Process mining application considering the organizational perspective using social network analysis

By

Ahmed Adel Fares Gadelrab Mohamed

Dissertation for the degree of Master in Modeling, Data Analytics and Decision Support Systems.

Supervised by

Professor Joao Gama
Professor Pedro Campos

Faculdade de Economia

Universidade do Porto

2016

Acknowledgments

To my family, especially my wife, who has always supported me during the two years of this master and without her efforts, This work couldn't be accomplished.

To Professors João Gama and Pedro Campos for their excellent assistance, support, guidance and understanding.

Special thanks to all coordinators of Erasmus Mundus Project "UNetBA", for giving me this great opportunity to achieve this point of submitting my Master's dissertation.

Last but not the least, to all my friends I made here in Portugal who became a real family for me.

Biographical Note

Ahmed Adel Fares was born on 12th of June 1988 in Alexandria, Egypt.

He received a bachelor degree in Management Information Systems at Faculty of Commerce, Alexandria University, Egypt, in 2010, and with a passion for Data Analysis, and an Erasmus scholarship, he joined the Master's Degree in Modeling, Data Analytics and Decision Support Systems in the School of Economics and Management of the University of Porto (FEP).

He works as Functional/Business consultant - Microsoft Dynamics ERP for Paradigm Solutions, Alexandria, Egypt, since 2012.

Abstract

Process Mining aims to extract useful information from event log by providing techniques and tools that are used for discovering process, control flow, organizational, and social structures from these logs. During this work, we will cover definitions of process mining, social network analysis and how we can mine a social network using process mining techniques. Event logs normally record information about the users executing the activities recorded in the log, so it is possible to extract social network from these logs for further analysis. To do so we combine concepts from process mining and social network analysis. Also a case study will be executed in order to proof the applicability of what has been theoretically mentioned.
Keywords: Process Mining, social network analysis, mining social networks, mining organizational perspective

Table of contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Nowadays almost all enterprise information systems store relevant events in more or less structured form using one or more software. For example, Enterprise Resource Planning (ERP) which record all transactions performed in any business process like sales, purchasing, manufacturing, etc. by recording all the steps from start to end. Another example could be Customer relationship management (CRM) like Microsoft Dynamics CRM (Microsoft, 2016), where all interactions with customers are being recorded including marketing, sales and customer services.

From the mentioned examples, there is a common definition of the structure of the saved event log, even if it is been referred by different names, it would be named transaction log, audit trial, history, etc., but it still has information about events which explained in a form of "Case" and "Activity". The case is the process instance which has begging and end steps, also between these steps, there could be some intermediate steps. The activity is the step in the case which sometimes named task. Moreover, as most of the steps are performed by people, event logs also would contain information on the people dealing with the steps like the person executing or initiating the event.

Process Mining is the link between classical process model analyses and data-oriented analysis like Data Mining and machine learning, it's focusing on processes (end to end processes) using the real data (Van der Aalst, 2011). Process Mining would be able to extract a useful knowledge from event logs by answering the following questions:

- What is the process that people are really following?
- Where are the bottlenecks in the process?
- Where do people (or machines) deviate from the expected or idealized process?

1.2 Dissertation overview

Dissertation consists of 4 chapters distributed over 2 parts as the following:

- Part one
  - **Introduction**: A brief summary of the dissertation, starting with the motivation mentioning a high-level definition for Process Mining and when we can use it. Then an overview of the structure of the dissertation mentioning the main idea behind each chapter. Finally, we will define the main problems and challenges, that lead us to do this work.
  - **State-of-the-Art**: Contains all theoretical definitions, algorithms and tools that will support this work. Also, the life cycle model for Process Mining, which will be mentioned during this work.
- Part Two
  - **Case study**: Our case study will be performed in this chapter, after introducing objectives and data source that will be used.
  - **Discussion of results**: The conclusions from methodology review in chapter 2 and results we have achieved during the case study execution in chapter 3 will be discussed in this chapter.

1.3 Problem definition

1.3.1 Objectives

There are three main perspectives in dealing with Process Mining:

- Process perspective (How?)
- Organizational perspective (Who?)
- Case perspective (What?)

During this work, the process perspective will be covered starting with process discovery until doing conformance checking. On the other hand, the organizational perspectives will be covered by extracting the social network from the process and analyzing it using social network analysis measures.

1.3.2 Challenges

Some challenges are not allowing objectives to be reached, as Process Mining is requiring date set in log format where events are referring to well-defined activities and each event is related to a particulate case. With additional information like date and time of executing the event, the resource who performed and type of the event would help in extracting useful information regarding the process.

Data itself is always the major concern. There are two types of procedures could help to avoid these challenges. Pre-logging procedures and post-logging procedures.

Firstly, pre-logging is to systematize logging events is the first step to getting meaningful analysis results in order to avoid GIGO (Garbage In Garbage Out (Van der Aalst, 2011)). There are 12 guidelines **(GL1-GL12)** for logging introduced by (Van der Aalst, 2015) that should be taken into consideration when building such systems.

GL1 Names of variables and its values should have the same meaning and could be interpreted in the same way between all people involved.

GL2 Names should be organized in a hierarchical form, so any update or new names introduced should satisfy this hierarchical form in order to make sure that it is not duplicated or misplaced.

GL3 Case identifier should not be reused or rely on the context. Should not create different logs depending on the language, time, or region settings.

GL4 Values should satisfy the desired precision as much as possible.

GL5 Uncertainty should be indicated.

GL6 Events should be ordered or at least partially ordered based on the timestamps or observed causalities in the case of timestamps are not precise enough.

GL7 Transactional information (start, complete, abort, schedule, assign, suspend, resume, withdraw,etc.) would help if it possible to be stored.

GL8 Regularly checks should be performed in order to ensure data quality over time.

GL9 Ensure comparability. The logging should not change over time unless it is reported, confirmed and ensure the past and future comparability.

GL10 While logging, events should not be aggregated, as it should be done during analysis.

GL11 Do not remove events just because they are no more active, mark them as irrelevant instead of deleting them.

GL12 Ensure privacy, but instead of removing Sensitive data, it could be coded or hashed.

Secondly, Post-logging procedures is how to get the right event data from existing database:

- Knowing exactly what process wanted to be mined and selecting relevant data distributed through data sources.
- Flatten event data by converting the nonstructured data from different sources and format into a single log format dataset.
- Mapping database to process instance and corresponding activities
- Data quality also is an issue which needs to consider: (later, more explanation for each problem will be provided)
  - Missing data
  - Incorrect data
  - Imprecise data
  - Irrelevant data

Chapter 2

State of the art

2.1 Introduction

This chapter contains all theoretical definitions, algorithms, and tools that will be used in this work. Starting with basic definitions of Process Mining and mining social networks, also some related works will be mentioned which motivate us to do this work.

Later, we will introduce some advanced definitions for Process Mining components, algorithms, modeling languages, tools, and perspectives.

Finally, The life cycle model for Process Mining will be mentioned. This life cycle contains the steps that should be followed in order to conduct a Process Mining project with a short definition for the main two processes types, that any process would be in one of these types.

2.2 Some basic definitions

2.2.1 Process Mining

Although Process Mining is a relatively young research field, but the interest in research in Process Mining is growing by the time because of the confrontation between the increasing number of events which being recorded in order to provide detailed information about the history of processes. (i.e., observed behavior) and the urgent need to improve and support business processes in rapidly changing and competitive environments.

Process Mining can be defined as "the missing link" between business process analysis and data analysis techniques. Very relevant for almost all organizations as it can be applied to analyze and improve any type of operational processes (organizations and systems) in a variety of domains. These event data could be organized and stored in a way that they will contain a history of what already happened during process execution, and this could also be analyzed using Process Mining techniques.

The term "Process Mining" could be referred to methods for defining a structured process description from real transactions. (W.M.P. van der Aalst and Others, 2003; R. Agrawal and Leymann, 1998; D. Grigori and Shan, 2001; M. Sayal and Dayal, 2002). The term "structured process description" could be defined in a several ways, starting from control-flow model explained in a graphical presentation like Petri net (Van der Aalst, 2011), till reaching modeling the organizational structure and social aspects.

There are three types of relationships between models and event data (Van der Aalst, 2011):

- Play-out: Start from a model we generate behavior based on what we have modeled before, this is typically simulation or workflow automation system configured on the basis of models.
- Play-in: Start from behavior we automatically generate models based on that. Here, we talk about Process Discovery that means learning process model from observed behavior. This learning shows what is really happening in organizations and in the system.
- Replay: This is the most important type in Process Mining because we confront model and reality. Considering a model behavior, we can find bottlenecks and other types of problems. This is useful for conformance checking, bottleneck analysis, predictions and many other purposes for improvements.

Examples of applications:

- Analyzing treatment processes in hospitals.
- Improving customer service processes.
- Understanding customers browsing behavior.
- Analyzing failures of a baggage handling system.

All of these applications have in common that dynamic behavior needs to be related to process models.

2.2.2 How Process Mining relates to data science?

- Process Mining versus Business Intelligence: Business Intelligence is a set of methodologies, processes, architectures, and technologies that transform raw data into meaningful and useful information used to enable more effective strategic, tactical, and operational insights and decision-making (Evelson B., 2010).

The main point where Process Mining differentiates from Business Intelligence (Rozinat, 2014a) is in the depth of the analysis. BI reporting tools focus on displaying the Key Performance Indicators (KPIs). However, they are not able to reach the root causes for the case. Process Mining can provide much deeper analysis into the actual processes by extracting the process flows and bottlenecks based on existing logs. Essentially, BI having the assumption that the processes are known. Process Mining assumes that even for well-defined processes it is not guaranteed that reality will follow exactly the planned, so activities and cases would require more advanced analysis.

- Process Mining versus Data Mining:

  Most of Data Mining tools are used to support business decisions in specific areas i.e., which products should be placed together, or: what are the most promising locations marketing flyers should be sent, but they do not focus on the process itself.

  At the same time, organizations are investing a lot on process modeling because it is done manually, with high potential to become outdated quickly and not following the reality.

  BY combining both Data Mining and process modeling strengths, Process Mining can automatically create process models based on existing log data and produces live models that are more connected to the current and most recent business that can be updated easily (Rozinat, 2014b).

2.2.3 Social network analysis

Overview

SNA aims to analyze social networks regarding the organizational perspective. It has a very important role as it evaluates the relations between people, teams, departments and/or the entire organizations (CROSS, 2001).

This kind of analysis is able to deliver important information which could able to improve the flow of communication inside the organizations and provide managers a clear view of how the work is being done as the main goal of SNA is to make the communication process as much transparent as it can and provide tools to enhance the process of communication.

All SNA techniques are based on a graphical representation, so the social network would be represented as a graph, where each node is a person (or group of people) and each link between two nodes is a relationship (HU, 2008; Moses and Boudourides, 2001).

SNA Measures

- Measures for an individual level
  - Degree: The Degree of a node (sometimes called Degree Centrality) is a number of nodes that are connected to it. This measure can be seen as the popularity of each actor (W.M.P. van der Aalst and Song, 2005).
  - Betweenness Centrality: Computes the influence that a node has over the spread of information through the network. For example, a node (i.e., person) with high betweenness centrality value means that it performs a critical role in the network because this person enables the connection between two different groups (W.M.P. van der Aalst and Song, 2005).
  - Closeness Centrality: Computes how close each node is to the other nodes in the network. For example, a node (i.e., person) with a higher closeness centrality value, will need to contact a lot of nodes in its ways to reach another node, and vice versa is happening for a node with a lower closeness centrality value as it has better chance to achieve the rest of nodes in a lower number of steps (W.M.P. van der Aalst and Song, 2005).
- Measures for the network level
  - Density: The value of this measure should range from 0 to 1 indicating how interconnectivity is in the network. In the social context, a high-density network means that almost everyone communicates with everyone. The density is defined as:

    $$Density = n/N2$$

    where n represents the links that there are in the network and N represents the maximum number of possible links (Hansen and Shneiderman, 2009).
  - Clustering coefficient: This metric determines by the probability of a network to be split into a finite number of sub-networks. In the social context, a new cluster is defined as a new team/group in the organization (Hansen and Shneiderman, 2009).
  - Centralization: This measure is connected to the individual notion of centrality. The lower the number of nodes with high centrality, the higher is the centrality of a network. i.e., the high centralized network is dominated by one or a few persons. If this person is removed, the network quickly split to unconnected sub-networks. A very high central network is not a good sign because it means that it has critical points of failure, putting too much trust and power in a single individual (Hansen and Shneiderman, 2009).

2.2.4 Mining social networks

All organizations establish a formal social structure where all the hierarchy relationships between employees are defined. However, in most cases, the relationships that really exist in the organizations have some gaps do with the predefined structure (CROSS, 2001).

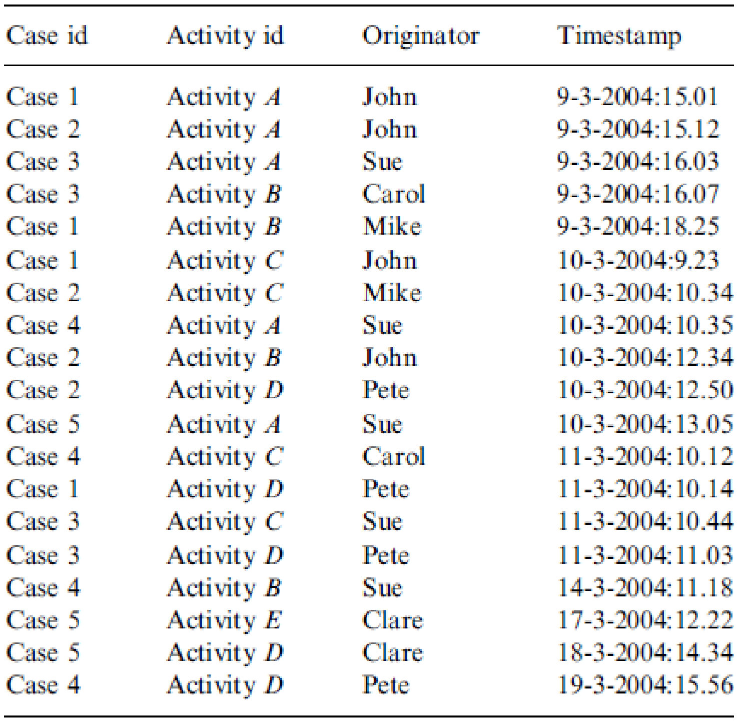
As it has mentioned in introduction, there are three main perspectives in dealing with Process Mining: (W.M.P. van der Aalst and Others, 2007)

- Process perspective: focuses on the flow of the information, i.e., the sequence of activities. The main goal here is to define a process which could cover all possible paths and expressed graphically i.e. Petri net.
- Organizational perspective focuses on the relations between the performers, i.e., which people are involved in the process model and how they are related and interacting. There are 2 main goals: extract the organizational structure by classifying people and show relationships among performers.
- Case perspective: focuses on the cases itself. There are several ways we can characterize case with, they can be characterized by paths they went through or by the values of some variables, e.g., a number of ordered products in a supply order would be a good variable to classify the case.

This part is related to the organizational perspective, more precisely in deriving social networks from event logs and makes it available for further SNA.

As events are being executed and/or initiating by people, so it is nature to collect information about these people while recording the activities details. For example we can define activity by (c, a, p) where c is the case, a is the activity, and p is the person, and if we have two followed activities regarding the same case could be defined as $(c, a1, p1)$ and $(c, a2, p2)$, in this case we can say that there is some work handover from $p1$ to $p2$ and also we can say that p1 has no relation with $p3$. Using this information, it is possible to derive a social network in terms of graph or matrix (van der Aalst and Song, 2004).

Figure 2.1 (W.M.P. van der Aalst and Others, 2007) shows an example of a log involving 19 events, five activities, and six originators. We can conclude that some activities are being executed by certain persons like activity A which is executed either by John or Sue. From this information we can derive a Petri net in figure 2.2 (a) (W.M.P. van der Aalst and Others, 2007). Also we can discover the organizational structure shown in figure 2.2(b)(W.M.P. van der Aalst and Others, 2007). Figure 2.2 (c)(W.M.P. van der Aalst and Others, 2007) shows another view on the organization based on the handover of work between individuals. Of course, for only five cases, this information are too clear do be caught but considering larger data sets it will not be that easy to capture the main and detailed roles in an organization.



Case id	Activity id	Originator	Timestamp
Case 1	Activity *A*	John	9-3-2004:15.01
Case 2	Activity *A*	John	9-3-2004:15.12
Case 3	Activity *A*	Sue	9-3-2004:16.03
Case 3	Activity *B*	Carol	9-3-2004:16.07
Case 1	Activity *B*	Mike	9-3-2004:18.25
Case 1	Activity *C*	John	10-3-2004:9.23
Case 2	Activity *C*	Mike	10-3-2004:10.34
Case 4	Activity *A*	Sue	10-3-2004:10.35
Case 2	Activity *B*	John	10-3-2004:12.34
Case 2	Activity *D*	Pete	10-3-2004:12.50
Case 5	Activity *A*	Sue	10-3-2004:13.05
Case 4	Activity *C*	Carol	11-3-2004:10.12
Case 1	Activity *D*	Pete	11-3-2004:10.14
Case 3	Activity *C*	Sue	11-3-2004:10.44
Case 3	Activity *D*	Pete	11-3-2004:11.03
Case 4	Activity *B*	Sue	14-3-2004:11.18
Case 5	Activity *E*	Clare	17-3-2004:12.22
Case 5	Activity *D*	Clare	18-3-2004:14.34
Case 4	Activity *D*	Pete	19-3-2004:15.56

Figure 2.1: An event log

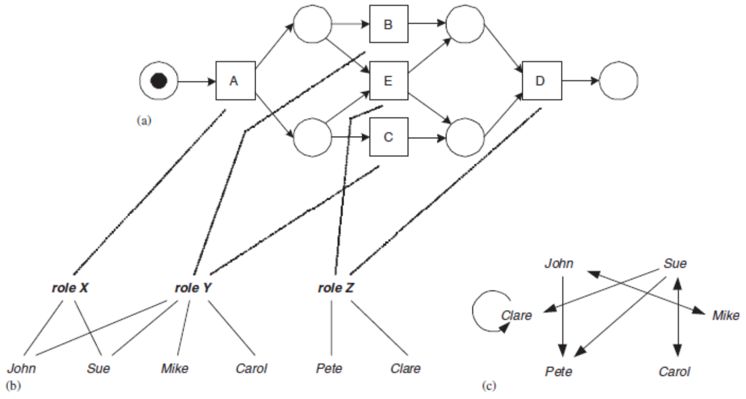


Figure 2.2: (a) the control-flow structure of a Petri net, (b) the organizational structure, and (c) a sociogram based on transfer of work

2.3 Related work

"Process Mining: Discovery, Conformance and Enhancement of Business Processes" (Van der Aalst, 2011) is the first book on Process Mining which could be named as the bible of Process Mining as it provides an inclusive overview of the state-of-the-art in Process Mining as a standalone branch of science. The book tends to be the main comprehensive reference covering the entire Process Mining concept.

"Mining Social Networks: Uncovering Interaction Patterns in Business Processes" (van der Aalst and Song, 2004) is the first paper proof the applicability to extract social structure from event logs, define the basic metrics and develop a tool to mine social networks from event logs (MiSoN).

"Business Process Mining: An industrial application" (W.M.P. van der Aalst and Others, 2007) is a case study describes the application of Process Mining in one of the provincial offices of the Dutch National Public Works Department, responsible for the construction and maintenance of the road and water infrastructure from three different perspectives: (1) the process perspective, (2) the organizational perspective, and (3) the case perspective. It has been built based on (Van der Aalst, 2011) in general and (van der Aalst and Song, 2004) in particular, especially when dealing with the organizational perspective.

"Social Network Analysis for Business Process Discovery" is a master thesis in Information Systems and Computer Engineering, Instituto Superior Tecnico, focuses on extracting and mining social networks using business process discovery with the respect of the organizational perspective as mentioned by (W.M.P. van der Aalst and Others, 2007; van der Aalst and Song, 2004) and also proposing approach that aims to overcome the difficulties of presenting the new information in a way that can be easily read by the ordinary users.

2.4 Advanced definitions

2.4.1 Process discovery

Process discovery is actually the main challenging task in Process Mining. Process Discovery means learning process model from observed behavior. This learning shows what is really happening in organizations and in the system. In order to do so, process discovery algorithms should consider the following characteristics according to (Van der Aalst, 2011):

- Representational Bias: Defines the search space by having the ability to represent concurrency, loops, silent actions, duplicate actions, OR-splits/joins, non-free-choice behavior and hierarchy.

- Ability to Deal with Noise: The discovered model should not include noisy (exceptional/infrequent) behavior as users are more interested in seeing the mainstream behavior. Also, rare patterns and/or activities can not produce such a meaningful information so it should be avoided while building the model. There are two ways to avoid noise, the first one is to be removed from the log before running the algorithm, the second one is letting the discovery algorithm to construct the model by abstracting from noise.

- Completeness Notion Assumed: Most process discovery algorithms make completeness assumption either implicit or explicit. Some algorithms assume that in order to say that one activity can be directly followed by another activity, this should be seen at least once in the log otherwise it will not be recognized as a relation and will tend to result in underfitting models. Other algorithms assume that the event log contains all possible traces, This is very unrealistic and results in overfitting models.

Noise and completeness used to determine the quality of the event log but did not mention the quality of the discovered model. Also, confusion matrix (Ting, 2010) can not be applied because

- No negative examples

- Log contains the only fraction of possible traces

- Almost versus poorly fitting traces

- In case of loops often infinitely many possible traces

- Murphy's Law of Process Mining (anything is possible, so probabilities matter)

There are so many dimensions being used in order to be able to validate how well a process model describes the observed data. In this work, we had used the four main quality dimensions: fitness, simplicity, precision, and generalization that has been introduced in (Van der Aalst, 2011). In this section, we briefly review these four dimensions and later they will be used in validating the discovered models in the case study.

These four quality forces are all pull in different directions at the same time. Whenever you optimize for one, you usually lose the quality in at least another quality dimension.

- Replay fitness: Is the model able to replay the observed behavior?

- Precision: Does the model not describe more behavior than we have actually seen? It's very easy to create a model that is capturing all the observed behavior but way too much. So we also want the model to be precise.

- Generalization: We need to be able to generalize the behavior from the limited observations. We've never seen all the behavior. But we want to describe it in a concise way.

- Simplicity: all this should be done in a simple process model. Any user should be able to read it. It shouldn't be a very large model with many crossing arcs.

Besides these four quality forces, there is an important factor in determining the quality of algorithm in order to produce reliable models which is soundness. Soundness or correctness is a set of properties that a process model should be able to reach the end state from the start state without bugs.

The first property is the *option to complete*, so once the process model is running for each state that can be reached in a process model, it should always be able to reach the end state, which is the target state.

The second property is *proper completion* so when the end state is reached, there should be no tokens or work left behind.

The third property is a property of *no transitions being dead.* So each transition in the process represents an activity. And each activity should be able to be executed at some point, through some path.

So soundness is captured by three properties. If all three properties are fulfilled, the process model is sound. If one or more properties are not fulfilled, the process model is not sound.

2.4.2 Modeling Languages

Petri Net

Petri net is the oldest process modeling language which is allowing modeling of concurrency, with simple and executable graphical notation. It has static network structure but, tokens can move in the network according to firing rules. The distribution of tokens over places is determining the state of Petri net and is referred to as its marking see Figure 2.3 (Van der Aalst, 2011).

- Places: Represented by the circles. Every Petri net has a start placing, usually with a token in it. In the case of parallelism, one transition will be connected with multiple output places. This is also a parallel split. Similarly, we have a parallel join,so multiple input places are connected to a single transition. We can also model choices, which is the opposite. We have one place, which is connected to multiple transitions. And each transition is competing for a token. Whoever fires first consumes a token, hence disabling the other transition. And therefore, we also have a choice join. So whenever you split a choice, you also have to join it to synchronizing. And finally, we have a final place that marks the end of the process. So as soon as a token is put in the end place, the process is done.

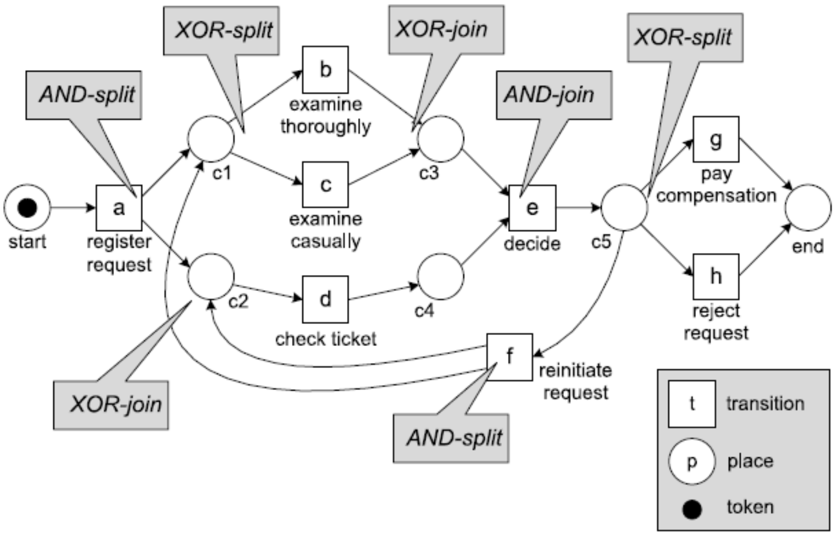


Figure 2.3: A marked Petri net (Van der Aalst, 2011)

- Transition: Represented by squares. The transitions can move tokens from all the input places that they are connected to all the output places that they are connected to. And they can only fire if all the input places have a token to consume. And it will produce tokens by the number of output places. Also, we may have silent transitions. They do not represent an observable activity, but they mainly distribute tokens in the Petri net.

Business Process Model and Notation (BPMN)

Similarly to Petri net, BPMN also has a start and an end place with boxes in between representing the activities see Figure 2.4, but the places in between the transitions or activities have been removed. And the parallel and choice constructs have been made explicit. So diamond operators with a plus sign in between, they represent parallelism. So activities can be executed in parallel, and a diamond operator with an X in it represents a choice, So either one of connected activities can be executed, but not all (Muehlen and Recker, 2008). So both the BPMN modeling notation and a Petri net describe exactly the same behavior. However, most process discovery algorithms will use Petri nets, because it's easier to model and as a more mathematical foundation.

Process Trees

Process trees guarantee to represent sound process models (van Eck et al., 2015).

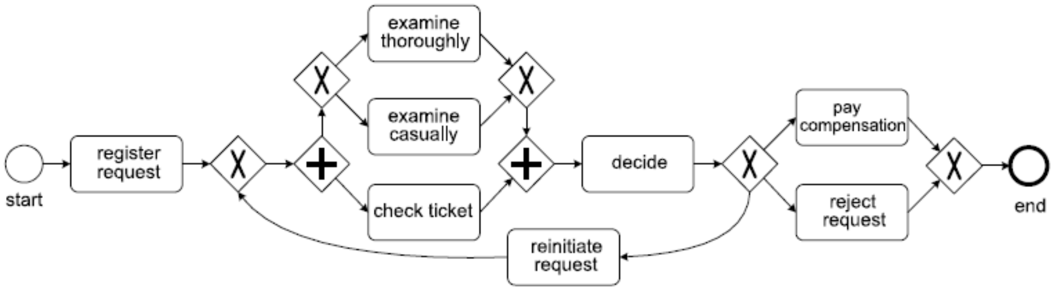


Figure 2.4: Process model using the BPMN notation (Van der Aalst, 2011)

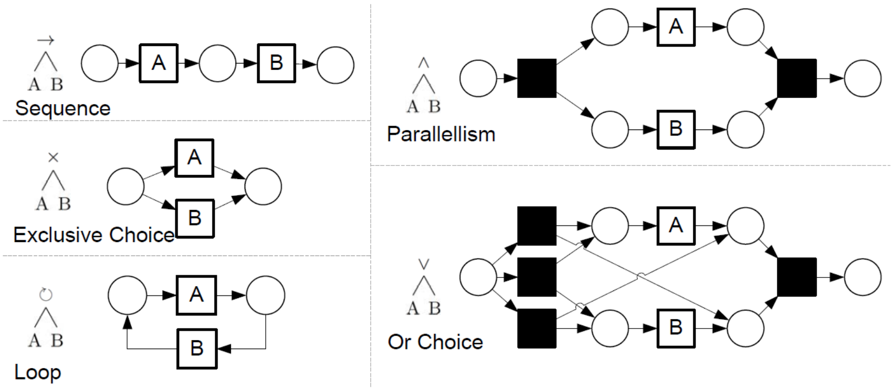


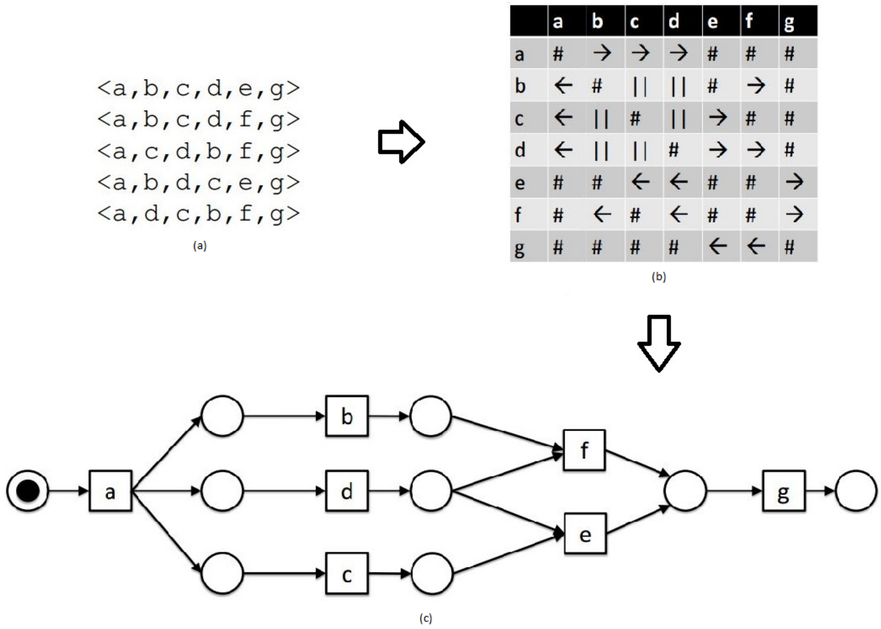
Figure 2.5: Relation between process trees and Petri nets (Buijs et al., 2012)

Figure 2.5 shows the Relation between process trees and Petri nets. The five available operator types are: sequence, exclusive choice, loop execution, parallel execution and non-exclusive choice. The order of the children matters as it specifies the order in which they will be executed (from left to right).

2.4.3 Process Mining Algorithms

Alpha algorithm

Alpha miner is the first algorithm that fit the gap between event logs, and the process model. For sure it has its flaws as it was the very first algorithm to be created, but it was a good starting point for later algorithms. The alpha miner has a few main steps (Van der Aalst, 2011).The first step is scanning the traces for ordering relations between activities then building a footprint matrix using these relations. There are 3 ordering relations that the alpha miner can detect.



```
<a,b,c,d,e,g>
<a,b,c,d,f,g>
<a,c,d,b,f,g>
<a,b,d,c,e,g>
<a,d,c,b,f,g>
```

(a)

	a	b	c	d	e	f	g
a	#	→	→	→	#	#	#
b	←	#	||	||	#	→	#
c	←	||	#	||	→	#	#
d	←	||	||	#	→	→	#
e	#	#	←	←	#	#	→
f	#	←	#	←	#	#	→
g	#	#	#	#	←	←	#

(b)

Figure 2.6: Modeling Petri net from event log using Alpha miner (a) Event log, (b) Footprint matrix, (c) Petri net

- Sequence (→): If A is sometimes directly followed by B, but never that B is directly followed by A.

- Parallel (||): If both A being directly followed by B, as well as B being directly followed by A.

- No direct relation (#): If A never being directly followed by B and B never being directly followed by A.

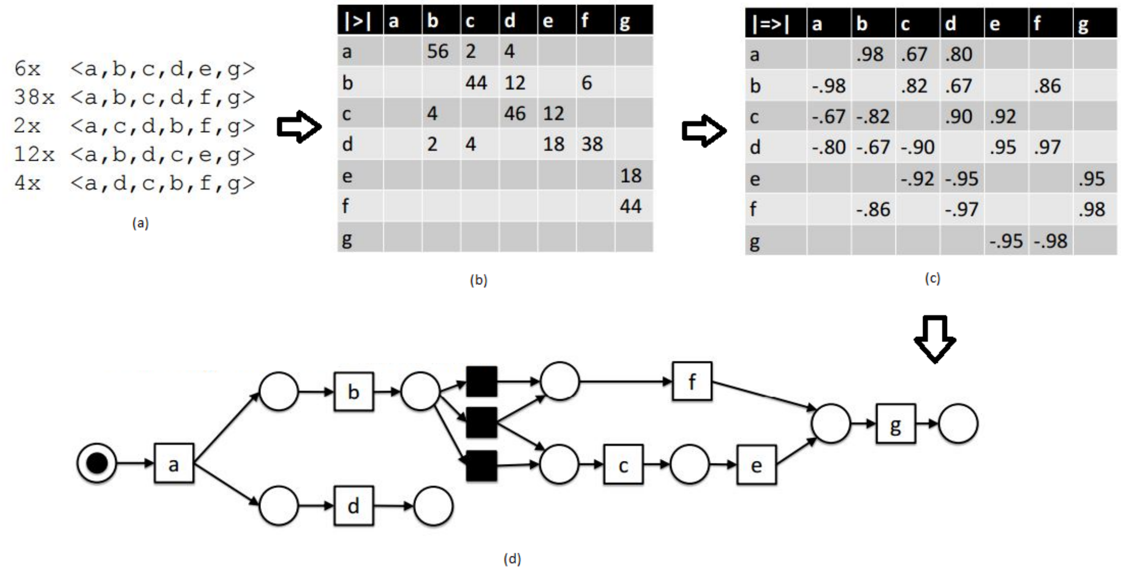
The next step, it takes this footprint matrix and converts it to a Petri net Figure2.6.

The α-algorithm nicely illustrates some of the main ideas behind process discovery. However, this simple algorithm is unable to manage the trade-offs involving the four quality dimensions (fitness, simplicity, precision, and generalization).

The main limitations in the alpha algorithm, it is ignoring frequencies which lead to completeness issues and also it does not guarantee soundness, so the produced model is not reliable.

Heuristics Mining algorithm

Heuristics miner is an improvement of the Alpha miner, especially on three issues. It takes frequencies into account, so it can filter out a noisy behavior or infrequent



```
6x  <a,b,c,d,e,g>
38x <a,b,c,d,f,g>
2x  <a,c,d,b,f,g>
12x <a,b,d,c,e,g>
4x  <a,d,c,b,f,g>
```

(a)

|>|	a	b	c	d	e	f	g
a		56	2	4			
b			44	12		6	
c		4		46	12		
d		2	4		18	38	
e							18
f							44
g							

(b)

|=>|	a	b	c	d	e	f	g
a		.98	.67	.80			
b	-.98		.82	.67		.86	
c	-.67	-.82		.90	.92		
d	-.80	-.67	-.90		.95	.97	
e			-.92	-.95			.95
f		-.86		-.97			.98
g					-.95	-.98	

(c)

(d)

Figure 2.7: Modeling Petri net from event log using Heuristics miner (a) Event log, (b) Directly-follows matrix, (c)Dependency matrix, (d) Petri net

behavior, it's able to detect short loops, and it allows skipping of single activities. However, it still does not guarantee sound process models.

Firstly, We can build the directly-follows matrix Figure 2.7(b) using frequencies in Figure 2.7(a).

Secondly, building dependency matrix Figure 2.7(c) using the following formula.

$$|a \Rightarrow_L b| = \begin{cases} \frac{|a >_L b| - |b >_L a|}{|a >_L b| + |b >_L a| + 1} & \text{if } a \neq b \\ \frac{|a >_L a|}{|a >_L a| + 1} & \text{if } a = b \end{cases}$$

Where $|a \Rightarrow_L b|$ is dependency between a and b (b is dependent on a), $|a >_L b|$ is direct relation from a to b.

Values between -1 and 1. A negative value means that there's a negative relation. A high value in positive means that there's a strong relation.

Finally, using these two matrices, we can filter certain relations, since we have both the frequency and the significance. And then using particular patterns, we can build a Petri net. And if we apply the Heuristics miner on these two matrices, we get the Petri net is shown in Figure 2.7(d).

Evolutionary Tree Miner (ETM) algorithm

ETM algorithm is a genetic algorithm enables to optimize the process discovery results based on user-defined quality dimensions: replay fitness, precision, generalization, and simplicity. Also, it is guarantee soundness as it uses process trees which

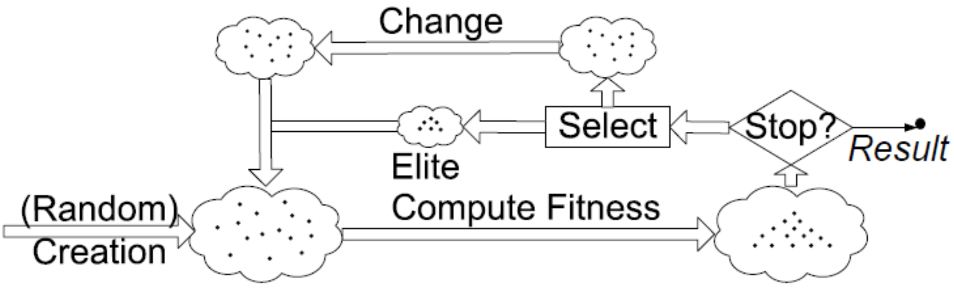


Figure 2.8: The different phases of the genetic algorithm (Buijs et al., 2012).

reduce the search space since unsound models will not be considered (Buijs et al., 2012).

By using a genetic algorithm for process discovery, ETM gains flexibility for changing the weights of different fitness factors, so the process discovery could be guided based on the weighted average of predefined quality factors based on the importance of each factor for the user.

Figure 2.8 shows the high-level steps most evolutionary algorithms are followed and so the ETM. From an input of event log, a population of random process trees is generated and quality dimensions are calculated for each candidate. The overall fitness of each process tree is calculated using the weight which the user originally provided to each dimension . In the case of one of stop criteria are satisfied, i.e., finding a candidate with the desired overall fitness, then return the fittest candidate otherwise, the population is changed and the fitness is again calculated. (Buijs et al., 2012).

2.4.4 Conformance checking

Causal Footprint approach

The first step is to create two footprint matrices, one based on the model and another one based on the original log. Then compare these two footprint matrices and count the differences. Conformance $= 1 - \frac{\text{Not matched cells}}{\text{All cells}}$ (Van der Aalst, 2011)

Limitations:

- Does not consider frequencies
- Just focus on directly follow relations and not ignore indirect relations.
- Capture fitness, precision, and generalization in a single matrix.

Token-based reply approach

The fitness of a case with trace σ on WF-net N is defined as follows: (Van der Aalst, 2011)

$$fitness(\sigma, N) = \frac{1}{2}\left(1 - \frac{m}{c}\right) + \frac{1}{2}\left(1 - \frac{r}{p}\right)$$

Where m represent missing tokens, c consumed tokens, r remained tokens and p produced tokens.

The fitness of an event log L on WF-net N:

$$fitness(L, N) = \frac{1}{2}\left(1 - \frac{\sum_{\sigma \in L} L(\sigma) \times m_{N,\sigma}}{\sum_{\sigma \in L} L(\sigma) \times c_{N,\sigma}}\right) + \frac{1}{2}\left(1 - \frac{\sum_{\sigma \in L} L(\sigma) \times r_{N,\sigma}}{\sum_{\sigma \in L} L(\sigma) \times p_{N,\sigma}}\right)$$

Note that $\sum_{\sigma \in L} L(\sigma) \times m_{N,\sigma}$ is total number of missing tokens when replaying the entire event log, because $L(\sigma)$ is the frequency of trace σ and $m_{N,\sigma}$ is the number of missing tokens for a single instance of σ. The value of fitness(L,N) is between 0 (very poor fitness; none of the produced tokens is consumed and all of the consumed tokens are missing) and 1 (perfect fitness; all cases can be replayed without any problems).

Limitations:

- Assumed visible and uniquely labeled transitions
- Too optimistic
- Local decision making may cause misleading results (Can not suggest better paths)

Alignment-based approach

Is the most advanced conformance checking approach? It starts with aligning observed behavior from the log to the modeled traces (by choosing the best-matched trace in the model to the behavior from the log). In the case of not having a perfect match (Synchronous move), the most similar trace would be chosen based on the cost function. (*Cost function*: giving a cost to each move happening in log only and does not have a similar move in the model and another cost for move in the model only which does not have a similar move in the log). The total cost defines which alignment should be considered (which trace should be considered for a particular behavior), of course, lower cost is always better. There could be more than one alignment solution (equals cost), so fitness and other quality measures will be used to decide the best alignment solution.

$$\text{Fitness} = 1 - \frac{\text{Number of moves in log only}}{\text{Number of actions in log only in worst case} + \text{Number of actions in shortest path}}$$

Advantages:

- Directly related to modeled behavior

- Very flexible as we can use any cost structure

- Detailed diagnostics

- The aligned model could be used for further analysis

2.5 Process Mining tools

There are several Process Mining tools. Some are free to use and open source (e.g. ProM). Many of the concepts of ProM have been embedded in commercial tools such as Fluxicon's Disco (www.fluxicon.com), Perceptive Process Mining (www.perceptivesoftware.com), Celonis (www.celonis.de), Aris, BPM One (Pallas Athena), Interstage (Fujitsu), Futura Reflect, Comprehend (OpenConnect), Process Discovery Focus (iontas), Enterprise Visualization Suite (Businesscape) and QPR ProcessAnalyzer (www.qpr.com).

2.5.1 Tools selection

Basically, ProM tool (Prom, 2010), which is an extensible framework that is completely pluggable environment, will be used in not just discovering, enhancing and doing a conformance checking for the process but also to extract the social network from the event log (W.M.P. van der Aalst and Others, 2007) as ProM has five different types of plug-ins:

- Mining plug-ins: Include Process Mining algorithms

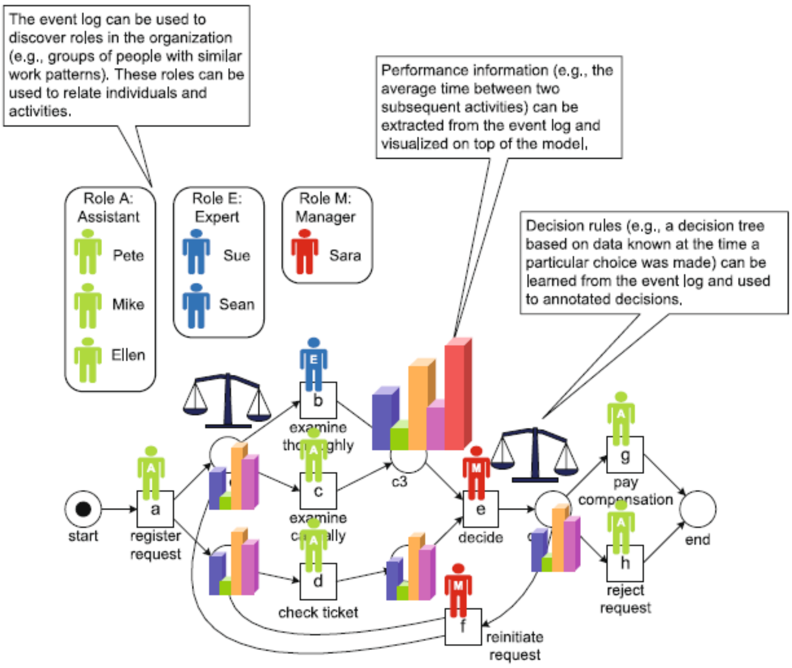
- Export plug-ins: Allow to save data, graphs or any kind of results as different types and extensions

- Import plug-ins: load different kind of data or objects

- Analysis plug-ins: Analyze the results of mining algorithms

- Conversion plug-ins: Convert data from different data formats

Earlier a tool named MiSoN (van der Aalst and Song, 2004) has been implements in order to extract social network from the event log and later this tool has been embedded in ProM as a plug-in and add some basic social network analysis tools.



The event log can be used to discover roles in the organization (e.g., groups of people with similar work patterns). These roles can be used to relate individuals and activities.

Performance information (e.g., the average time between two subsequent activities) can be extracted from the event log and visualized on top of the model.

Decision rules (e.g., a decision tree based on data known at the time a particular choice was made) can be learned from the event log and used to annotated decisions.

Figure 2.9: The process model extended with additional perspectives: the organizational perspective ("What are the organizational roles and which resources are performing particular activities?"), the case perspective ("Which characteristics of a case influence a particular decision?"), and the time perspective ("Where are the bottlenecks in my process?") (Van der Aalst, 2011).

2.6 Additional perspectives

The main perspective is processed perspective (control-flow) which focuses on the ordering of activities in order to find a good characterization of all possible paths, expressed in terms of a Petri net or some other notation (e.g., EPCs, BPMN, and UML ADs). Beside that perspective the model can be extended based on one or more additional perspectives see Figure 2.9.

2.6.1 Organizational perspective

Focuses on mine for hidden information about resources in the log, i.e., who are involved in the process and how are they related to each other. It has three main goals,(a) mine for the social network , (b) extract the organizational structure by classifying people in terms of roles and/or organizational units (c) analyze the relation between resources and activities (W.M.P. van der Aalst and Others, 2007).

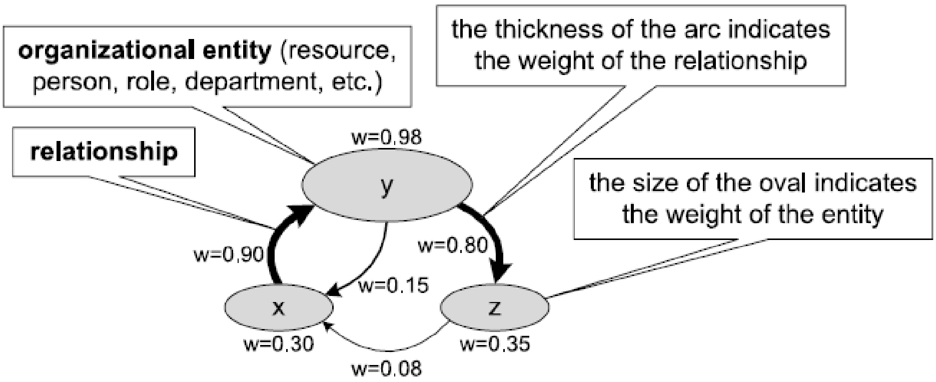

organizational entity (resource, person, role, department, etc.)
the thickness of the arc indicates the weight of the relationship
relationship
the size of the oval indicates the weight of the entity
w=0.98
y
w=0.90
w=0.80
w=0.15
x
z
w=0.30
w=0.35
w=0.08

Figure 2.10: A social network (Van der Aalst, 2011).

Social Network Analysis

In figure 2.10 The nodes correspond to the performer or actor of activities. It could be the users, computers, machines, etc who did actually perform the activity or it could correspond to organizational entities which the performer is a part of it like roles, groups, and departments. Choosing the actor is always based on the deepness of the analysis required and business requirements. Arcs represent relationships. Both nodes and arcs can have weights as a result of the frequent of appearing in the log. The thickness of the arc represent the Weight of the relationship between the two nodes in the edges of the arc and the size of the node represent the frequency where the actor perform an activity.

Sometimes the weight of an arc is represented by distance instead of thickness. The small distance represents high weight and vise verse.

Obviously, when event logs have a resource attribute, they could provide an excellent input for social network analysis. Based on this information, it is able to count the number of times when: (W.M.P. van der Aalst and Song, 2005)

- Work is handed over from one resource to another Within a case. For example, the handover of work from resource i to resource j within two subsequent activities where the first activity is completed by i and the second activity is completed by j.

- Reassigning an activity from one individual to another. For example, if i frequently delegates work to j but not vice versa, so we can assume that i is in a higher hierarchical relation with j.

- Tow resources doing Similar tasks. For example, if i and j are frequently doing similar tasks then, they have stronger relation than resources doing completely different tasks.

- Work is subcontracting between two resources. For example resource j is frequently execute a task in-between two tasks executed by individual i.
- Tow resources are Working together in the same case regardless causal dependencies. For example, resource i was working in three cases, where resource j was working in the same cases plus additional three cases where resource i was not working on, then the weight for i to j should be larger than the weight for j to i.

Discovering Organizational Structures

Each resource has a profile which indicates the frequent of activities done by that resource. Such profiles could be used by various clustering techniques like agglomerative hierarchical clustering and k-means clustering in order to discover similar resources. Also relevant information about resources and be added to the profile before clustering like salary, age, sex, etc.

After creating the resources clusters, these clusters can be matched to activities in the process. For example, if a resource executed a task, the task is assigned to the cluster (or we can call it organizational entities) which the resource belong to. (Song and van der Aalst, 2008).

Analyzing Resource Behavior

Performance results extracted from the event log could be applied on resources as long as to organizational entities as organizational entities are indirectly related to events in the log.

Some logs record the timestamp when a task is assigned to a resource and accomplish. In that case, it would be easy to extract information regarding utilization and response times of resources.Also, in most cases, organizations would like to do that analysis at a higher hierarchical level rather than at the level of resources. (Van der Aalst, 2011)

2.6.2 Time perspectives

Focuses on frequency and timing of events using timestamps which allow discovering bottlenecks, measure service levels and utilization of resources and also make a prediction of the remaining processing time for running cases.

2.6.3 Case perspective

Focuses on properties of single process instance (case). A path in the process or originators working on it could characterize a case. However, it is most interesting when different properties of individual cases are available. These properties are directly linked to a case, i.e., the amount of money of an invoice. cases can also be characterized by the values of the corresponding data elements. (W.M.P. van der Aalst and Others, 2007)

2.7 Conduct a Process Mining project

2.7.1 L* life cycle model for Process Mining

- Stage 0: Plan and Justify planned activities
  - A data-driven Process Mining project
  - A question-driven Process Mining project
  - A goal-driven Process Mining project
- Stage 1: Extract
  - Data
  - Objectives
  - Questions
  - Domain knowledge
- Stage 2:
  - Discover control flow model (Process model)
  - Connect events in the log to activities in the model (Alignment)
- Stage 3: Extend the model
  - Add perspectives
  - Return integrated model which is a starting point for further analysis
    - Diagnosis
    - Reengineering
    - Operational support
- Stage 4: operational support (Real time)

2.7.2 Mining Lasagna process

A process is a Lasagna process if with limited efforts it is possible to create as agreed-upon process model that has a fitness of at least 0.8.

It is structured, regular, controllable and repetitive. Also, Process Mining is usually been using for conformance checking rather than process discovery.

2.7.3 Mining Spaghetti process

Spaghetti process is unstructured, irregular, and flexible. It is very hard to discover such process except some simplifications are performed like

- Subset of activities
  - Only most frequent ones
  - Selected region of process
- Subset of cases
  - Homogeneous groups of cases (Clustering)
  - Natural sub-classes (Gold customers)
- Subset of paths
  - Only most frequent paths

Chapter 3

Case Study

3.1 Introduction

During this chapter, we will first define our case study objectives and goals, the business process and source of data and a detailed event log review with respect to variables and its value and the steps done to prepare the event log for the execution process.

Then in section 3.6, the case study will be performed starting from uploading describing the log file in ProM, going through process discovery and conformance checking techniques, and finally extending the model using organizational perspective.

3.2 Sources of Data

The data has been received from Statistics Portugal (INE) in order to perform this work. The data is related to Customer Relationship Management system, INE uses in recording all communication with clients requesting for data or information. In the following subsection, we will introduce briefly INE and their CRM system.

3.2.1 Statistics Portugal (INE) Process

INE overview

INE stands for *Instituto Nacional de estatística* in Portuguese and the literal translation is "National Institute of Statistics", but it will be mentioned in this work by abbreviation INE or the English name which the institute choose to be named with "Statistics Portugal".

The main responsibility for Statistics Portugal is to produce and publish official statistical information in the optimal possible accuracy and reliability, as this information recently became one of the main requirements for any economic or social development.It was created in 1935 and has its head office in Lisbon with delegations in the cities Porto, Coimbra, Évora, and Faro. (INE, 2015).

INE CRM

- Function: INE is using CRM to record and track all Emails and communications with the client requesting any kind of information such as surveys, data sets, census, etc. CRM is keeping track for all requests during the whole process and includes all resources working on that cases.
- Stakeholders: There are different stakeholders using the CRM system at INE like Customer service resources, Internal department consultants, Information technology department, higher management, etc.
- Process: The process of CRM starts with receiving an Email from a client requesting some kind of information. First, it is the responsibility of customer service resource to record and follow up the request that been assigned to him/her. later he/she is evaluating the request and give the client a feedback in case the request was within his/her knowledge and authority. Then the request is being forwarded to one of the internal department resources according to the type and specification of the request. After that internal department resource response to customer service employee who is forwarding this response to the client. In some cases requests need to be evaluated for the budget then this budget is being sent to the client for his confirmation, and in the case of approval, the data is sending to the client with and an invoice.

3.3 Questions

The following questions regarding the dataset would be answered during the work

- What is the form of the process?
- Which activities should be included in the model, and Which other activities will not have a big impact on the model in case we ignored them?
- Does the process comply with the guidelines?
- Does the process take too long time?
- Which performers are involved and how are they related?
- Are there employees not working with any other employee and only connected to specific employees?

3.4 Objectives

After analyzing the process and the social network we should be able to:

- Figure out how the real process should look like.
- Determine which activities should be included in the model and Which other activities will not have a big impact on the model in case we ignored them?
- Analyze whether there are deviations from the guidelines.
- Analyze and understand the social network by showing the relations between the performers.

3.5 Event log review

Originally the log was not in the desired format as rows represent cases and not activities with a total number of records 6812 related to all requests received during the year 2015. Most of the requests had been closed during 2015 but some of them did not close until March 10, 2016. This is why we can see some activities happening during the first quarter of 2016. Sometimes two or more records merged in the case of having more than one value for any variable, so the number of cases (Requests) is less than the number of records.

After translating variables and its values from Portuguese to English and unifying format of all date fields, we got the first log file version to start working with the following Variables list:

- Entry date: The date when the INE received and recorded the request.
- Final State: the last state achieved regarding a request.
- Request ID: Is the identifier of the request (Case)
- Name (Int.Dep.): Name of the responsible person for a particular request from the internal department.
- Last contact with the client: Latest reply from the client.
- Issue Classification: Requests are classified based on its nature (i.e.Population census, International trade, Healthcare, etc.))
- Closing date: The date when INE send the final answer to the client and close the case.
- Name (Cust.Serv.): Name of customer service agent who is responsible for a particular request.

- Response date from Int. dep: The date when internal department employee sends his response to customer service agent.
- Customer Response: The feedback regarding the budget.
- Requires budgeting?: classify request into payable and non-payable requests.
- Budget value: budget value in case of payable requests.

The first challenge was to find a way to clean and reformat the log, so it would be suitable for Process Mining in the following steps:

- Coding the values of the following variables for the data confidentiality:
  - Name (Cust.Serv.) –> Agentxx (i.e.Agent01, Agent02,etc.)
  - Name (Int.Dep.) –> Consxx (i.e. Cons01, Cons02, etc.)
  - Blank values for employees names had been filled by imaginary two employees Cons00 and Agent00.
  - Budget amounts using some kind of conversion formula.
- Merge values with similar functions in a more general name.
- Ignore nonclosed requests
- convert the data set into log format with variables (Request ID, Activity, Time, Resource, Role (Customer service, Internal department), Point of specialty, Requires budgeting?, Budget Value,)in the following steps:
  1. Convert 'Entry date' variable into 'Record the request' activity
  2. Convert 'Response date from Int. dep.' variable into 'Response from Int. dep.' Activity
  3. Covert 'Closing date' variable into 'Close the request' activity
  4. Convert 'Last contact with the client' variable based on its values.
  5. Convert 'Final State' variable based on its value and the same date of 'Closing date' as it is the final activity just before closing the request.
  6. Convert 'Customer Response' variable based on its values with the same date of ' the Last contact with the client date'

3.6 Execution guide

3.6.1 Prepare and describe event log in ProM

1. Upload the .csv file to ProM using "CSV File (XES Conversion with Log package)" plug-in.

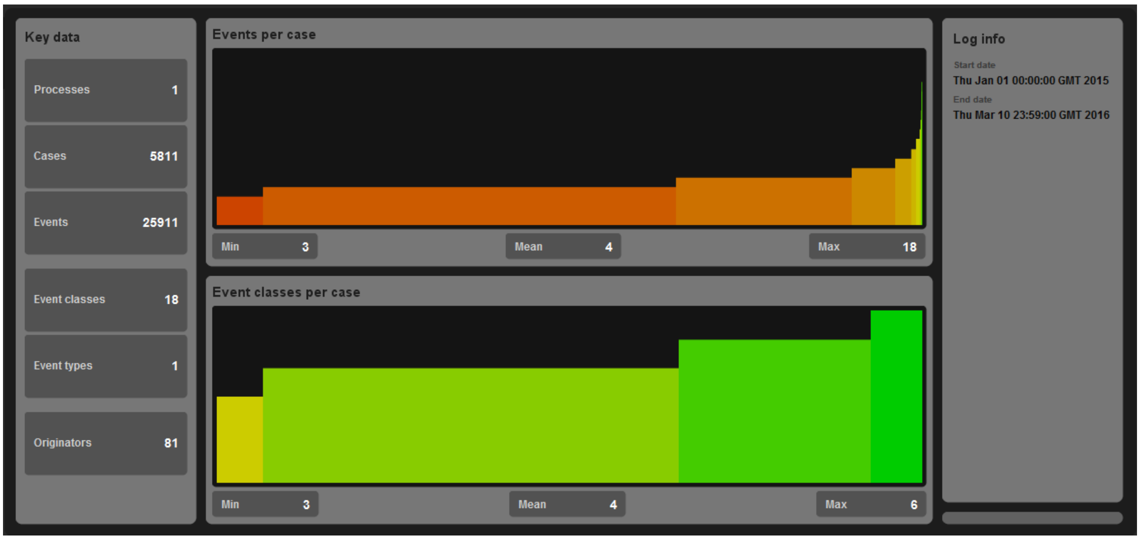
2. Convert the file into .XES format using "Convert CSV to XES" plug-in using the following mapping"

   - Request ID –> Concept:instance
   - Activity –> Concept:name
   - Date –> time:timestamp (dd-mm-yy h:mm)
   - Resource –> org:resource
   - Role –> org:role
   - Requires budgeting –> cost:type
   - Budget value –> cost:amount

3. Now we extract some statistical summary of the log (i.e. Number of cases, total,minimum ,maximum, average number of activities per case,Start date, End date, etc.) see Figure 3.1. Also, we can know occurrences of events in log (How many times an event has appeared). From Figure 3.2 we can see both activities 'recording the request' and 'Close the request' have been taken placed exactly the same number of times (5811 times) which are the same number of cases in the log. Also, 100% of cases were started with 'record the request' event, but only 3.7% of cases ended with 'Close the requests'. This happened because some case have the same timestamp, so we should reorder tasks with the same date in more logical order using "Enhance log: change the ordering of Event log with same timestamp (In place)" plug-in. From Figure 3.3 it makes more sense now by having more than 98% of cases ended with 'Close the request' event.The cases did not end with 'Close the request' (some activities followed the closing) will be studied in the following point.

4. By focusing more on traces using another view 'Explore Event Log (Trace variants, Searchable, Sortable)', we can see the most frequent cases in Figure 3.4, shortest traces in Figure 3.5, longest traces in Figure 3.6, The cases did not ended with 'Close the request' in Figure 3.7.

5. Dotted chart view had been used to view cases and activities behavior over time. For a clearer view, activities had been colored based cost type (Payable vs. non-payable). From Figure 3.8 we can see that cases are almost having the same behavior of using activities over time except for the last quarter of 2015



Key data

Processes	1
Cases	5811
Events	25911
Event classes	18
Event types	1
Originators	81

Events per case: Min 3, Mean 4, Max 18

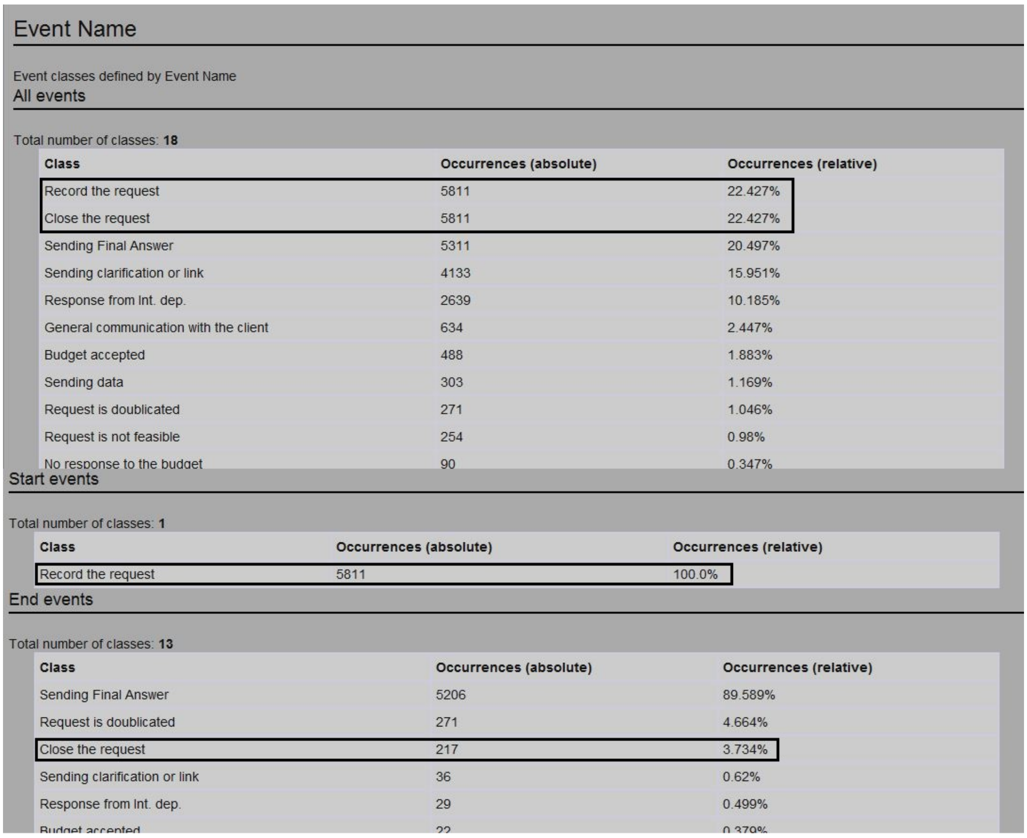
Event classes per case: Min 3, Mean 4, Max 6

Log info

Start date
Thu Jan 01 00:00:00 GMT 2015

End date
Thu Mar 10 23:59:00 GMT 2016

Figure 3.1: ProM log dashboard



Event Name

Event classes defined by Event Name

All events

Total number of classes: **18**

Class	Occurrences (absolute)	Occurrences (relative)
Record the request	5811	22.427%
Close the request	5811	22.427%
Sending Final Answer	5311	20.497%
Sending clarification or link	4133	15.951%
Response from Int. dep.	2639	10.185%
General communication with the client	634	2.447%
Budget accepted	488	1.883%
Sending data	303	1.169%
Request is doublicated	271	1.046%
Request is not feasible	254	0.98%
No response to the budget	90	0.347%

Start events

Total number of classes: **1**

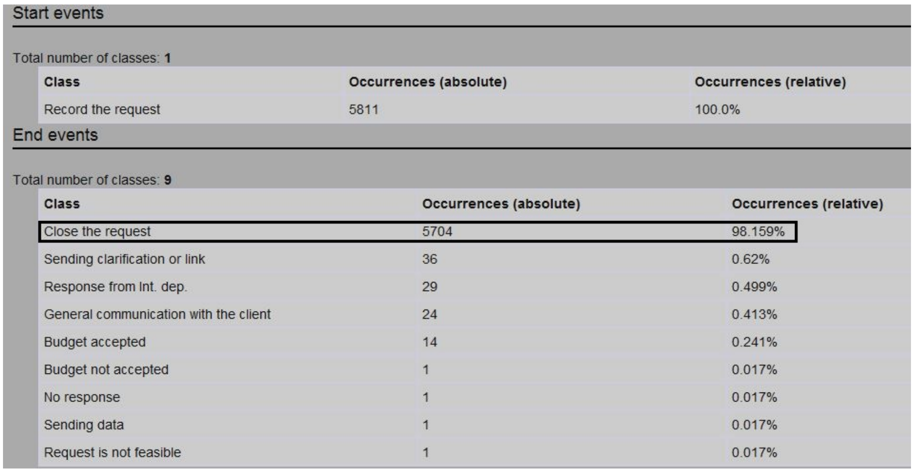
Class	Occurrences (absolute)	Occurrences (relative)
Record the request	5811	100.0%

End events

Total number of classes: **13**

Class	Occurrences (absolute)	Occurrences (relative)
Sending Final Answer	5206	89.589%
Request is doublicated	271	4.664%
Close the request	217	3.734%
Sending clarification or link	36	0.62%
Response from Int. dep.	29	0.499%
Budget accepted	22	0.379%

Figure 3.2: ProM Log Start and End Events



Start events

Total number of classes: **1**

Class	Occurrences (absolute)	Occurrences (relative)
Record the request	5811	100.0%

End events

Total number of classes: **9**

Class	Occurrences (absolute)	Occurrences (relative)
Close the request	5704	98.159%
Sending clarification or link	36	0.62%
Response from Int. dep.	29	0.499%
General communication with the client	24	0.413%
Budget accepted	14	0.241%
Budget not accepted	1	0.017%
No response	1	0.017%
Sending data	1	0.017%
Request is not feasible	1	0.017%

Figure 3.3: ProM Log Start and End Events (Reordered)

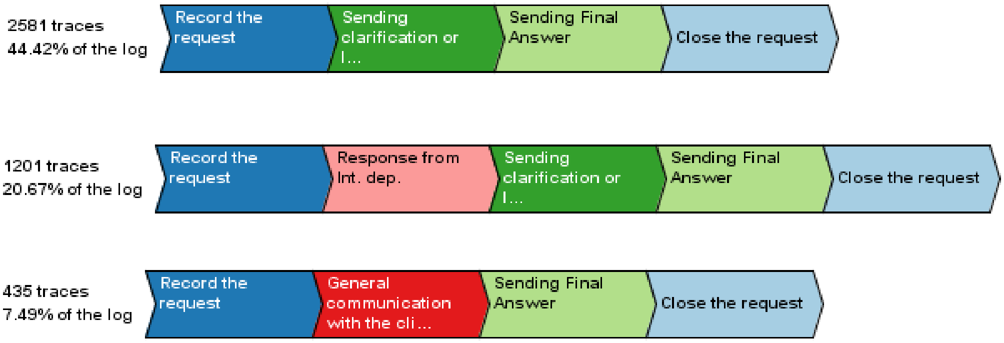


Figure 3.4: Most frequent traces

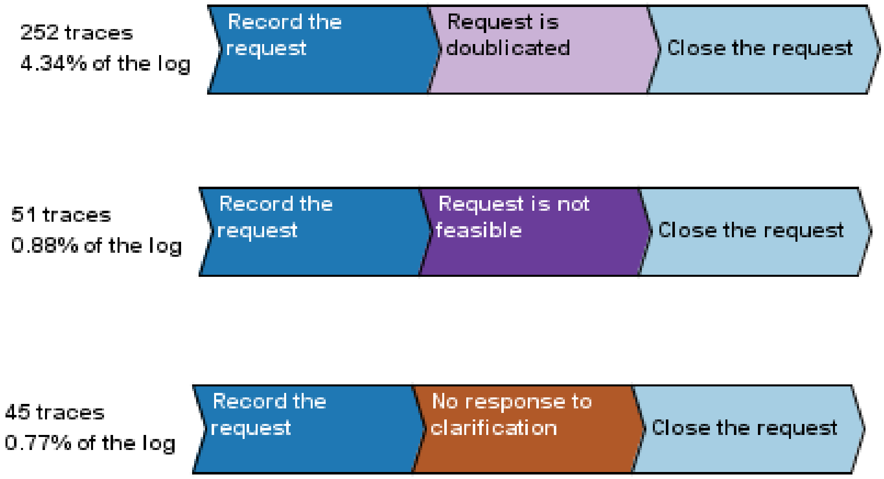


Figure 3.5: Shortest traces

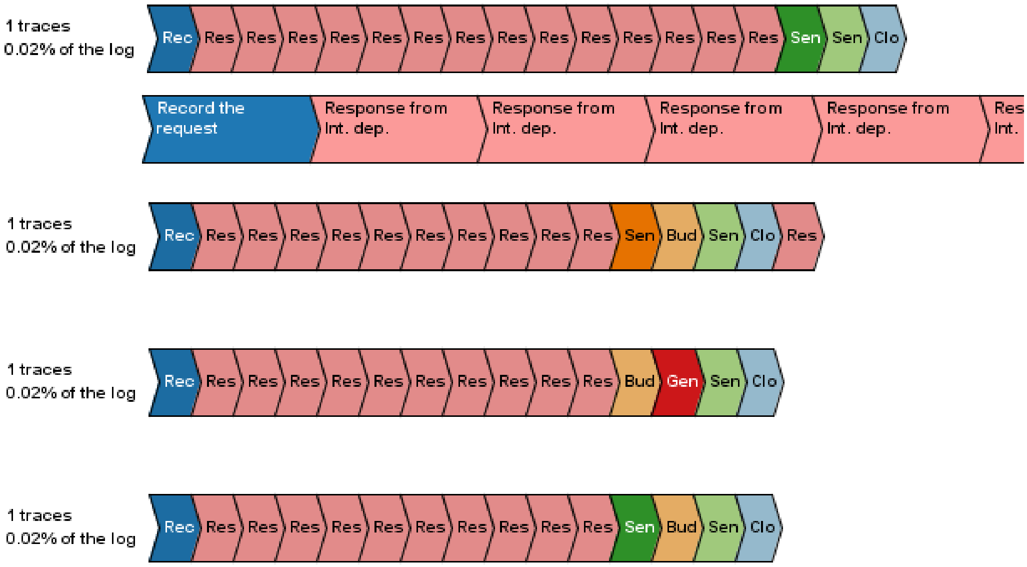


Figure 3.6: Longest traces

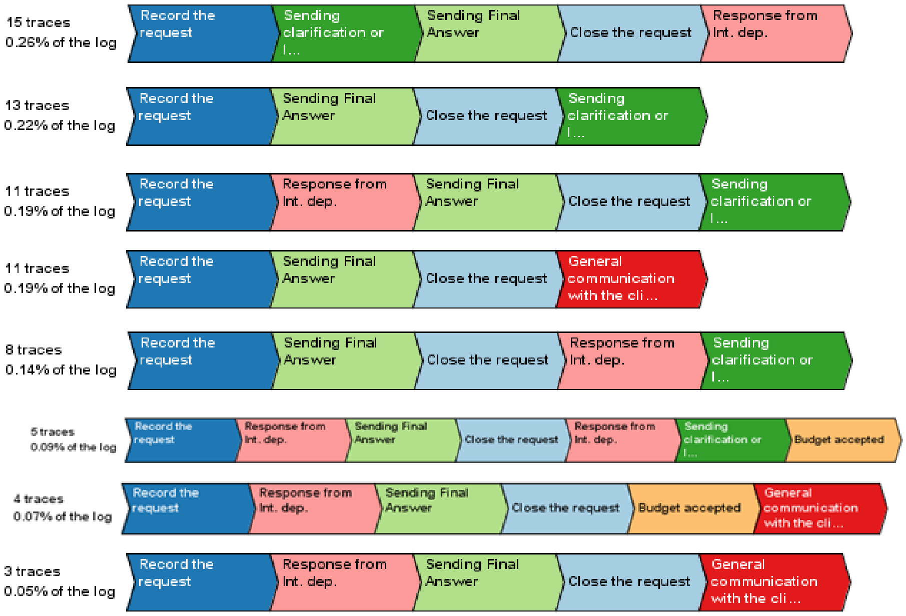
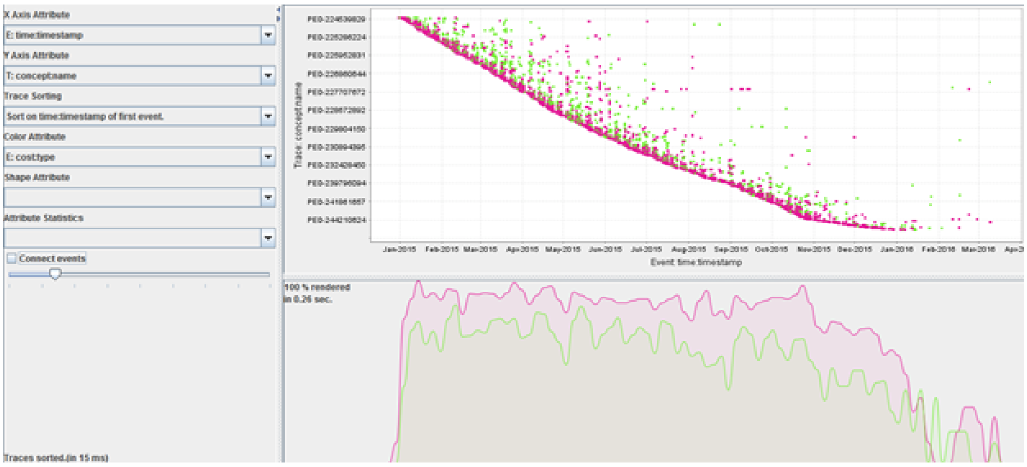


Figure 3.7: Traces ends after closing the requests

and the first quarter of 2016. Also after connecting events per case in Figure 3.9, it is obviously that cases are not having the same duration and some cases consumed very long duration (almost a year). So in order to have a clear vision for the behavior of newcomers cases, the activities had been filtered based on "Record the request" activity and sort the cases based on the cost type for clear distinguish between the two types as in Figure 3.10.Both types almost having the same behavior of constant frequency of new cases arrival, but the payable cases are always having a frequency lower than non-payable cases. Of course, there are no new cases recorded in 2016 because our log just has data regarding 2015 and the only reason of having some activity in 2016 is that some cases have not been closed during 2015 and some of the activities – rather than "Record the request" activity- are recorded during 2016.During the last quarter of 2015, the frequency of receiving new cases started to decrease gradually also with very similar behavior between the payable and non-payable cases except that payable cases started the decreasing phase earlier –after the first week of September- and non-payable cases started to decrees by the last week of October.



X Axis Attribute

E: time:timestamp

Y Axis Attribute

T: concept:name

Trace Sorting

Sort on time:timestamp of first event.

Color Attribute

E: cost:type

Shape Attribute

Attribute Statistics

□ Connect events

Traces sorted.(in 15 ms)

100 % rendered in 0.26 sec.

Figure 3.8: Requests over time colored based on the cost type (Payable in light green and non-payable in purple)

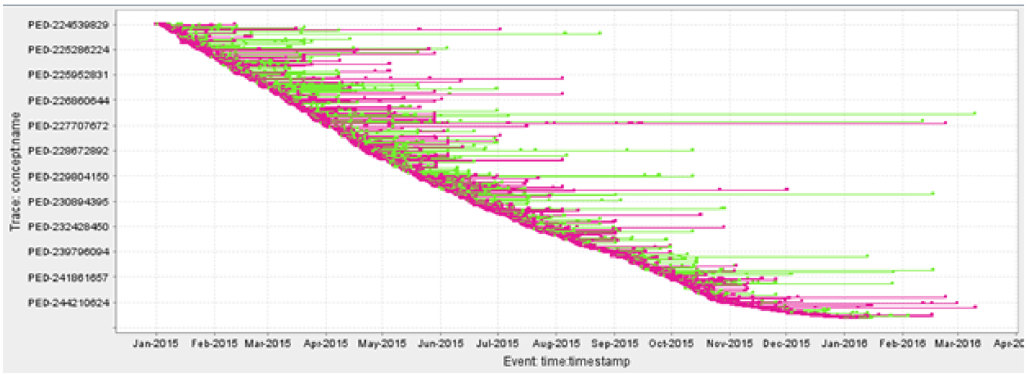


Figure 3.9: Requests over time (connecting events per case)

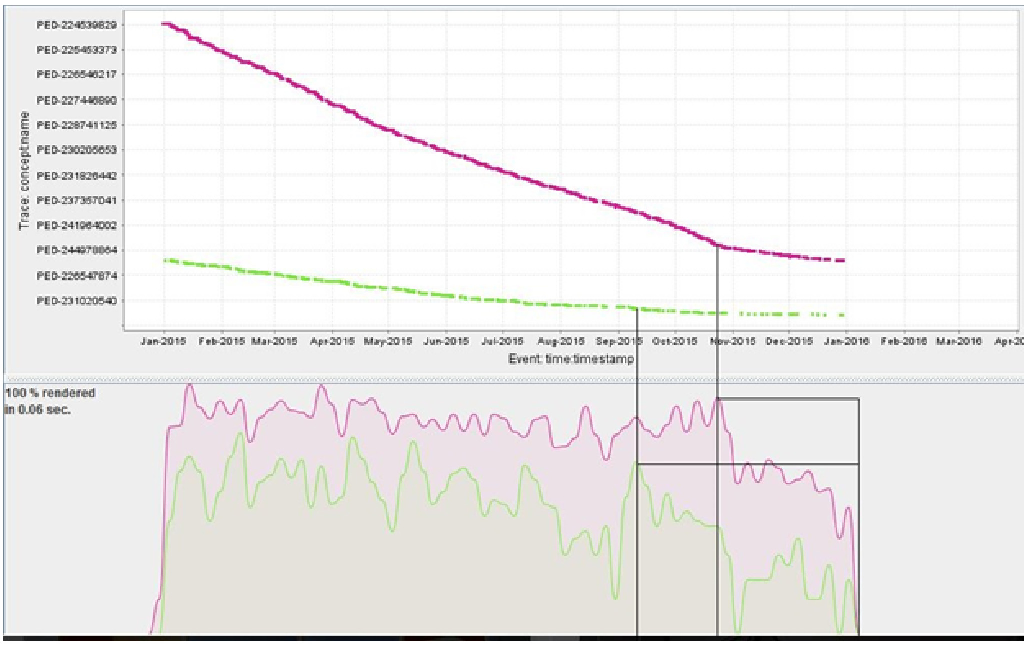


Figure 3.10: Requests over time (filtered based on Record the request)

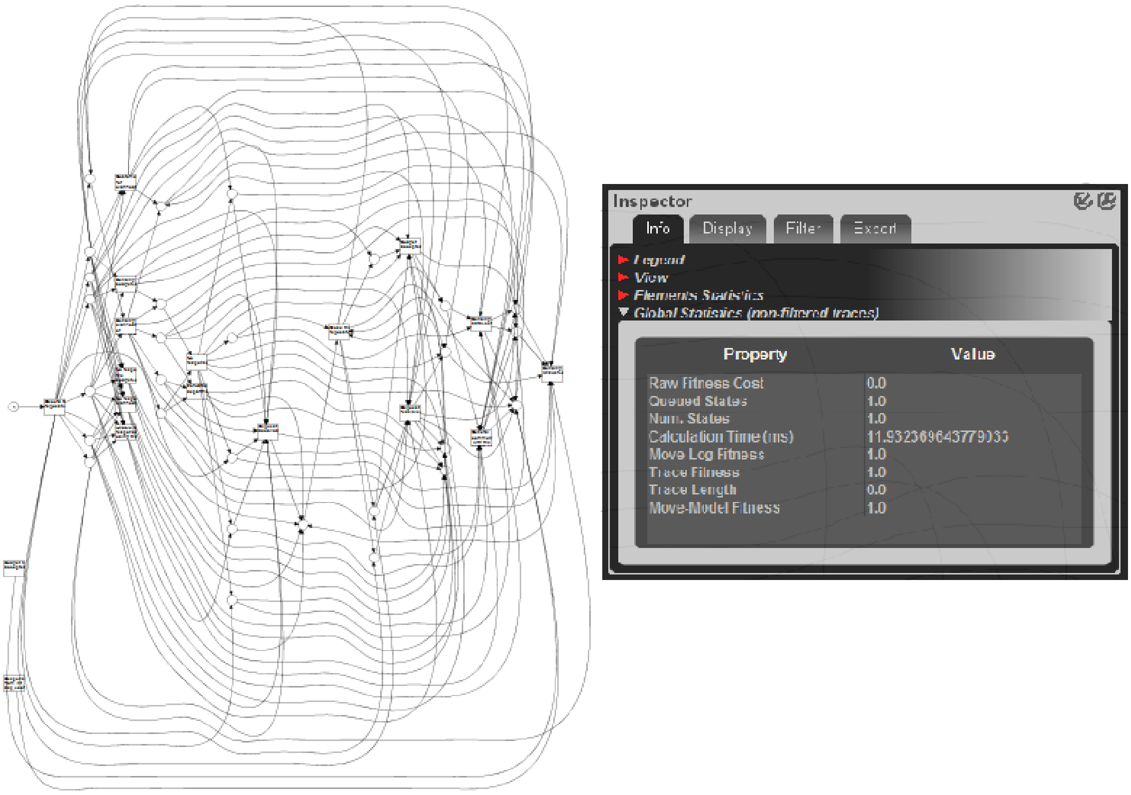
3.6.2 Compare process discovery techniques

Alpha algorithm

Using “Mine for a Petri Net using Alpha-algorithm” , The Petri Net in Figure 3.11 has been generated. It has only one start point “Record the request” because 100% of requests started with this activity, but it has no end point as there are 8 end activities (none of them is not followed by any other activity at least one time in the log). Alpha-algorithm is not considering the frequency of the activities, so all of them are presented and all connections between activities (2 activities followed each other) are presented even if it just happened only one time so model generated does not represent a process, it just represent what was done in the log producing a very complicated model which sometimes does not make any sense.

In order to avoid outliers and infrequent activities we could simplify the model by filtering the less frequent activities before running the alpha algorithm, so we would be able to choose the activities to be included and the level of simplicity.Using “Filter Log using Simple Heuristics” plug-in and considering most 98% frequent activities, we will have just one Start event 'Record the request' and only one end event 'Close the request'. The Petri Net in Figure 3.12 has been generated. The model looks more simple and fitness still high 99.5%, but alpha algorithm limitations still exist like:

- Representational bias: Alpha algorithm does not support of having duplicated



Property	Value
Raw Fitness Cost	0.0
Queued States	1.0
Num. States	1.0
Calculation Time (ms)	11.932369643779033
Move Log Fitness	1.0
Trace Fitness	1.0
Trace Length	0.0
Move-Model Fitness	1.0

Figure 3.11: Petri Net using Alpha algorithm

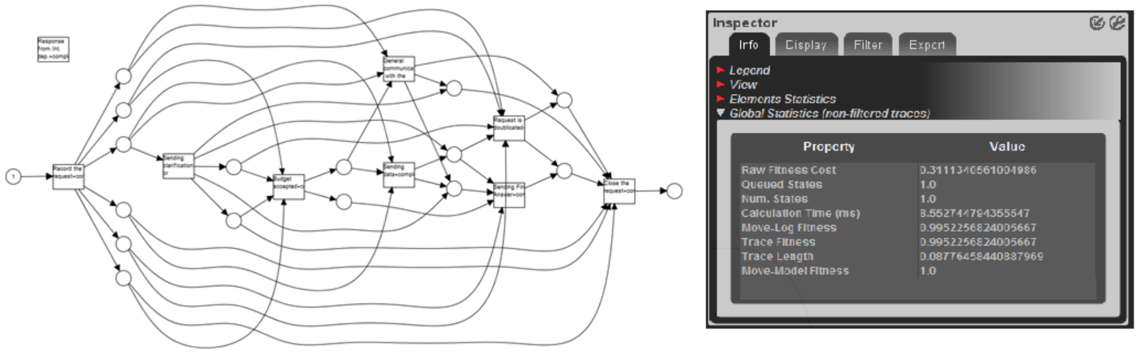
transitions (transitions with the same name) or silent transitions which are used as a solution for OR split.

- Loops of length 1 does not work with the alpha algorithm like in the case of "Response from Int. dep." Activity which in sometimes followed by itself in the log, but it is presented in the model as a separate activity (not connected to any other activities as alpha algorithm consider if an activity followed by itself, it cannot be connected with any other activities).

- The model is not sound, as the transitions "Response from Int. dep." is dead because it could not be executed at any point of time. Also, the model has a problem of proper completion as sometimes there are tokens left behind when the end stat is reached.

Heuristics Mining algorithm

As we mentioned in section 2.4.3, Heuristics miner is an improvement of the Alpha miner and it has a solution for some of Alpha miner limitations. Here we will present



Property	Value
Raw Fitness Cost	0.3111340561004986
Queued States	1.0
Num. States	1.0
Calculation Time (ms)	8.552744794355547
Move-Log Fitness	0.9952256824005667
Trace Fitness	0.9952256824005667
Trace Length	0.08776458440887969
Move-Model Fitness	1.0

Figure 3.12: Petri Net using Alpha algorithm (filter the less frequent activities)

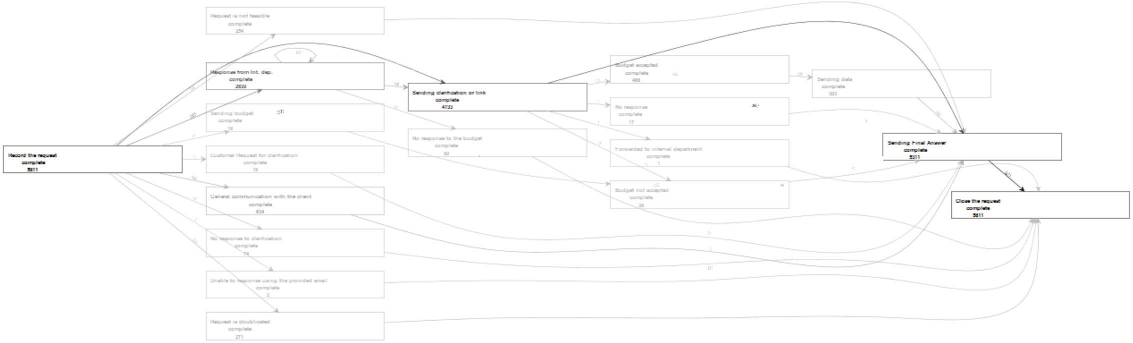

Request is not feasible complete 254
Response from Int. dep. complete 2633
Sending budget complete 18
Customer Request for clarification complete 18
General communication with the client complete 634
No response to clarification complete 74
Unable to response using the provided email complete 2
Request is doublicated complete 271
Record the request complete 5811
Sending clarification or link complete 4133
No response to the budget complete 30
Budget accepted complete 488
No response complete 17
Forwarded to internal department complete 1
Budget not accepted complete 38
Sending data complete 303
Sending Final Answer complete 5311
Close the request complete 5811

Figure 3.13: Heuristics Net

the enhancements using our real life date.

The first step is to produce Heuristics net in Figure 3.13 using "Mine for Heuristics Net using Heuristics Miner" plug-in with dependency threshold equals 100% and Relative to best equals to 0% in order to Consider only relations from dependency matrix with numbers almost equals 1. (Weijters and Ribeiro, 2011).

The second step is transferring this heuristics net into Petri net in order to check the enhancements using "Convert Heuristics net into Petri net" plug-in as mentioned in Figure 3.14.

The first observation is the model had overcome the representational bias problem. Now it has silent transitions (black transitions) that did not appear in Alpha miner model. The main idea behind producing these transitions is to avoid the problem of missing non-exclusive choice notation (OR), so the model now is allowing for non-exclusive choice by creating additional paths using silent transition as an alternative to (OR) choice.

Second observation is the model also had overcome the problem of loops of length

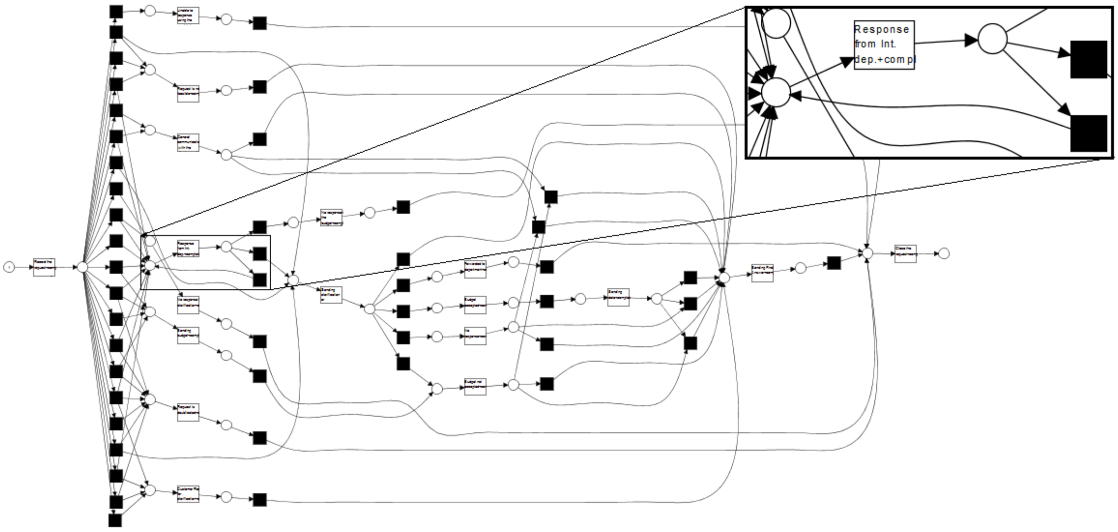


Figure 3.14: Heuristics Petri Net

1. So "Response from Int.dep." transition (which sometimes be followed by itself) now is not separated from the model and it is represented in a loop. See zoom in in Figure 3.14.

Although the model generated by Heuristics miner had overcome the soundness problem of having dead transitions, the model still not sound because the proper completion problem still exists as some silent transitions produce 2 tokens, one of them is consumed through the model and reach the end stat while the second token could stick in the model.

ETM algorithm

We had chosen to use ETM miner for two reasons. First reason, is to be able to guarantee to generate sound models because we could not generate sound models using Alpha and Heuristics miners. Second reason, to be able to choose our preferred quality criteria in case we are not satisfied with the results to come with the default parameters. In this section we will present the model comes with the default quality weights, then we will manipulate quality criteria and produce some other models in order to look for alternative models.

During our work, we had tried several ways to generate a model using ETM miner using different parameters, here we will present 2 of these ways using the parameters that gave us the best results during our trials. The first way is using predefined weights for the quality criteria and ETM miner will search for the best model satisfying these weights. The second way is to set ranges of the desired qualities for each criterion separately and ETM will return a set of models satisfying

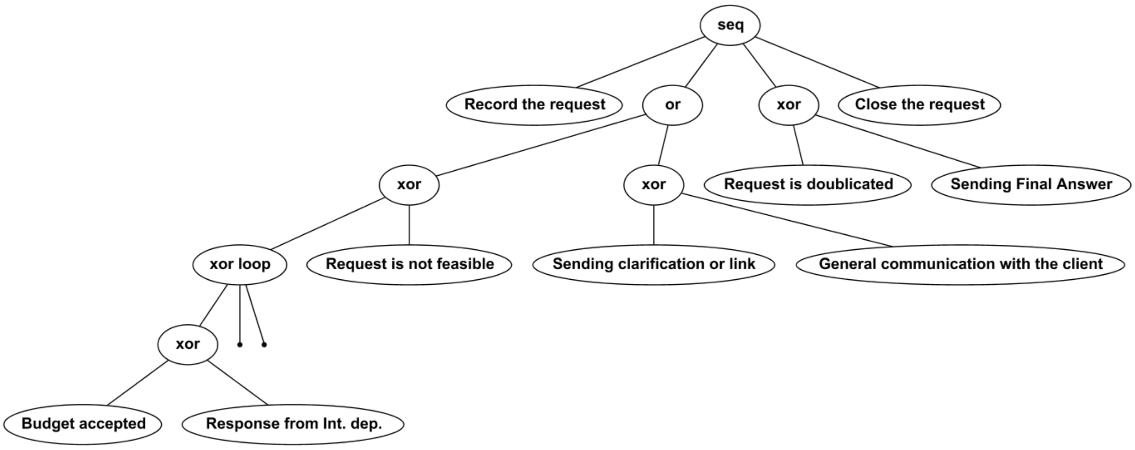


Figure 3.15: Process tree Produced by ETM miner

these ranges and user can list a preview the models based on any of the criteria.

In a first way, we had used "Mine a process tree with ETMd" plug-in using the default quality criteria in the following weights: Fitness = 10, Precision = 5, Simplicity = 1, Generalization = 1. Also, we used a constraint of 30 steady states as stop criteria in order to not to take very long time. So if the algorithm could not find better models after 30 attempts, it will return the best model so far.

Starting from generation number 70 and until generation number 99, the algorithm has reached the best model with fitness of 0.967 and the result was promising as shown in figure 3.15 .

The model makes a lot of sense in representing the actual behavior of the process in a simple way of representing the most important (or most frequent) activities in a meaningful way and good order. The process tree has 9 nodes or activities, starts with "Record the request" activity, then having non-exclusive choice (OR) between 3 activities on the first hand that either "Request is not feasible" or exclusive choice loop with "Budget accepted" or "Response from Int.dep.". Normally the process instant has one "Budget accepted" event and one or more "Response from Int.dep." which in most of the cases followed itself because the system does not record the event of sending the request to internal department and just sufficed with recording the response from internal department. Also, There could be several responses from the internal department for the same request as the response could be not satisfied by the client or the client had changed his requirements during the same request also . On the other hand of the non-exclusive choice, there could be one of the 2 activities "Sending clarification of link" and "General communication with the client". Then the process goes to an exclusive choice (XOR) between either "Request is duplicated" or "Sending final answer" as a final step before reaching the end of the process by "Close the request" activity.

Seq.	Ftiness	Precision	Generalization	Simplicity	Overall average quality
1	0.962	0.907	0.967	1.000	0.959
2	0.962	0.895	0.974	1.000	0.958
3	0.966	0.895	0.967	1.000	0.957
4	0.968	0.862	0.975	1.000	0.951
5	0.971	0.853	0.972	1.000	0.949

Table 3.1: Quality dimensions for the best 5 models in Pareto front with ETM

In a second way, we had used "Mine Pareto front with ETMd in live mode" plug-in with ranges of minimum 0.85 and maximum 1.00 for fitness and precision, minimum 0.75 and maximum 1.00 for simplicity and generalization. Also, we used a constraint of 1000 maximum number of generations as it was running in live mode and we had the option to force stop the searching whenever we satisfy with the results.

After more than 55 minutes and 1000 generation, 292 models had been generated that satisfying our quality ranges. Because the main quality dimension is replay fitness, we had choose the best 5 models with the highest Fitness values but also taking into consideration the business requirements of the desired model as Some models returns unacceptable results, for example, having the option to start with activities other than "Record the request" which is an original business requirement that should be satisfied in the model. The results are shown in table 3.1, ordered by the average quality assuming equal weights for all dimensions.

Although numbers in table 3.1 are very close to each other, but the models represented in process trees are quite different. For example, model 1,3 and 5 have 10 nodes but models 2,4 have just 9 and even the nodes are not exactly the same. Another example of the differences between models is that all models do not have a parallel relation (AND) except in models 2 and 5 between activities "General communication with the client" and "Budget accepted" that 2 activities are represented in the rest of model by non-exclusive relation. See figures 3.16 and 3.17. These differences proof that quality dimensions are helping to filter the candidate models but the opinion of business experts and business process requirements will always have the final decision.

The major drawback of ETM miner is not in the generated model but in the time consumed in generating that model. As ETM is a genetic algorithm, it is building the models based on repeating the following cycle until one or more stop criteria are satisfied (which could be a maximum number of generations, maximum time consumed and/or number of steady states). The cycle starts with generating random models, evaluate each model by computing overall quality based on the user defined weights, select the best n candidates and add them to the new generated random models, and then repeat the cycle. After satisfying the stop criteria, the

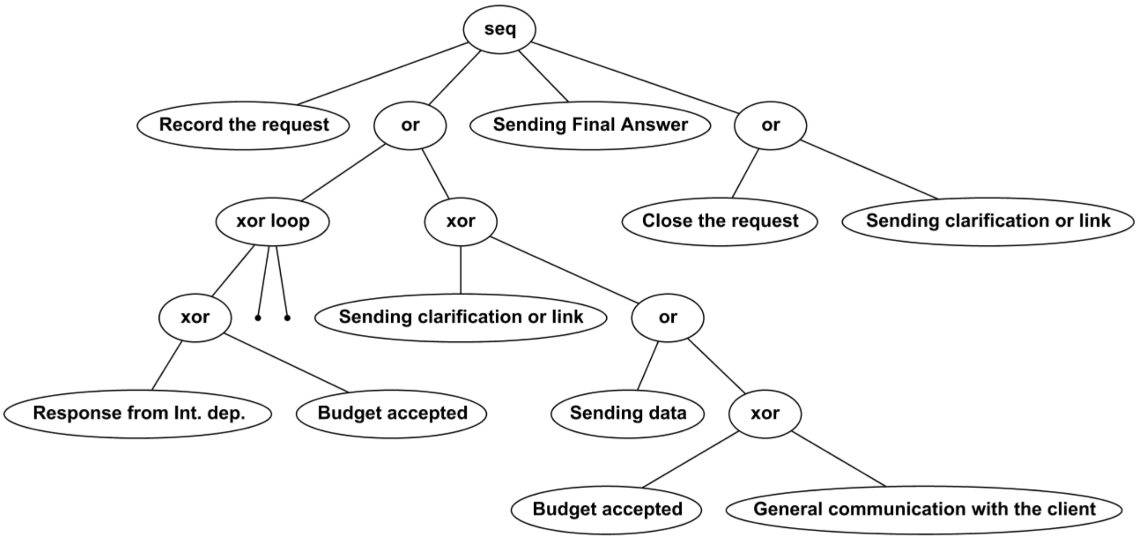


Figure 3.16: The first model shown in table 3.1

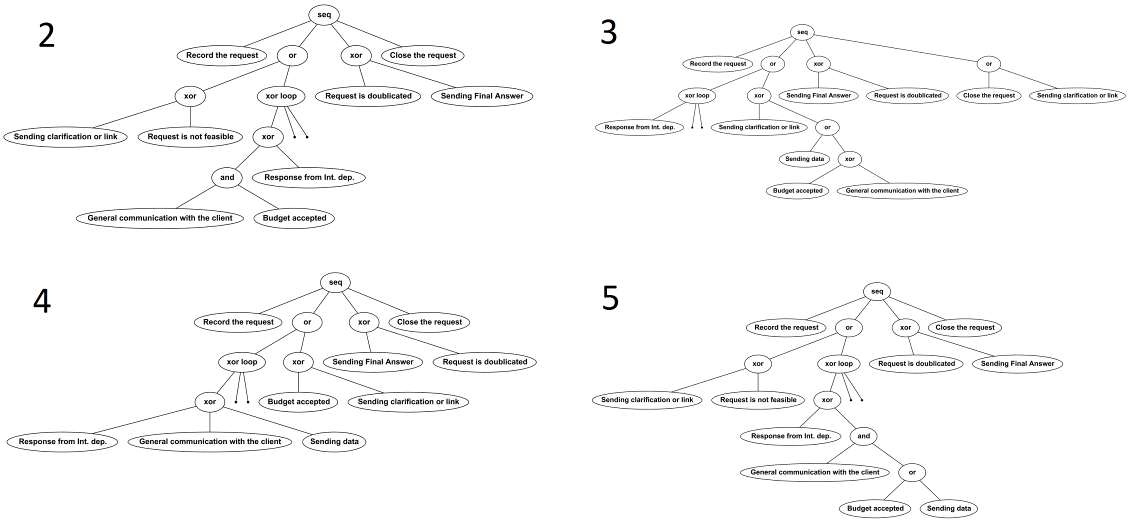


Figure 3.17: Models from 2 to 5 in table 3.1

best model in the last generation is selected.

3.6.3 Conformance checking

It does not make sense to check the conformance of not sound models because even if the fitness was satisfied and the rest quality dimensions were within acceptable ranges, the results still are not reliable and we could not depend on them. In this section we will connect events in the log to activities in the model in order to check the performance of the sound model which has been generated from ETM miner in two points of interest, Control flow and time consumed. We will use Alignment-based approach which mentioned in section 2.4.4.

Control flow

The first step was converting the process tree into a Petri net using "Convert process tree to Petri net" plug-in as shown in figure 3.18. Then aligning the model on the log file using "Replay a Log on Petri Net for Conformance Analysis" plug-in. The Petri in figure 3.19 showing the same Petri net in figure 3.18 after projected with alignment. Also, it contains the legend for the colors used and a zoom-in of two different transitions. First transition is "sending clarification or link" which is totally aligned with the model (100% Synchronous) with total number of cases went through it 4083, and the second one is "General communication with client" which have total of 1035 moves in the model, 425 of them moved in model only without occurrence in the original log, and 610 moves were Synchronous in both model and log.

Finally, Because some of the activities had been eliminated from the model, some of the moves on log (the place had been visited in cases occurred in log but could not be visited in the model as they are coming from activities that do not exist in the model) could not be projected in the model. So the projected model presented them using places, while place size shows the frequency of move log only. Also, we can know in which activities these moves on log should come from before visiting the selected place. In figure 3.20 a place named "sink 49" had been selected, so the tab named "Elements Statistics" shown 5 activities with the number of traced that each activity had a move on log in order to reach this place. For example, activity "Sending data" was missing in the model, so alignment algorithm had to make 297 moves on log in order align these 297 traces into the selected place, which would be a good point to enhance the model by adding this activity.

Time consumed

Time perspective is very similar to case perspective as it is shown in figure 3.21. Using "Replay a Log on Petri Net for Performance/Conformance Analysis", we

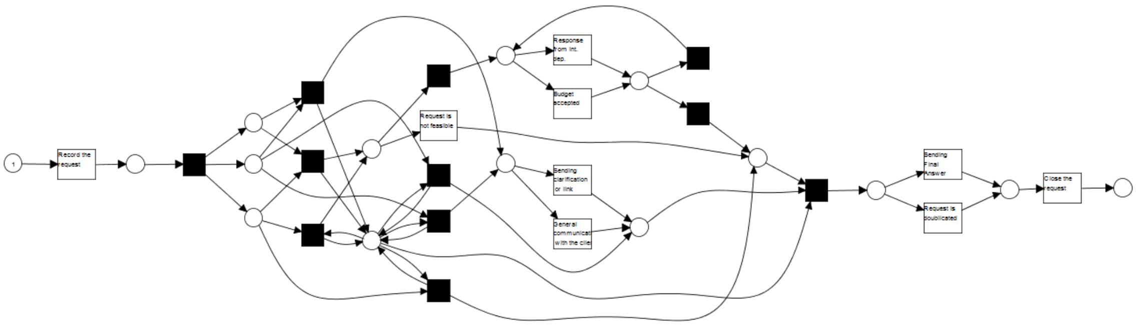


Figure 3.18: Petri net as converted from the process tree

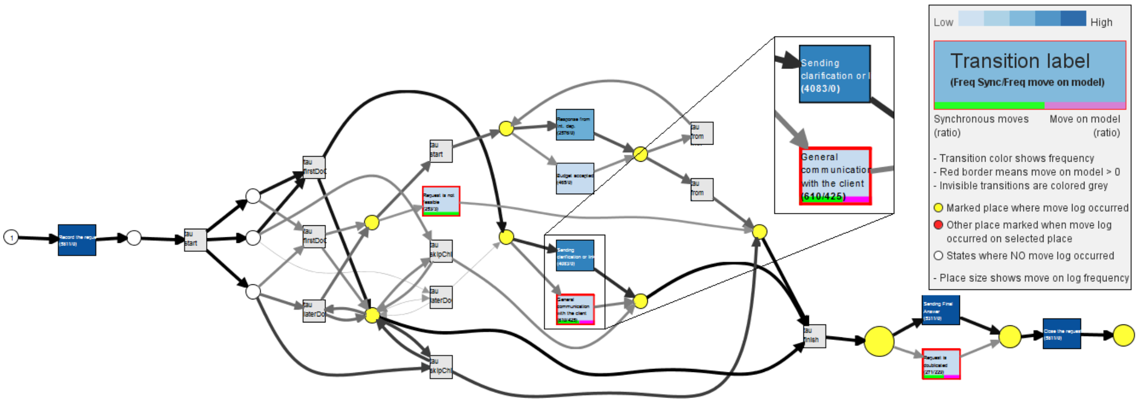
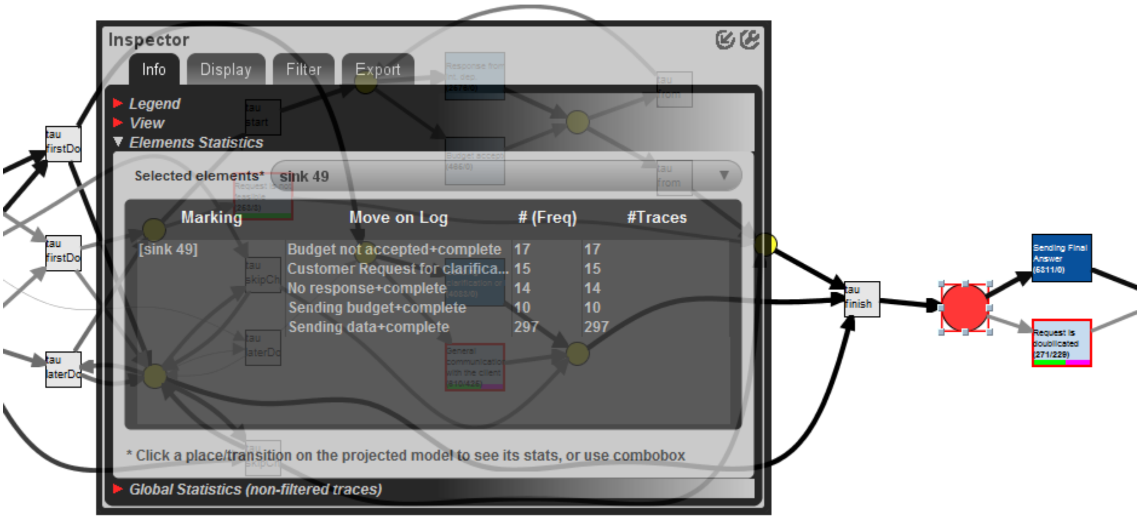

1
Record the reque (5311/0)
tau start
tau firstDoc
tau laterDoc
Request is not feasible (253/3)
tau skipChi
Response from int. dep. (2576/0)
Budget accepted (465/0)
Sending clarification or lin (4083/0)
General communication with the client (610/425)
tau from
tau finish
Sending Final Answer (5311/0)
Request is doublicated (271/229)
Close the request (5311/0)
Sending clarification or l (4083/0)
General com m unication with the client (610/425)
Low
High
Transition label
(Freq Sync/Freq move on model)
Synchronous moves (ratio)
Move on model (ratio)
- Transition color shows frequency
- Red border means move on model > 0
- Invisible transitions are colored grey
Marked place where move log occurred
Other place marked when move log occurred on selected place
States where NO move log occurred
- Place size shows move on log frequency

Figure 3.19: Petri net after projected with alignment



Inspector

Selected elements* sink 49

Marking	Move on Log	# (Freq)	#Traces
[sink 49]	Budget not accepted+complete	17	17
	Customer Request for clarifica...	15	15
	No response+complete	14	14
	Sending budget+complete	10	10
	Sending data+complete	297	297

* Click a place/transition on the projected model to see its stats, or use combobox

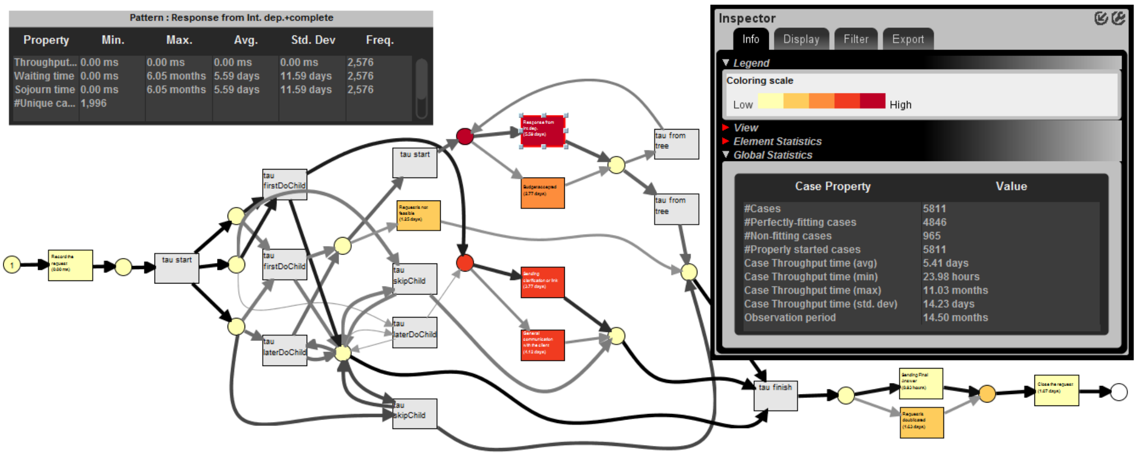
Figure 3.20: Move on log place

could generate the same process model (Petri net) but now it is projected with time consumed. That allow us to detected the bottlenecks in the process by discovering which activity takes much time to be performed.

Using the alignments, we know exactly how to relate events to the process model, and we can annotate activities with the times at which it has been observed. For now, we only know the completion times and we have no clue when they had started. So the time has been calculated for each activity as the difference between its completion time and the previous activity's completion time (which leads to the current activity). If we also had the starting time, we could distinguish between waiting time and execution time of an activity.

From the inspector box on the right-hand side in figure 3.21, we may know the coloring scale and some good information about the time consumed in order to perform the whole process from start to end. It shows that it takes on average 5.41 days with a standard deviation of 14.23 days. Also, cases have the time range from 1 day and up to 11 months.

Also from figure 3.21, we know there is a potential bottleneck in Activity "Response from Int.dep", as it appears with very dark color which means it is taking so much time to be accomplished comparing with the rest of activities. In the left-hand box, there is detailed information regarding "Response from Int.dep" activity. It shows that it takes on average 5.59 days which is even higher that the average time for the whole process. Also, it shows that in some cases it takes more than 6 months in order to be performed. So this activity needs a further analysis which



Pattern : Response from Int. dep.+complete

Property	Min.	Max.	Avg.	Std. Dev	Freq.
Throughput...	0.00 ms	0.00 ms	0.00 ms	0.00 ms	2,576
Waiting time	0.00 ms	6.05 months	5.59 days	11.59 days	2,576
Sojourn time	0.00 ms	6.05 months	5.59 days	11.59 days	2,576
#Unique ca...	1,996				

Case Property	Value
#Cases	5811
#Perfectly-fitting cases	4846
#Non-fitting cases	965
#Properly started cases	5811
Case Throughput time (avg)	5.41 days
Case Throughput time (min)	23.98 hours
Case Throughput time (max)	11.03 months
Case Throughput time (std. dev)	14.23 days
Observation period	14.50 months

Figure 3.21: Performance analysis using time perspective

should be done with the cooperation of business owners in order to know the origin of the problem.

3.6.4 Extend the model

Extending the model or analyzing the model in some different perspectives rather than the process perspective which had been mentioned during this case study, is an important factor in order to understand the process in different points of views.

In this section, we will consider Organizational perspective mentioned in section 2.6.1.

As a reminder, our event log has 81 originators or resource distributed in two roles "Customer service" and "Internal department". "Customer service" role include 13 resources coded with names from Agent00 to Agent12. "Internal department" role include 68 resources coded with names from Cons00 to Cons71. "Internal department" resource are responsible for performing only one activity which is "Response from Int.dep." and the rest of activities are being performed by "Customer service" resources. Each case has only one resource from "Customer service" plus one resource from "Internal department" in cases required feedback from internal department resources.

In ProM 6 (Prom, 2010) there are 5 plug-ins responsible for mining for the social network using event log. "Mine for a handover-of-work Social network", "Mine for a Reassignment Social network", "Mine for a Similar-Task Social network", "Mine for a Subcontracting Social network", and "Mine for a Working-Together Social network". The idea behind these plug-ins had been mentioned in section 2.6.1.

As we already know from the log and process, each case has maximum one resource from each department and "Internal department" resources are responsible for only one activity, so there is no need to mine for Reassignment or Similar-Task as they will not add any new information.

The most suitable plug-ins for our work are "Mine for a handover-of-work Social network", "Mine for a Subcontracting Social network", "Mine for a Working-Together Social network".

Before start in presenting the social networks, there are 2 main parameters, that we will consider in each social network preview, Layout, and rank.

In layout parameter, there are 6 different types of layouts that we can choose from them for the most suitable presentation to our case. In our work, we had used "ISOMLayout" because it is working on arranging the nodes according to homogeneous clusters, so the nodes related to the same clusters will be presented in the nearest distances to each other as possible and they will have the same color.

Ranking parameter has the tree main social network analysis measures introduced in section 2.2.3, and during our work we had used "Degree" ranking in order to be able to distinguish between resources that are connected with higher number of other resources (in the center of the network) and resources with fewer connections with others (in the edges of the network).

Using "Mine for a handover-of-work Social network" plug-in, we can extract social network in figure 3.22. It shows that resources with very high number of connections are in the center of the network, that resources are not very interested to be studied as they are representing the normal scenario of the applicability for any resource from "Customer service" to be Connected to any resource from "Internal department", but the resources on the edges of the network are more interested as they are representing the special cases where resources are always contacted with a specific resource(s).

In bottom left corner, cons39 is selected and all the connected resources are highlighted. it shows that Cons39 is connected with just 4 resources, but always receiving tasks from 3 of them (Agent01, Agent03, Agent08) and handover tasks to 2 resources (Agent01, Cons41). It is not normal to have 2 resources with the same type connected to each other (Cons39 and Cons41) but this happened in some cases when the case was delegated to an internal resource, then for any reason this case is delegated to another internal resource.

Also in the down right corner, it shows that Agent00 and agent12 are not handing over any work or receiving work from others (they are always working alone).

Using "Mine for a Subcontracting Social network", we got the social network in Figure 3.23. It is very close to the network from handover in figure 3.22, but here the relations between nodes are in one way (Customer service resources are always sending and internal department resources are always receiving), which makes perfect sense regarding the process. As we can see on the right-hand side, that Cons39 now is only connected with one resource which is Agent01 who is always executing tasks before and after Cons39 execute his task. and for sure the direction is one way from Agent01 (Customer service) to Cons39 (Internal department).

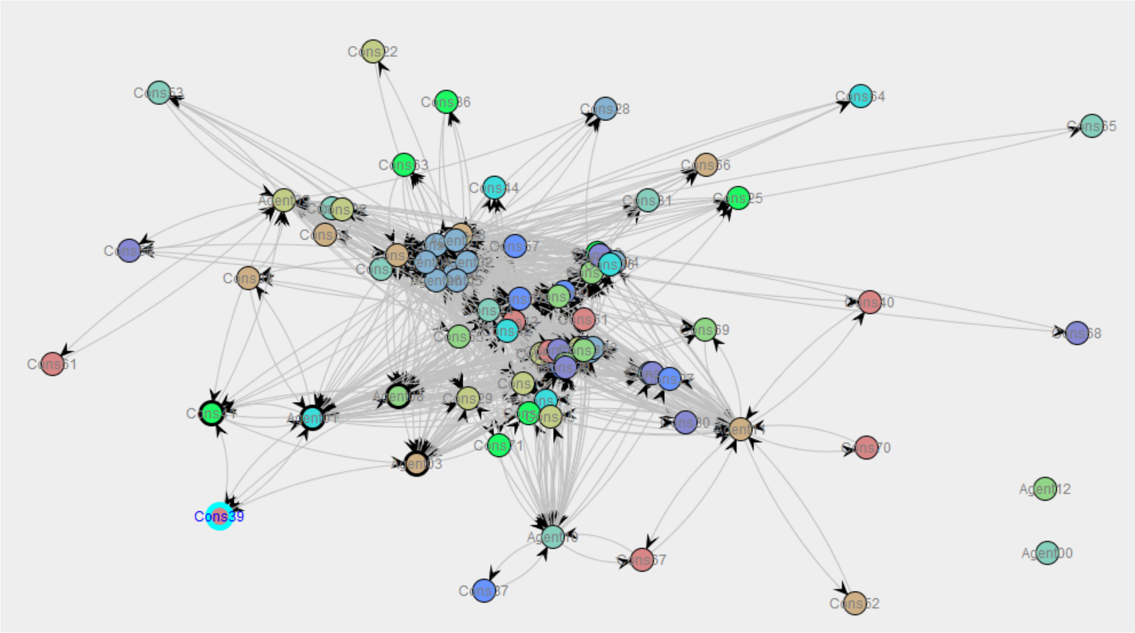


Figure 3.22: Mine for a handover-of-work Social network

Also on the upper-left corner, we can see 5 isolated resources: Agent00, Agent12, Cons12, Cons30, and Cons58. These resources are not subcontracting or being subcontracted by any other resources. We already know from Handover network that Agent00 and Agent12 are not communicating with any other resources, but now additional 3 resources from Internal department had been isolated as they had only one-way connection in handover network.

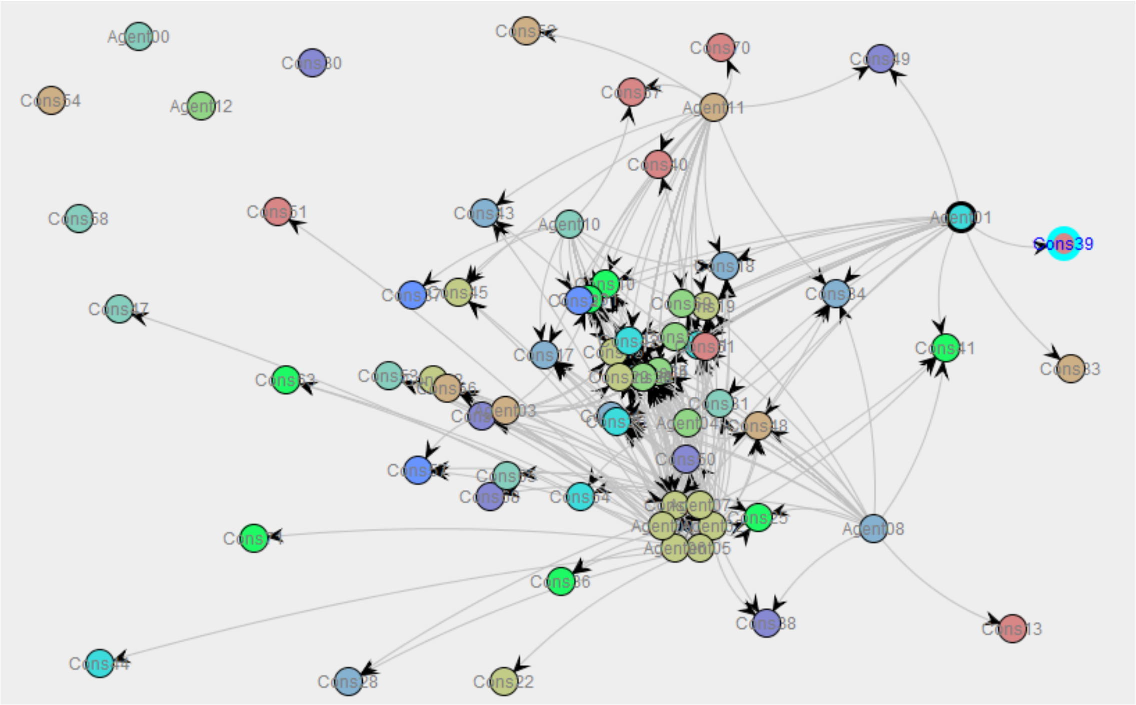


Figure 3.23: Mine for a Subcontracting Social network

Chapter 4

Discussion of results

4.1 Introduction

In this chapter, the conclusions and results we have got during this work will be discussed starting from methodology review, which has the conclusions we have made regarding methodologies and algorithms discussed in chapter 2.

After that, we will discuss the results we have achieved during the case study execution in chapter 3 for the different algorithms and perspectives had been used.

4.2 Methodology review

There are three main functions that cover the functionality of Process Mining: Play-out (Simulation), Play-in (Process discovery), and Replay (Conformance analysis). Starting with process discovery, which has several algorithms that allowing to extract a model from an event log. These algorithms should consider (1) Ability to represent all different kind of notations such as concurrency, loops, silent actions, duplicate actions, OR-splits/joins, non-free-choice behavior and hierarchy. (2) Ability to Deal with Noise by preventing infrequent behaviors from being presented in the model. And (3) Ability to not to restrict the model just on the observed behaviors, but also should be flexible enough to accept new behavior which did not appear in the original event log. Also the model the generated model should satisfy a minimum threshold of the four quality dimensions: fitness, simplicity, precision, and generalization. the model should be able to balance between all four dimensions (or forces). So the model should consider as much as possible behaviors shown in the log, but in the same time, the model should be kept simple. Also, the model should be able to balance between being precise and not allowing too many behaviors not seeing in the log, and being general and do not restrict only to behaviors in the model.

On the other hand, soundness is very important factor to determine the quality of the model. It could be the most important factor used to determine if the model is reliable enough in order to rely on it. The model is sound, if and only if all its three properties are fulfilled. These properties are (1) Option to complete: no bottlenecks, (2) Proper completion: no tokens left behind after reaching the end stat, and (3) no transitions being dead. During our work, we had considered only the results from ETM algorithm, mainly because it guaranteed soundness, unlikely, Alpha miner and Heuristics miner that we had mentioned in order to present the most basic algorithm that fits the gap between event logs, and the process model, and also to proof that they are no guaranteeing soundness.

The alignment-based approach, is the approach that has been used in conformance analysis, because of its advantages of the other approaches as it is directly related to modeled behavior, very flexible as we can use any cost structure, it can offer a detailed diagnostics, and the aligned model could be used easily in further analysis.

There are so many tools that support Process Mining, the main and the most powerful (in terms of available plug-ins) non-commercial tool is ProM. During our work, we had used PromM version 6.5.1 which was the latest version at that time.

The control-flow model can be extended considering one or more perspectives we had mentioned in section 2.6. our main focus in this work was in organizational perspective which had been implemented in the case study using ProM social network package. Although the powerful of this package in mining for the social network in different kind of methodologies and in the nice graphical representation, it is only giving a high level results not giving well-detailed results. There is no way to know the relations between nodes in a numerical way to be used in any further analysis. The result is just presented in a graphical form and can not be extracted from ProM with any form except as a picture. There is no way to configure clusters or even know how it works.

The main idea of Process Mining is to define the missing link between business process analysis and data analysis techniques. This could be applied for example by discovering the process model based on the real data and independently from the predefined business process. But, How far we can work independently from business requirements? In our opinion, Process Mining (like all other data science) should always consider business requirements and need in all steps of any project lifecycle.

Previous knowledge of how business process should look like is a mandatory even in the very early stage of choosing which process needs to be discovers, going through choosing the related variables and asking the right questions, until choosing the best model which is most describing the real data, but also satisfying predefined business requirements and needs.

The same is applicable for all stages of Process Mining. Even in conformance analysis and exploring bottlenecks, business process previous knowledge would consider if it is a real bottleneck of just a normal behavior which is understandable and acceptable by the business.

Also regarding other perspectives like organizational perspective, knowing the type of resource and what are the activities he/she should/shouldn't execute is a very important factor to understand the social network and relations between resources.

In the following section 4.3. Real examples of such cases where the previous knowledge plays a major role in making decisions and interpreting the results, will be mentioned.

4.3 Case study results

The data has been received from Statistics Portugal (INE) in order to perform this work was an event log contains information regarding requests received by INE customer service from clients requesting some kind of information. The event log variables, values, the number of records starting and ending dates have been explained in details in section 3.5. Also, we had mentioned some results from different views in ProM that allowed us to know the most frequent cases, shortest and longest traces, and cases and activities behavior over time using dotted charts. Review figures from figure 3.1 to figure 3.10

Later, three Process discovery techniques had been introduced, in order to discover the best process model.

The First algorithm was Alpha algorithm which we choose because it is the simplest and earliest algorithm used to discover process models. The model was a very complicated model because Alpha algorithm does not consider frequencies, so all connections between activities are presented even if it just happened only one time. and even when we had filtered the log based on most frequent activities, the limitations of the Alpha algorithm still existed which were (1) Representational bias: Alpha algorithm does not support of having duplicated transitions (transitions with the same name) or silent transitions which are used as a solution for OR split. (2) Loops of length 1 do not work with alpha algorithm like in case of "Response from Int. dep." Activity which in sometimes followed by itself in the log, but it is presented in the model as a separate activity (not connected to any other activities as alpha algorithm consider if an activity followed by itself, it cannot be connected with any other activities). (3) The model is not sound, as the transitions "Response from Int. dep." is dead because it could not be executed at any point of time. Also, the model has a problem of proper completion as sometimes there are tokens left behind when the end stat is reached.

The second algorithm was Heuristics Mining algorithm. As it was an improvement of the Alpha algorithm, we decided to give it a try after the bad results we got. Fortunately, the model was better and had overcome some of the Alpha algorithm limitations like representational bias and loops of length 1, but the model still not sound because the proper completion problem still exists, so we can not rely on that model too.

The third algorithm was ETM algorithm because it guarantees to generate sound models. Also it has another important feature where either we can configure weights for the four quality dimensions in order to generate models according to our preferences or we can set ranges of the desired quality for each criterion separately and ETM will return a set of models satisfying these ranges, and later we can choose from them while the qualities are calculated and presented for each model. The major drawback of ETM miner is not in the generated model but in the time consumed in generating that model.

After that, we used the model generated from ETM algorithm for conformance checking by replaying the event log on the model and check the alignment (which activities are perfectly matching the model? and Which are not?). The aligned Petri net shown in figure 3.19 stating that most of the activities have a perfect match between log and model (100% Synchronous) except for activities "General communication with the client" and "Request is duplicated". Some cases had deviated in these two activities and algorithm had to make some moves on the model in order to avoid the deviation.

Also, a performance check had been done based on the time consumed for each activity and for the overall process. A potential bottleneck in Activity "Response from Int.dep", shown in figure 3.21 as it shows that it takes on average 5.59 days which is even higher that the average time for the whole process which is 5.41 days. We call it potential bottleneck as we need a confirmation from business owners in order to know the origin of the problem.

Finally, the model was extended by adding the organizational perspective. we had generated two social networks based on the type of the relation between resources. First, we have to mention that we already know according to previous information that internal department resources are responsible for performing only one activity which is "Response from Int.dep." and the rest of activities are being performed by customer service resources. Also, each case has only one resource from "Customer service" plus one resource from "Internal department" in cases required feedback from internal department resources.

The first social network was based on the handover-of-work which was shown in figure 3.22. It shows the customer service resources with a very high number of connections with internal department resources are centralized in the network while resources on the edges (most of them are internal department resources) are only communicating with a specific customer service resources which is not a business requirement.

The second social network was based on Subcontracting which was shown in figure 3.23. It is very similar to the network from handover, but here the relations between nodes are in one way (Customer service resources are always sending and internal department resources are always receiving). so arrows are always in the direction from "Agentxx" to "Consxx" otherwise it would be irregular behavior which needs further analysis.

Bibliography

Buijs, J. C. A. M., van Dongen, B. F., and van der Aalst, W. M. P. (2012). *On the Role of Fitness, Precision, Generalization and Simplicity in Process Discovery*, pages 305–322. Springer Berlin Heidelberg, Berlin, Heidelberg.

CROSS, R. (2001). Knowing what we know: Supporting knowledge creation and sharing in social networks. *Organizational Dynamics*, 30:100–120.

D. Grigori, F. Casati, U. D. and Shan, M. (2001). Improving business process quality through exception understanding, prediction, and prevention. In *Proceedings of 27th International Conference on Very Large Data Bases.*

Evelson B., N. N. (2010). Business intelligence. forrester research papers business process professionals. Technical report, Forrester.

Hansen, D. and Shneiderman, B. (2009). Analyzing social media networks: Learning by doing with nodexl. `http://archives.cerium.ca/IMG/pdf/Les_analyses_de_reseaux_avec_NodeXL.pdf`.

HU, P. C. (2008). Visual representation of knowledge networks: A social network analysis of hospitality research domain. *International Journal of Hospitality Management*, 27:302–312.

INE (2015). statistics portugal. `https://www.ine.pt/xportal/xmain?xpgid=ine_main&xpid=INE`.

M. Sayal, F. C. and Dayal, M. S. U. (2002). Business process cockpit. In *Proceedings of 28th International Conference on Very Large Data Bases.*

Microsoft (2016). Dynamics crm. `https://www.microsoft.com/en-us/dynamics/crm.aspx`.

Moses, N. and Boudourides, M. (2001). Electronic weak ties in network organizations. In *4th GOR Conference.*

Muehlen, M. Z. and Recker, J. (2008). How much language is enough? theoretical and practical use of the business process modeling notation. *Proceedings of* *the 20th International Conference on Advanced Information Systems Engineering (CAiSE'08)*, 5074:465–479.

Prom (2010). Prom 6 tutorial. `http://www.promtools.org/prom6/downloads/prom-6.0-tutorial.pdf`.

R. Agrawal, D. G. and Leymann, F. (1998). Mining process models from workflow logs. In *Sixth International Conference on Extending Database Technology.*

Rozinat, A. (2014a). How process mining compares to business intelligence. `https://fluxicon.com/blog/2011/01/how-pm-compares-to-bi/`.

Rozinat, A. (2014b). How process mining compares to data mining. `https://fluxicon.com/blog/2011/02/how-process-mining-compares-to-data-mining/`.

Song, M. and van der Aalst, W. (2008). Towards comprehensive support for organizational mining. *Decis. Support Syst.*, 46(1):300–317.

Ting, K. M. (2010). *Confusion Matrix*, pages 209–209. Springer US, Boston, MA.

Van der Aalst, W. (2011). *Process Mining: Discovery, Conformance and Enhancement of Business Processes.* Springer.

Van der Aalst, W. (2015). Extracting event data from databases to unleash process mining. In *BPM - Driving innovation in a digital world*, pages 105–128. Springer.

van der Aalst, W. and Song, M. (2004). Mining social networks:uncovering interaction patterns in business processes. In *International Conference on Business Process Management (BPM 2004)*, Berlin, Germany.

van Eck, M. L., Buijs, J. C. A. M., and van Dongen, B. F. (2015). *Genetic Process Mining: Alignment-Based Process Model Mutation*, pages 291–303. Springer International Publishing, Cham.

Weijters, A. J. M. M. and Ribeiro, J. T. S. (2011). Flexible heuristics miner (fhm). In *Computational Intelligence and Data 565 Mining (CIDM)*, 2011 IEEE Symposium on (pp. 310–317).

W.M.P. van der Aalst, H.A. Reijers, A. W. and Others (2007). Business process mining: An industrial application. *Information Systems*, 32(5):713–732.

W.M.P. van der Aalst, H. R. and Song, M. (2005). Discovering social networks from event logs. *Compute. Supported Coop. Work*, 14:549–593.

W.M.P. van der Aalst, B.F. van Dongen, J. H. and Others (2003). Workflow mining: A survey of issues and approaches. *Data and Knowledge Engineering*, 47(2):237–267.