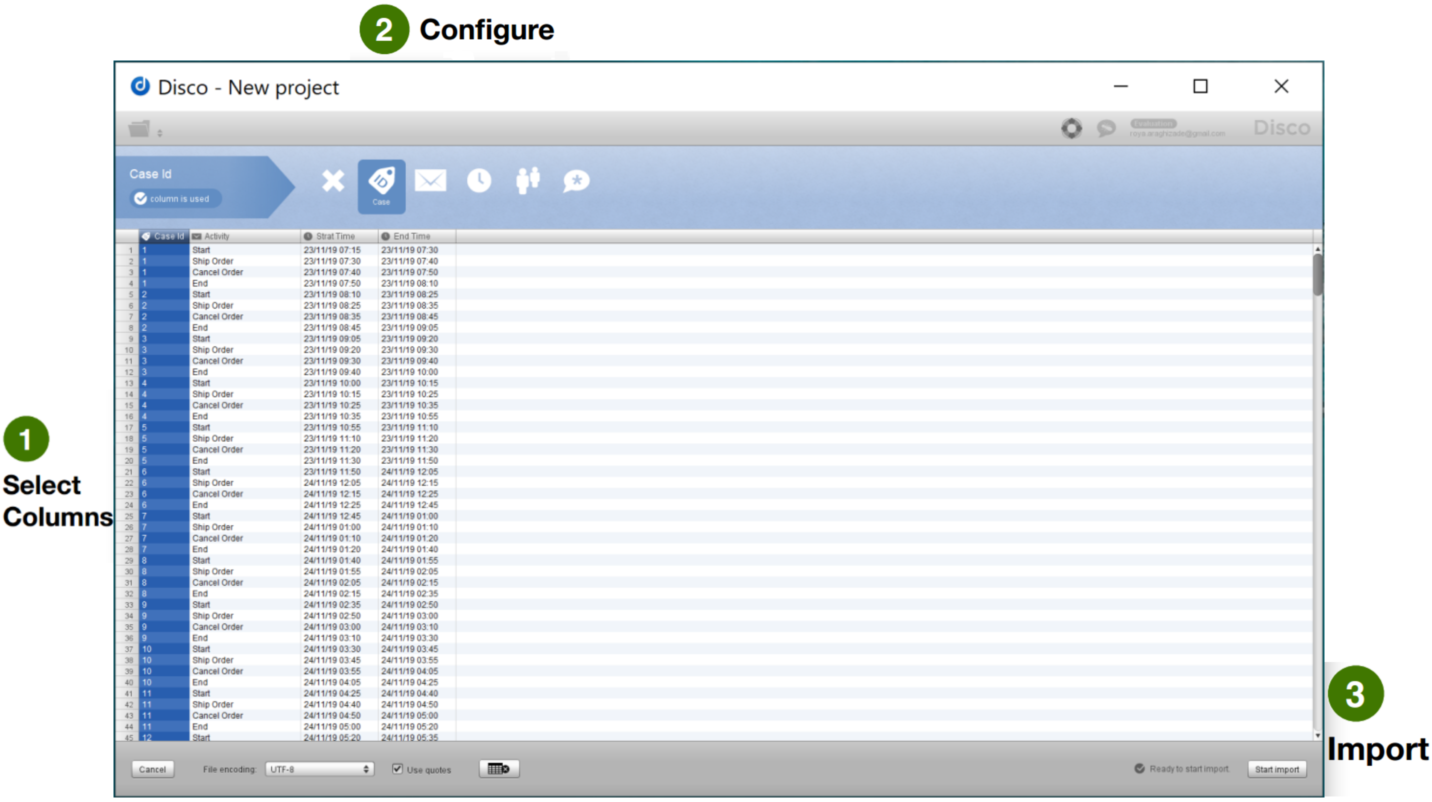
The highlight here is the probability of each path calculated by the software. For example, the first path has a 35% probability, which means that in the created log file, 35 tuples are assigned to that path exactly.

[Start, Ship Order, Billing, Create Invoice, email Receipt or Invoice, End]

And as explained before The Start and End time are calculated based on the Average Duration that is entered in the first stage. The Start Time of the First Activity (here is Case ID: 1 and Activity: Start) is set to the current system time.

5.2 Disco and the Generated Log File

Once we have extracted the right data, importing our event log in Disco is really easy. we can just open our file and simply select each column to configure it either as Case ID, Activity, Timestamp, then press Start import as shown in Figure below.



	Case Id	Activity	Strat Time	End Time
1	1	Start	23/11/19 07:15	23/11/19 07:30
2	1	Ship Order	23/11/19 07:30	23/11/19 07:40
3	1	Cancel Order	23/11/19 07:40	23/11/19 07:50
4	1	End	23/11/19 07:50	23/11/19 08:10
5	2	Start	23/11/19 08:10	23/11/19 08:25
6	2	Ship Order	23/11/19 08:25	23/11/19 08:35
7	2	Cancel Order	23/11/19 08:35	23/11/19 08:45
8	2	End	23/11/19 08:45	23/11/19 09:05
9	3	Start	23/11/19 09:05	23/11/19 09:20
10	3	Ship Order	23/11/19 09:20	23/11/19 09:30
11	3	Cancel Order	23/11/19 09:30	23/11/19 09:40
12	3	End	23/11/19 09:40	23/11/19 10:00
13	4	Start	23/11/19 10:00	23/11/19 10:15
14	4	Ship Order	23/11/19 10:15	23/11/19 10:25
15	4	Cancel Order	23/11/19 10:25	23/11/19 10:35
16	4	End	23/11/19 10:35	23/11/19 10:55
17	5	Start	23/11/19 10:55	23/11/19 11:10
18	5	Ship Order	23/11/19 11:10	23/11/19 11:20
19	5	Cancel Order	23/11/19 11:20	23/11/19 11:30
20	5	End	23/11/19 11:30	23/11/19 11:50
21	6	Start	23/11/19 11:50	24/11/19 12:05
22	6	Ship Order	24/11/19 12:05	24/11/19 12:15
23	6	Cancel Order	24/11/19 12:15	24/11/19 12:25
24	6	End	24/11/19 12:25	24/11/19 12:45
25	7	Start	24/11/19 12:45	24/11/19 01:00
26	7	Ship Order	24/11/19 01:00	24/11/19 01:10
27	7	Cancel Order	24/11/19 01:10	24/11/19 01:20
28	7	End	24/11/19 01:20	24/11/19 01:40
29	8	Start	24/11/19 01:40	24/11/19 01:55
30	8	Ship Order	24/11/19 01:55	24/11/19 02:05
31	8	Cancel Order	24/11/19 02:05	24/11/19 02:15
32	8	End	24/11/19 02:15	24/11/19 02:35
33	9	Start	24/11/19 02:35	24/11/19 02:50
34	9	Ship Order	24/11/19 02:50	24/11/19 03:00
35	9	Cancel Order	24/11/19 03:00	24/11/19 03:10
36	9	End	24/11/19 03:10	24/11/19 03:30
37	10	Start	24/11/19 03:30	24/11/19 03:45
38	10	Ship Order	24/11/19 03:45	24/11/19 03:55
39	10	Cancel Order	24/11/19 03:55	24/11/19 04:05
40	10	End	24/11/19 04:05	24/11/19 04:25
41	11	Start	24/11/19 04:25	24/11/19 04:40
42	11	Ship Order	24/11/19 04:40	24/11/19 04:50
43	11	Cancel Order	24/11/19 04:50	24/11/19 05:00
44	11	End	24/11/19 05:00	24/11/19 05:20
45	12	Start	24/11/19 05:20	24/11/19 05:35

Figure 29 Process mining Tool – Disco – Importing Log File

Timestamps can come in various formats. Different conventions regarding the order, separating characters, with or without spaces, etc. often make it a pain for data analysts to deal with timestamps.

Disco makes it as easy for us as possibly possible: When it parses the first rows of our data set to make suggestions for how we might want to configure data columns, different timestamp patterns are tested against our data to see which one gives the best match. This means that in more than 90% of the cases timestamp format is automatically detected and we do not need to manually configure anything about it at all. we can verify that everything is in order by selecting timestamp column in the configuration screen—like in Figure 3.5(a): If Disco says that the pattern matches all rows (see Figure 3.7 below) then everything is OK.

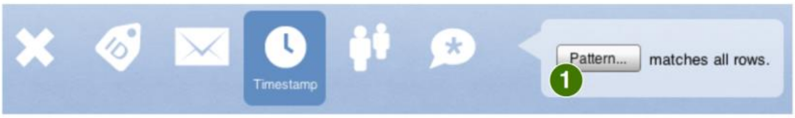
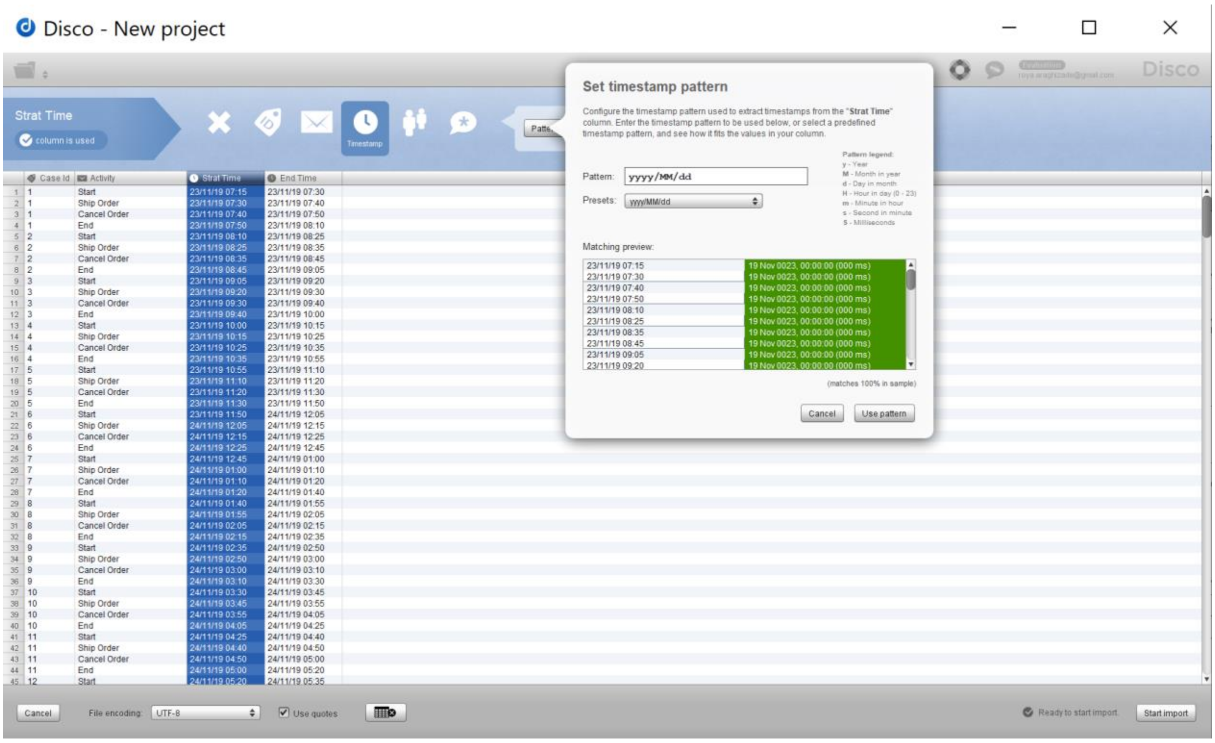


Figure 30 Process mining Tool – Disco – Selecting Time Stamp Pattern



	Case Id	Activity	Strat Time	End Time
1	1	Start	23/11/19 07:15	23/11/19 07:30
2	1	Ship Order	23/11/19 07:30	23/11/19 07:40
3	1	Cancel Order	23/11/19 07:40	23/11/19 07:50
4	1	End	23/11/19 07:50	23/11/19 08:10
5	2	Start	23/11/19 08:10	23/11/19 08:25
6	2	Ship Order	23/11/19 08:25	23/11/19 08:35
7	2	Cancel Order	23/11/19 08:35	23/11/19 08:45
8	2	End	23/11/19 08:45	23/11/19 09:05
9	3	Start	23/11/19 09:05	23/11/19 09:20
10	3	Ship Order	23/11/19 09:20	23/11/19 09:30
11	3	Cancel Order	23/11/19 09:30	23/11/19 09:40
12	3	End	23/11/19 09:40	23/11/19 10:00
13	4	Start	23/11/19 10:00	23/11/19 10:15
14	4	Ship Order	23/11/19 10:15	23/11/19 10:25
15	4	Cancel Order	23/11/19 10:25	23/11/19 10:35
16	4	End	23/11/19 10:35	23/11/19 10:55
17	5	Start	23/11/19 10:55	23/11/19 11:10
18	5	Ship Order	23/11/19 11:10	23/11/19 11:20
19	5	Cancel Order	23/11/19 11:20	23/11/19 11:30
20	5	End	23/11/19 11:30	23/11/19 11:50
21	6	Start	23/11/19 11:50	24/11/19 12:05
22	6	Ship Order	24/11/19 12:05	24/11/19 12:15
23	6	Cancel Order	24/11/19 12:15	24/11/19 12:25
24	6	End	24/11/19 12:25	24/11/19 12:45
25	7	Start	24/11/19 12:45	24/11/19 01:00
26	7	Ship Order	24/11/19 01:00	24/11/19 01:10
27	7	Cancel Order	24/11/19 01:10	24/11/19 01:20
28	7	End	24/11/19 01:20	24/11/19 01:40
29	8	Start	24/11/19 01:40	24/11/19 01:55
30	8	Ship Order	24/11/19 01:55	24/11/19 02:05
31	8	Cancel Order	24/11/19 02:05	24/11/19 02:15
32	8	End	24/11/19 02:15	24/11/19 02:35
33	9	Start	24/11/19 02:35	24/11/19 02:50
34	9	Ship Order	24/11/19 02:50	24/11/19 03:00
35	9	Cancel Order	24/11/19 03:00	24/11/19 03:10
36	9	End	24/11/19 03:10	24/11/19 03:30
37	10	Start	24/11/19 03:30	24/11/19 03:45
38	10	Ship Order	24/11/19 03:45	24/11/19 03:55
39	10	Cancel Order	24/11/19 03:55	24/11/19 04:05
40	10	End	24/11/19 04:05	24/11/19 04:25
41	11	Start	24/11/19 04:25	24/11/19 04:40
42	11	Ship Order	24/11/19 04:40	24/11/19 04:50
43	11	Cancel Order	24/11/19 04:50	24/11/19 05:00
44	11	End	24/11/19 05:00	24/11/19 05:20
45	12	Start	24/11/19 05:20	24/11/19 05:35

Set timestamp pattern

Configure the timestamp pattern used to extract timestamps from the "**Strat Time**" column. Enter the timestamp pattern to be used below, or select a predefined timestamp pattern, and see how it fits the values in your column.

Pattern: yyyy/MM/dd

Presets: yyyy/MM/dd

Pattern legend:
y - Year
M - Month in year
d - Day in month
H - Hour in day (0 - 23)
m - Minute in hour
s - Second in minute
S - Milliseconds

Matching preview:

23/11/19 07:15	19 Nov 0023, 00:00:00 (000 ms)
23/11/19 07:30	19 Nov 0023, 00:00:00 (000 ms)
23/11/19 07:40	19 Nov 0023, 00:00:00 (000 ms)
23/11/19 07:50	19 Nov 0023, 00:00:00 (000 ms)
23/11/19 08:10	19 Nov 0023, 00:00:00 (000 ms)
23/11/19 08:25	19 Nov 0023, 00:00:00 (000 ms)
23/11/19 08:35	19 Nov 0023, 00:00:00 (000 ms)
23/11/19 08:45	19 Nov 0023, 00:00:00 (000 ms)
23/11/19 09:05	19 Nov 0023, 00:00:00 (000 ms)
23/11/19 09:20	19 Nov 0023, 00:00:00 (000 ms)

(matches 100% in sample)

Cancel | Use pattern

Cancel | File encoding: UTF-8 | ☑ Use quotes | Ready to start import | Start import

Figure 31 Process mining Tool – Disco – Matching Time Stamp Pattern[9]

The timestamp pattern configuration screen allows us to inspect and modify the timestamp pattern to fit our data.

After we have configured our columns, the import can be started. As soon as the import is finished, we are directly taken to the Map view. In the Map view, we see a process map that visualizes the actual flow of our process based on the imported log data.

[9] *Feedback on how many rows of the selected timestamp column match the current timestamp pattern*

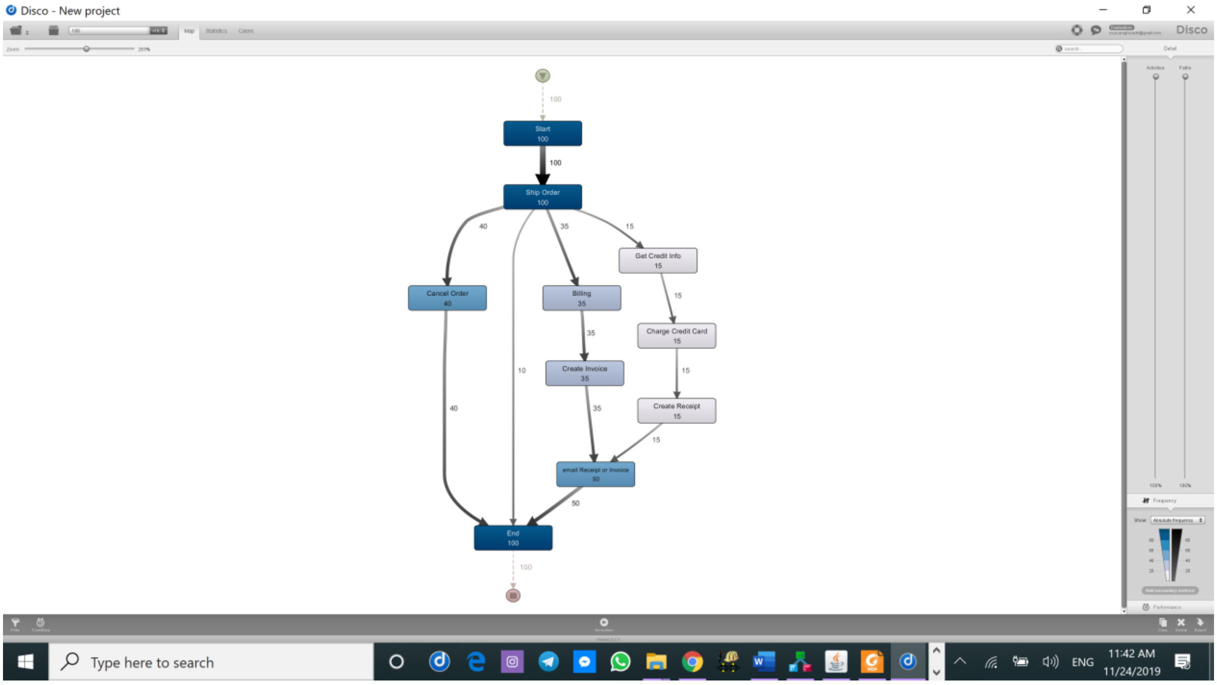


Figure 32 Process mining Tool – Disco – Process Map

The process map is the most important analysis result in Disco. It shows how our process has actually been executed. The process flows that we see in the Map view are automatically reconstructed ("discovered") based on the sequence and timing of the activities in your imported event log data. So, without further knowledge about the process, or any pre-existing process model, we obtain an objective picture of the real process.

The discovered process is visualized in a simple and intuitive way: The start of the process is illustrated by the triangle symbol at the top of the process map. Similarly, the end of the process is illustrated by the stop symbol. Activities are represented by boxes and the process flow between two activities is visualized by an arrow.

By default, the absolute frequencies are displayed in the numbers at the arcs and in the activities. The thickness of the arrows and the coloring of the activities visually support these numbers.

For example: In Figure above we can see that there are 100 cases (different instances of the Fill Order process) in the data set that all start with the activity Ship Order.

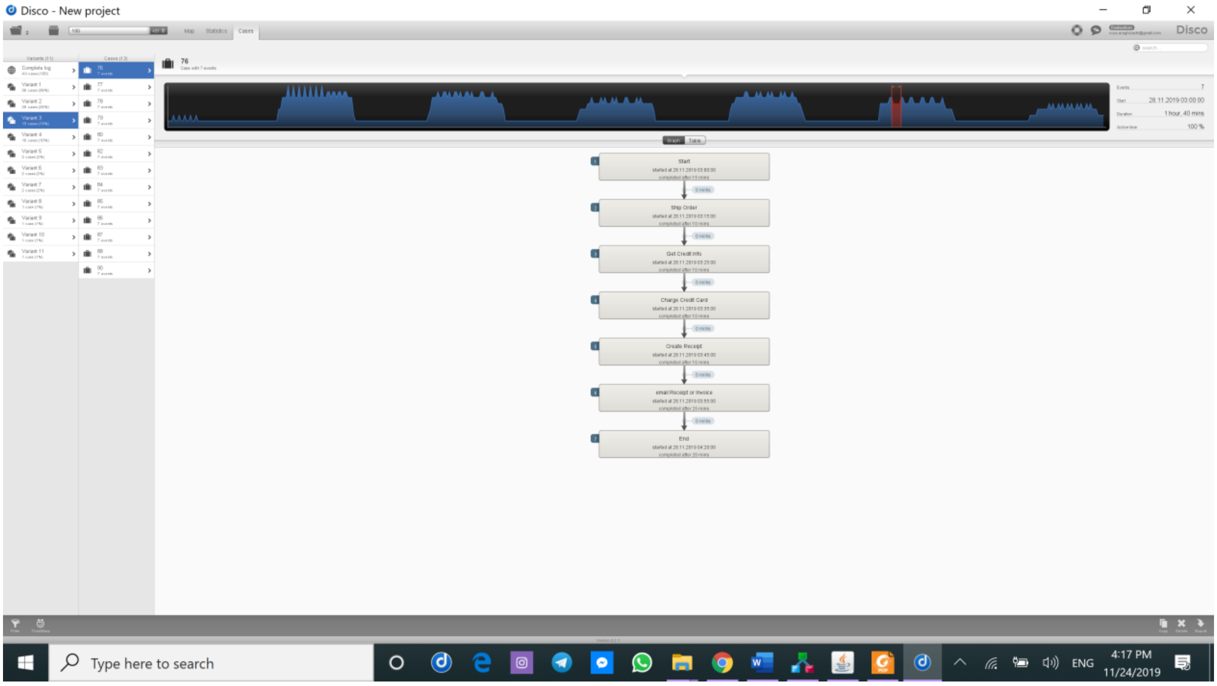
Afterwards, the process splits into four alternative paths: In 40 cases the activity Cancel Order was performed after Ship Order instead. In 15 cases the activity Get Credit Info and in 10 cases the activity End was performed directly after Ship Order. The other 35 cases perform activity Billing instead.

Totally, activity Ship Order is the one that is executed most often (in total 100 times) and then activity email Receipt or Invoice (in total 50 times) and activity Cancel Order (in total 40 times). It is visualized by thicker arrow and darker color.

While the Map view gives us an understanding about the process flows, the Cases view actually goes down to the individual case level and shows the raw data.

To be able to inspect individual cases is important, because we will need to verify our findings and see concrete examples particularly for "strange" behavior that will most likely discover in the process analysis. Furthermore, looking at individual cases with their history and all their attributes can give additional context (like a comment field) that sometimes explains why something happened. Finally, the ability to drill down to individual cases is important to be able to act on analysis.

Variants are an integral part of the process analysis. In Disco, a variant is a specific sequence of activities. we can see it as one path from the beginning to the very end of the process. A variant is then one "run" through this process from the start to the stop symbol, where also loops are unfolded. It is useful to look at the distribution of cases over variants in your data set to know what the most common activity sequences are, see how much variation there is and simplify our process map.



1 Start
started at 28.11.2019 03:00:00
completed after 15 mins

0 millis

2 Ship Order
started at 28.11.2019 03:15:00
completed after 10 mins

0 millis

3 Get Credit Info
started at 28.11.2019 03:25:00
completed after 10 mins

0 millis

4 Charge Credit Card
started at 28.11.2019 03:35:00
completed after 10 mins

0 millis

5 Create Receipt
started at 28.11.2019 03:45:00
completed after 10 mins

0 millis

6 email Receipt or Invoice
started at 28.11.2019 03:55:00
completed after 25 mins

0 millis

7 End
started at 28.11.2019 04:20:00
completed after 20 mins

Figure 33 Process mining Tool – Disco – Cases Analysis

Note that there are certain types of processes, which do not exhibit many common activity patterns. For example, in a hospital the diagnosis and treatment process for almost each patient is unique. These kinds of processes are often called "unstructured" and the variants and the variation filter are not of much help in these situations. Instead, you can use the simplification controls of the Map view to understand and analyze the process flows

5.3 Validation

By comparing the initial PM in TWE with the process model created at the Disco, after analyzing and executing process mining tool, can easily understand the strengths of Process Mining. Obviously in the process of the final model, analysis and follow-up of the process of performing the activities, it is very clear. It will also reveal the amazing potential power of the Process Mining by showing the probability of each activity and in general each path in Disco.
Just go back a bit and compare the Process Models. Apart from the analytics map, Disco enables the user to obtain statistical analysis as well as a case analysis, which examines both the probability of each route, the time and date of performing each activity.
In the example above, we saw a small and simple model. Now consider how difficult it can be to process very complex models. Process Mining helps us make the complexity of analyzing and analyzing the process of large models easy and understandable.
All the activities in the initial process model are visible in the created process model. It is also easy to understand the relationship between the data received from the user (probability of any IEPs) and the data displayed on the disco. Ten Activities, four Splits which shows four independent Execution Paths, a hundred Cases which clarifies the number of Tuples and also every execution frequency of each path. With all these matches we can figure it out the generated Process Model is the complete version of initial PM.

6. Result

As it has been mentioned earlier, the objective of this Thesina is become familiar with the concept of Business Process Modeling, Data exploitation ways, identifying existed independent execution paths, generation Event Logs that are randomly created according to the requirements and parameters chosen by user, utilizing it in suitable Process Mining Tool and in the end the final analysis using the available facilities of Process Mining.

Looking better at the initial phase of this work it is obvious the process model analysis could be pretty complicated, time consuming and in some particular cases could not even be trustworthy. Therefore, such analysis could cause serious damages in the business with irreversible consequences. Thus, the role of process mining is crucial.

What needs to be considered is that the initial process model was not understandable and we are not able to analyze it correctly. The investigated version of example cited in this Thesina has been over simplified compare to the real-world business processes. Having said that, it would be impossible or in other word even infeasible to identify execution paths, their sequences, probabilities, duration, start/end points of each activity and their execution variances, without process mining. In order to obtain trustworthy and useful results of analysis process model, process mining would be the best solution ever.

The power of process mining is to the point that only by taking advantage of all its facilities, it would be possible to achieve future substantial decisions of enormous businesses.

Finally, it is worth mentioning that the effort that has been made in this Thesina was to come up with practical solutions by gained acknowledgments and observations.

7. Conclusions

In this chapter an overall conclusion is drawn on the whole Business Process Modeling, creating event log, discussion of the experience during the development and process mining processes, reaching final result (a Business process) and finalizing with highlighting of possible points for future work.

We first started with a sample Business Process Model. After a basic introduction, and of course a little nonsense of the sample used with the help of Together Workflow Editor tool, one of the important purposes of this Thesina, which was developing log file generator software, was addressed. Along the way, we came across many challenges, including selecting proper log file format (CSV), the complexity of process model, the extraction of independent execution paths, and finally the selection of the suitable tools for the process mining (Disco).

The final result in this Thesina is a process model which can be visualized and analyzed carefully by using many factors that the process mining tool provides. By using this process model, user can easily be aware of the process of execution of each event, its activity and path and also the probability of each event.

Since one of the key aspects of process mining is to deliver an overview of processes to business users who are not familiar with raw data analytics, visualization should be applied with usability in mind. When choosing a process mining tool, it's important to ensure that visualizations provide precise, unbiased information in a comprehensive and structured way and Disco is fully capable in that.

Last but not the least is all data set generated by process mining tools can be exported in various formats. (Png, Jpg, Pdf, XML, CSV etc.)

7.1 Future research directions

As mentioned before, in this Thesina we consider that Ideally, our data is in perfect shape and we can immediately use it for process mining analysis without any changes (no loop). however, there are many situations, where this is not the case and we actually need to prepare data set a little bit to be able to answer analysis questions for example Unfold Loops for Cases.

In this case we are sure each independent execution path will run only once and computing the number of total execution paths can be easy. Future research directions can be related to this challenge which is how to remove loops or in another word how can find how many times an independent path will execute. The presence of loops increases the complexity of the process model and, as a result, process mining can also be complicated as well.

Process mining is an important tool for modern organizations that need to manage non-trivial operational processes. On one hand, there is an incredible growth of event data, on the other hand, processes and information need to be aligned perfectly in order to meet requirements related to compliance, efficiency, and customer service. Despite the applicability of process mining there are still important challenges that need to be addressed; such as Finding, Merging, and Cleaning Event Data, Dealing with Complex Event Logs, Cross-Organizational Mining, Combining Process Mining with Other Types of Analysis and Improving Usability for Non experts. These illustrate that process mining is an emerging discipline.

8. References

[1] G. Pozzi, "Workflow Management Systems," Como, 2016

[2] A. Bolt, M. de Leoni and Wil M. P. van der Aalst, Scientific workflows for process mining: building blocks, scenarios, and implementation, Journal-International Journal on Software Tools for Technology Transfer, Volume 18, Number 6, Pages 607-628, Year 2016, Publisher-Springer

[3] Julia Rudnitckaia, Process Mining: Data Science in Action, Journal-University of Technology, Faculty of Information Technology, Pages 1-11, Year 2015

[4] N. Dey, H. Das, B. Naik and H.S. Behera, Big Data Analytics for Intelligent Healthcare Management, Year 2019, Publisher-Academic Press

[5] J. Sun, Chandan K. Reddy, Big data analytics for healthcare, Book title-Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, Pages 1525-1525,Year 2013,Organization ACM

[6] Sabyasachi Dash, Sushil Kumar Shakyawar, Mohit Sharma and Sandeep Kaushik, Big data in healthcare: management, analysis and future prospects, Journal-Journal of Big Data, Volume 6, Number 1, Pages 54, Year 2019, Publisher-Springer

[7] George M. Giaglis, A Taxonomy of Business Process Modeling and Information Systems Modeling Techniques, Journal-International Journal of Flexible Manufacturing Systems, Volume 13, Number 2, Pages 209-228, Year 2001, Publisher-Springer

[8] O. Altuhhov, R. Matulevičius and N. Ahmed, International Journal of Information System Modeling and Design (IJISMD)

[9] Remco Dijkman, Jörg Hofstetter and Jana Koehler, Business Process Model and Notation Third International Workshop, BPMN 2011, Lucerne, Switzerland, Journal-Lecture Notes in Business Information Processing, Volume 95, Year 2011

[10] Stephen A. White, Process Modeling Notations and Workflow Patterns, Journal-Workflow handbook, Volume 2004, Pages 265-294, Year 2004

[11] Álvaro Rebugea and Diogo R.Ferreira, Business process analysis in healthcare environments: A methodology based on process mining, Journal-Information systems, Volume 37, Number 2, Pages 99-116, Year 2012, Publisher-Elsevier

[12] Fluxicon, Disco. Retrieved October 1, 2017, from fluxicon

[13] Wil M. P. van der Aalst, M. de Leoni and A. H. M. ter Hofstede, Process mining and visual analytics

[14] Workflow Handbook 2003 published by Future Strategies Inc., in collaboration with the WfMC

[15] Jorge Cardoso, Approaches to Compute Workflow Complexity, Dagstuhl Seminar, "The Role of Business Processes in Service Oriented Architectures, July 2006, Dagstuhl, Germany, Journal-International Journal of Business Process Integration and Management, Volume 2, Number 2, Pages 75, Year 2007

[16] Arthur, W. B., Complexity and the Economy, Journal-Science, Volume 284, Number 5411, Pages 107-109, Year 1999, Publisher-American Association for the Advancement of Science

[17] Wil M. P. van der Aalst, The Application of Petri Nets to Workflow Management, The Journal of Circuits, Systems and Computers, Journal-Journal of circuits, systems, and computers, Volume 8, Number 01, Pages 21-66, Year 1998, Publisher-World Scientific

[18] M. Castellanos, A.K. Alves de Medeiros, J. Mendling and A.J.M.M. Weijters, Handbook of Research on Business Process Modeling, Year 2009, Publisher-Information Science Reference Hershey

[19] R. A. Brown, J. C. Recker and S. West, using virtual worlds for collaborative business process modeling, Journal-Business Process Management Journal, Volume 17, Number 3, Pages 546-564, Year 2011, Publisher-Emerald Group Publishing Limited

[20] Cay S. Horstmann, Core Java Volume I – Fundamentals, Year 2002, Publisher-Pearson Education

[21] Joshua Bloch, Effective Java, Year 2017, Publisher-Addison-Wesley Professional

[22] Robert C. Martin, Clean Code, A Handbook of Agile Software Craftsmanship, Journal-Citado na, Pages 19, Year 2008

[23] Anne Rozinat, Disco User's Guide

[24] Christian W. Gunther and Anne Rozinat, Disco: Discover Your Processes, Journal-BPM (Demos), Volume 940, Pages 40-44, Year 2012, Publisher-Citeseer

[25] Maria Isabel Ribeiro, Gaussian Probability Density Functions, Year 2004

[26] William O. Baker, Phillip Y. Goldman, Computer Science: Programming with a Purpose, Online Course