Universiteit Hasselt | Campus Diepenbeek | Agoralaan Gebouw D | BE-3590 Diepenbeek
Universiteit Hasselt | Campus Hasselt | Martelarenlaan 42 | BE-3500 Hasselt

2011
2012

FACULTY OF BUSINESS ECONOMICS

Master of Management: Management Information Systems

Masterproef

Performance analysis of business processed through the use of process mining

Promotor :
Prof. dr. Koenraad VANHOOF

Veronika Boyanova

Master Thesis nominated to obtain the degree of Master of Management , specialization Management Information Systems

2011
2012

FACULTY OF BUSINESS ECONOMICS

Master of Management: Management Information Systems

Masterproef

Performance analysis of business processed through the use of process mining

Promotor :
Prof. dr. Koenraad VANHOOF

Veronika Boyanova

Master Thesis nominated to obtain the degree of Master of Management , specialization Management Information Systems

Acknowledgements

First and foremost I would like to offer my sincerest gratitude to Prof. Dr. Koen Vanhoof for his support and guidance during my master project development. I am grateful to him for appointing his assistant Jo Swinnen to guide me during my whole thesis improvement.

I am very thankful to my advisor Jo Swinnen for her continuous support of my master thesis study, for her constant patience, enthusiasm and motivation during all the months I was writing this thesis.

I also thank all my friends and classmates whose support I have been receiving during the whole writing period and who provided me with motivation and help all the time.

Last but not least I would like to thank my family for supporting me throughout my whole studies in the Hasselt University and encouraging me during my whole master year

Thank you all!

Preface

Business organizations nowadays need to constantly monitor and develop their processes. The environment gets hard to be controlled by management in enterprises and quick measures for any problem have to appear. Fast, flexible and qualitative actions have to be ensured for the competitive advantage support.

Business managers start looking inwards the organization in order to improve the processes inside and gain customer satisfaction. Process mining is a relatively young research area that aims to support these issues by digging into processes in order to analyze them throughout and release sufficient result from the mining. Different techniques are available and may be used to help the monitoring, analyzing, development and improvement of the tasks in a company.

The goal of this master thesis is to present the methods that process mining provides and to analyze how they can be used for analyzing the performance of business processes. There are different plug-ins introduced by the ProM framework – an open source software which supports the process mining techniques. These plug-ins support the mining and analysis of business processes. Some of them are presented in details in this thesis and are used for the performance evaluation of a process occurring in a real life organization. A case study in a communication company is presented to reveal the abilities of the framework to monitor and give opportunity for development of processes.

Contents

List of figures

List of tables

Chapter 1 – Research plan

1.1 Research objective

Nowadays many organizations face the difficulty to keep a competitive advantage and recently even to retain on the market. When we add the quite famous financial crisis and the threat of new competitors that emerge in every sector of the economy – it becomes even worse. Consequently, every company tries to find a way to struggle in this environment. Moreover, it is quite clear that in every sphere as a whole and particularly in all the industries and business environment, constantly changes take place. The main challenge for companies therefore is flexibility and the way they will respond to changes.

Presently, many business processes are supported by information systems which help coordinating their performance and execution. Workflow systems, for example, focus on the automation of business processes. They manage and define series of tasks within an organization to help the production and realization of final outcomes. They automate redundant tasks and ensure uncompleted tasks in the chain are followed up (Van der Aalst & Van Hee, 2002).

As a result of the appearance of disciplines such as Business Process Management (BPM) or Business Process Reengineering (BPR), organizations have started to focus deeply on their activity chains to increase productivity, flexibility and reduce costs. BPM is the one to combine knowledge from information technology and management sciences and apply them to operational business processes. It has a broader scope than the workflow management, for example as BPM goes from automation through analysis to management of business processes. Effectiveness and efficiency are key words here. BPM is usually referred to as "process optimization process" and is often related to Business Intelligence (BI) (Vom Brocke & Rosemann, 2010).

BPR on its place is the analysis and design of business processes. It is the fundamental rethinking and redesign of business processes to achieve dramatic improvements in critical contemporary measures of performance such as cost, quality, service and speed [4]. It often makes radical transformations and changes attitudes and behaviors of people in a company, only executing a limited number of activities (Coulson-Thomas, 1992).

Process-Aware Information Systems (PAISs) include the traditional workflow systems but also Enterprise Resource Planning (ERP), Customer Relationship Management (CRM) systems, call central software, etc. that provide more flexibility or support specific tasks. However, they cannot analyze the processes they are used in and respectively are not involved in the management of these processes (Van der Aalst, 2011).

Although process mining is a young research area, it complements all the above mentioned systems and disciplines as it extracts not just some historical or current information but valuable event data in a meaningful and purposive way so that deep insights can be provided, bottlenecks identified and violations or frauds detected. In fact this is its overall goal (Van der Aalst, 2011).

Process mining is a powerful business tool that can really help the management of a company to overcome the difficulties in both their external and internal environment. With its help stakeholders can observe, monitor, control and improve every process in the organization. Bottlenecks in the tasks chain can be detected and processes can be redesigned and developed. A business process can be simply defined as a group of tasks or steps that produce a certain result for a customer (service or activity). This can be, for instance, a granting of a loan or processing of orders in a factory. A lot of challenges exist in processes – they are often not delivered on time, they are complex and expensive, no adequate documentation is presented concerning them and structured components can be hardly captured (Van der Aalst, 2011). This is where process mining can interfere to solve these problems that reflect negative on business activities. It gives the opportunity for a process to be managed – from observing it, through extracting essential information to redesign it in order to develop and improve it. That is what organizations need to realize and start to use this technique to gain a competitive advantage and improve their business cycles and activities. For this purpose the open-source framework, ProM, with all kinds of process mining algorithms was created at the Eindhoven University of Technology (Van Dongen, De Medeiros, Verbeek, Weijters, & Van der Aalst).

In general process mining is a management technique to analyze processes through the use of event logs (Van der Aalst, 2011). Normally the latters are registered by an information system and record each activity that has occurred in a process (e.g. complete a loan application form, check a plane ticket, request insurance) as well as additional event data as the resource (person or device that executed it), the timestamp (when exactly it was executed or how long its realization took place) or some event elements (e.g. the size of an order, identification number of a request, etc.). The idea of this kind of mining is to extract knowledge from the logs and use it to discover, understand, monitor, re-design, develop and improve business processes. What is important here is this is all executed on the base of reality, no assumed processes, no artificially constructed. The actual process is mined and thus a process model can be evaluated and enhanced (Van der Aalst, 2011). A process model is a description of a process that usually prescribes how things should, could or must happen (Rolland & Pernici, 1998). Roughly, it is an anticipation of what the process will look like. When related to reality, a process model will have essential value for the end users so a lot of attention should be paid on this alignment. With the help of process mining, models based on facts can be extracted. Not just one base model of the process can be created but a variety of views on the same reality at different abstraction levels can be provided (Van der Aalst, 2011).

According to the theory, there are a few aspects of the usability of a model. One main aspect is the performance analysis of a business process. With the use of different process mining techniques factors that influence service levels, response times, etc. can be discovered (Van der Aalst, 2011).

In the next lines and chapters as a whole, we will give a special attention to the above mentioned performance analysis. The performance of a chain of tasks can be defined in various ways and seen in different dimensions and namely time, cost and quality. For all of them various Key Performance Indicators (KPIs) are identified and they can help for the optimization of a business process model. Consequently, the aim of the analysis of the performance of the business processes is to improve them in respect to the three dimensions mentioned (time, cost and quality). This can be achieved through the use of different analysis techniques presented in the field of process mining. For instance the simulation technique can be used to capture the factors that influence the response time of an activity in a company (Van der Aalst, 2011)

The important thing here is that due to process mining and its pure connection with reality, the performance analysis can be based on facts and event data. Thus, performance analysis by using process mining can help managers in organizations realize why processes fail. The different process mining algorithms can assist in analyzing the business process performance. In this way companies get a better insight in their processes and performance can be improved.

1.2 Main research question

Considering all the above mentioned, the research question of this master thesis appears:

How can process mining be used to analyze the performance of business processes in organizations?

The aim of this master thesis is to find out whether process mining can be used in analyzing the performance of business processes and to what degree it can be used. It will be discovered and explained if process mining is valuable for performance analysis and how exactly it can support processes examination.

The problem statement formulation is based on the recent research in the field of process mining and the need of exploration of process performance in organizations. In reference with this question existing tools and functionality in the area of performance analysis will be examined and presented according to the literature that exists in the area, some examples will be given to show the benefit of such analysis and how it is supported by process mining techniques. Normally in literature the insight of processes presents its correct run, e.g. excluding anomalies, deadlocks and live locks. In real life some deep analysis is necessary to be seen how processes are actually performing in an organization. On the base of some examples and case studies, a few conclusions will be made and a few pieces of advice given.

In order to answer the research question and outline the structure of the whole thesis, some sub questions will be formulated and comments on them will be given as well as some main definitions that will be necessary as preliminaries of the whole subject. The sub questions will relate process mining to performance analysis and its benefit for the business organizations.

Next, the essence of process mining will be explained which requires the answer of the following questions and some short explanations to them:

1.3 Sub research questions

1.3.1 What is an event log?

The process mining technique can only be used if an event log for a certain process is recorded. Here an explanation of what is an event log will be given.

The event log is a piece of information, recorded by information systems that reflects an activity executed in a certain process which has taken place in an organization. The log can comprise not just the essence of what has been done but also additional information for the event such as who executed it (the resource), the time it was performed (timestamp, i.e. exact date and time) or some data elements recorded together with the task. (Van der Aalst, 2011). A lot of data like this is not structured but just scattered over the information systems. Therefore it is a big challenge for people in the organizations to deal with such a big and unstructured amount of data, to extract the right information they will need to understand the processes going through the company. In such situations, the process mining has to take place.

Chapter 2 will explain event logs in details and will show examples of event data as this is the source for the process mining techniques.

1.3.2 What is process mining and what are the goals of this technique?

After the necessary input for process mining to take place was shortly explained, we can go its essence. Its main concepts will be explained in details in Chapter 2 of this thesis.

Process mining extracts knowledge from the event data, stored in the logs and exploits it in a meaningful way. It obtains process related information and use different techniques to analyze it so that activities can be optimized and improved. The main goal of process mining is to discover, monitor and develop real but not assumed business processes. The mining process relates the actual event data to a constructed process model. With its help they can be compared, deviations can be detected, weak places found and repaired. In this way the conformability of process models with reality can be discovered and evaluated so that they can be redesigned and improved in different ways if necessary. With the help of process mining experts in an organization can look deep into the processes, notice the bottlenecks (an activity in a process which limited capacity reduces the capacity of the whole chain) and errors that has occurred. Deviations from a process model can be analyzed and quality of the process improved (Van der Aalst, 2011), (Gunther, Rinderle-Ma, Reichert, Recker, & Van der Aalst, 2007), (www.processmining.org). In Chapter 2 the essence of process models will be explained as well as the link between them and the real processes that take place in a company together with their recorded data. This link is made with the help of process mining. The three perspectives of mining – process, organizational and case perspective are going to be presented. According to the relation between event logs and process models three types of process mining exist.

1.3.3 Which are the types of process mining?

In Chapter 2 the different types of process mining will be explained. There are three main types, i.e. *process discovery*, *conformance checking* and *enhancement*. Performance analysis can be mostly categorized underneath enhancement as this type adds different perspectives to the process models and gives an opportunity for insight of the activities and their detailed examination and development (Van der Aalst, 2011).

1.3.4 What is ProM? What is a plug-in? What kinds of plug-ins are included in ProM?

Further, the framework ProM with all kinds of process mining algorithms will be shortly defined and Chapter 2 will reveal the precious opportunities it gives companies to deal with their processes.

ProM is a special open-source tool for process mining techniques to be applied to processes. It is a kind of software system that was created in the Eindhoven University to help students and experts deal with the different algorithms presented by the process mining theory (Van der Aalst, 2011), (www.processmining.org).

ProM gives the opportunity to include new plug-ins without the need of configuration of the framework. That is why in literature it is called “pluggable” environment (Van der Aalst, 2011). In general, a plug-in is an implementation of an algorithm that can be used in some way in the area of process mining. Currently, there are five types of plug-ins in the framework: *mining*, *export*, *import*, *analysis* and *conversion* (Van Dongen, De Medeiros, Verbeek, Weijters, & Van der Aalst). It also proposes great flexibility concerning different input and output files and formats (www.processmining.org). This makes it really usable and precious for companies that realize the need of improvement of their business processes and want to dive into the area. Experts can freely use the ProM to analyze process data and management can be supported in the decision of reconstruction and development of the business activities.

In addition to the explanation of the ProM as an available tool that support process mining, the Nitro commercial tool functionality is going to be presented shortly.

1.3.5 What is business process performance analysis and why is it necessary? What is the connection between this analysis and process mining?

After the explanation of different process mining types and the opportunities they give for process analysis, the essence of performance analysis will be explained in Chapter 3. Its relation with process mining techniques will be shown to reveal possible support in analyzing.
Performance analysis is normally referred to some improvements in an organization or solving of a problem (Rosset & Sheldon, 2001).

Process performance analysis (PPA) is a methodology that goes beyond the traditional process control and capability study (www.mvpprograms.com).

Process performance analysis use some specific measures called Key Performance Indicators (KPIs) to evaluate the performance of a process, detect variations, rate its potential and determine opportunities for improvement (Van der Aalst, 2011). The analysis may be conducted where control was not achieved and where a large stream of business processes exists (www.mvpprograms.com).

The key performance indicators are based on three performance dimensions: time, cost and quality (Van der Aalst, 2011). Chapter 3 will explain KPIs in details to reveal the opportunities for process analysis.

The methodology of performance analysis needs to be exercised on base of high quality business models. When they are not very close to reality, the analysis does not make much sense. Unfortunately models and real processes have very often little in common. Process mining helps in solving this problem by creating a direct connection between models already constructed and the actual process that takes place in a company. Different mining techniques like conformance checking, process discovery, organizational mining, social network analysis and decision mining support the execution of performance analysis of business activities so that redesign and development can be done (Van der Aalst, 2011). All these together with every process mining technique can be executed with the help of the specially created process mining framework (Van Dongen, De Medeiros, Verbeek, Weijters, & Van der Aalst), (www.processmining.org).

1.3.6 Which process mining techniques give the opportunity for process to be deeply analyzed and conclusions to be made concerning their performance?

In Chapter 3 and Chapter 4, some process mining techniques which can help in performance analysis of business processes will be presented.

The possibilities of process analysis through process mining to be shown, attention will be paid to the third type of process mining – enhancement and particularly extension. Various process mining techniques from the organizational perspective will be presented such as social network analysis, discovering of organizational structures and analyzing resource behavior. Time perspective will be added for a full insight of processes. Consequently, conclusions can be easily made concerning the workflows in a company, the frequencies and paths of task executed, people who are involved in the process run, the activities where bottlenecks appear, etc. This is the performance analysis based on process models and one will be presented with the help of a Petri net later on. The motivation to use a Petri net is that it represents a process clearly and has simple semantics (Van der Aalst, 2011).

Dotted charts can also help in business processes performance analysis. They represent the spread of events occurred in reality over time and can help to explain some causes of problems in processes performance (www.processmining.org).

A methodology to detect a failure of a process will be presented as well through the Performance Sequence Diagram Analysis which is a process mining technique. Sequence Diagrams can find the causes for certain behavior and prevent undesired performance of the process to occur (Van der Aalst, 2011), (Calderon-Ruiz & Sepulveda)**,** (www.processmining.org).

Chapter 4 includes the process diagnostics methodology which will serve as a base of the case study in this master thesis, presented in Chapter 5.

1.4 Research methodology and conclusion

Some basic ideas were introduced to show the importance of process mining for business performance analysis. In the next chapters all these concepts will be examined in depth, examples will be given and some case studies from real life used to support the theory will be presented. The research methodology for this master thesis will include literature review in the area of process mining and performance analysis of business process. Some case study will be used as well in order to show the practical side of process mining.

Process mining is a very powerful tool for business organizations. As shortly explained above it gives a lot of opportunities for a process to be analyzed according to its event data from a certain information system. From using the time perspective through the case perspective to the organizational one, process mining gives the opportunity for the processes to be analyzed in great depth, conclusions to be made about certain deviations, unnecessary tasks to be eliminated, etc. With the help of the ProM framework an organization has the great chance to use all the techniques available and combine plug-ins to “mine” its own processes in order to detect its problem areas and improve them in a sufficient way.

Chapter 2 - Process mining

Process mining is a relatively young research area. Its main idea is extracting substantial knowledge from events that have taken place in an enterprise and have been recorded by information systems. That is why the availability of an event log is an obligatory condition for the technique to take place. The aim of this chapter is to explain process mining. First, the essence of an event log as a main resource for the mining technique to take place, is explained. Second, the concept of process mining will be presented as well as the leverages it lays on. Then the three types of mining are going to be examined giving a clear vision of what the technique is capable of regarding discovering, monitoring and improvement of business processes across business units. Finally, as any other academic science, process mining faces some challenges but also provides valuable insights and different opportunities for the organization of nowadays workflow and the development of business activities.

2.1 What is an event log?

An event log is a set of data about the executions of a process within a company. This event log is the obligatory input for a process mining analysis. . Recently, it has become possible to gather a huge amount of data in a large number of domains across an organization. A data source may be a simple flat file, an Excel spreadsheet, a transaction log or a database table. Observation and learning from event logs attempts to generate useful knowledge and a new promising way of acquiring insights into business processes is the analysis of these logs, recorded by information systems. This type of study is a part of process mining research that has been developing more and more through the years (www.dataminingapps.com).

An event log provides detailed information concerning the activities that have been executed through a process that have taken place in an organization. In general, it contains sequentially recorded data that refers to one process and particularly all of the registered events related to it. Each event refers to an activity (a well defined step in the process) and is also related to a particular case (a process instance) to which the events are related, e.g. customer orders, loan applications, ticket booking, job applications, insurance claims, etc. Each time a process is executed, it will refer to one specific case. Every process instance has a start (case creation) and an end (case completion) as well as all the activities executed in between (Van der Aalst, 2011).

In general, an event log also contains some additional information for each event. These attributes can be a *timestamp* (date, start time of an activity and/or end time of an activity, execution time), a *resource* (person who has executed an activity) or any other attribute like *cost*, *product* or *customer ID.*, The included attributes depend on the process, the aim of the process mining analysis, the kind of business activity and type of organization. These attributes can be very useful when analyzing performance related to properties of the process and performance analysis of business processes that is going to be further discussed in Chapter 3. It is not necessary that all events which have been recorded in data systems possess exactly the same set of attributes (timestamp, resource, cost etc.) but typically those referring to same activities have the same attributes (Van der Aalst, 2011).

In summary, an event log contains information about a specific business process executed in an organization. A business process *execution* is related to a *case* that consists of *ordered events* that are represented in the form of a *trace* (a sequence of unique events). Events have different attributes like activity, time, resource, etc. that are included in the event data.

Table 2.1 illustrates a typical example of information presented in an event log. Each line in the table represents one event that has taken place during the process. Events are grouped per case but they are sequentially ordered according to the timestamp, i.e. the date and time when the events have taken place. By looking closer to case 1, it can be seen that it has 4 associated events in its trace. The first activity is ‘activity A’ that has been executed by John on March 9th, 2004. It is then followed by activities ‘B’, ‘C’ and ‘D’, executed respectively by Mike, John and Pete. Often a reference like a unique case ID for an event can be included so that events referring to the same activity can be distinguished. Concerning the timestamp, there are different ways of recording – here only the date and start time of an event is captured. Sometimes, more specific information about the timing is given, i.e. the start and end time of one event (in this way the throughput time can be calculated or sometimes it is already computed) and occasionally even when a task was offered to the resource.

Table 2.1 A fragment of an event log

Case ID	Activity ID	Originator	Time stamp
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	11-3-2004 11:18
Case 5	Activity E	Clare	11-3-2004 12:22
Case 5	Activity D	Clare	11-3-2004 14:34
Case 4	Activity D	Pete	11-3-2004 15:56

In Table 2.1, each event is associated to a resource but this data can be absent in some logs. In other event logs, it is possible that more detailed information is provided as the role of a resource in the organization. This is an example of a data attribute and in general many others may exist depending on the log recorded. Depending on the type of process mining analysis, only a particular part of the information in the event log will be used. The minimum information necessary for process mining is that every event needs to be related to both an activity and a case and that the events in the case are ordered. In Table 2.1, the columns

“Case ID” and “Activity ID” represent this minimum. The so-called *simple event log* includes only this type of information and does not include any attributes. A *simple trace* is a sequence of activities and a simple event log is then a multi-set of traces. Some examples of possible traces according to the example are:

- Case 1 – trace: activity A, activity B, activity C, activity D; or (A, B, C, D);
- Case 2 – trace: activity: A, activity C, activity B, activity D; (A, C, B, D);
- Case 5 – trace: activity A, activity E, activity D; (A, E, D);

If necessary *filtering* can be executed which means that different events can be removed from the log if not needed for a particular purpose of mining. It mainly helps removing unuseful information or incomplete cases and gives the opportunity of focusing on specific parts of the processes. An event can be filtered by type, category or a source for instance. Filtering can be a repetitive process – a log can be filtered more than once until it is simplified to the desired level. An example for filtering is to focus on the 10 most occurring activities and remove infrequent ones. It happens on the base of characteristics like absolute or relative frequency. An option can be for instance showing the tasks that have occurred in at least 5 % of all the cases; or presenting the top 80% of all activities recorded. Filtering is also used to simplify event logs and the consequent process models that are going to represent them as well as remove useless data and present some specific data, necessary for some particular process discovery.

According to the attributes shown in a log, different types of analysis can be executed. For instance, to analyze the different resources of a particular process, a resource analysis or social networks analysis can be performed. This will be further explained in the context of performance analysis of business processes.

2.1.1 Event log formats

In 2003, a standard for storing event logs was introduced, the *MXML* format (Mining eXtensible Markup Language). However, it started to become quite out-of-date and through the years some restrictions have appeared about what kind of data can or cannot be stored in an event log. As a result of these restrictions, a new event log format was developed later on – the *XES* (eXtensible Event Stream). Its main benefits are simplicity, flexibility (capturing event logs provided by any domain), extensibility (possibility to extend the standard for specific application domains or tool implementations for instance) and expressivity (encountering as little loss of information as possible). The XES standard introduces the concept of *extensions*. Event logs refer to the so-called extensions according to the particular attributes included in them. *Time* extension, for instance, defines a timestamp attribute where both date and time are recorded. *Concept* extension defines the name attribute for traces and events. *Life-cycle* extension defines the transition attribute for an event, e.g. "start", "complete", "skip", "suspend", "resume", etc. *Organizational* extension specifies three attributes for events: resource, role and group. The resource one refers to the resource or person that executed the event. The role (i.e. "position") and the group (i.e. "department") attribute characterize the resource's abilities and position in the organization. Different extensions can be added by organizations or users. They can be related to costs, risk, customers, etc (Van der Aalst, 2011).

XES also defines the concept of event *classifiers*. A classifier introduces a set of attributes which define the *identity* of an event. In other words, if in a log two or more events with the same value of all their attributes appear, they are considered to be equal (www.fluxicon.com).

A few tools exist which allow to work with event logs and the conversion of data sets. XESame is a tool which provides a generic way to convert data from a data source into a XES event log format. XESame can be adjusted to use a database from different vendors if required (Buijs, 2010). It is included in the ProM 6.1[1] framework, the last version of the open-source tool ProM. As already mentioned, the data extracted from the information systems is not ordered and more precisely not immediately available in an event log format. This makes them being unavailable to serve as an input in the ProM framework. The main advantage and a key strength of XESame is that it allows the conversion of a data source into an event log without programming. The entire conversion can be defined through the graphical user interface. The input data can be database tables, text files and others. The different event attributes are extracted from different fields of the records, e.g. the XESame "name" attribute

[1] www.processmining.org

is extracted from the “event name” field in the original data source, the “resource” attribute is extracted from the “username” field, etc. This way, structured XES files become available as an output and they are ready to be used for process mining in the ProM framework (www.processmining.org).

Nitro is a commercial tool that also allows conversion of files so that they can be loaded in the ProM. It transforms a .csv file in an event log with a .mxml or .xes format. Nitro provides an easy and user-friendly interface which makes the creation of an event log more accessible for a large group of people. The tool also allows already a general analysis of the event logs. The use of Nitro in the context of performance analysis is discussed further in chapter 3 (www.fluxicon.com).

In practice, some challenges appear when event logs are extracted. First of all, events have to be grouped by case. Therefore the unstructured and scattered data extracted from information system and contained in the event log has to be put in order according to the different paths that appear in the process, i.e. the various sequences of the activities. Further, in some situations, logs present only snapshots of in fact longer running processes so not all of the activities that have been executed appear and the processes come out incomplete. A precise choice of data in all the different information tables and their details is also essential. As a whole the provision of high – quality event logs is very essential for process mining and reliable and trustworthy recording of events is highly important (Van der Aalst, 2011).

2.2 What is process mining and what are the goals of this technique?

The goal of process mining is to analyze the event data provided by different information systems in a meaningful way, extract process related information by automatically discovering a process model out of it and finally monitor and improve the real processes that run in an organization. Process mining complements approaches like workflow management and business process management (Van der Aalst, 2011). Workflow management systems (WFM) design a process and search for discrepancies with the real workflow enactment (Weiters & Van der Aalst, 2001). However, they were introduced long time ago and have mostly been focused on the automation of processes. Business process management (BPM) extends the efforts of WFM, evolving from process automation and analysis to process management and organization of work. It supports creation of business process models and analyzes them for instance by using simulation. Process modeling is the activity of representing business processes held in an enterprise so that they can be subject of examination and development. The process models aim to be as close as possible to reality as they are used to orientate people to work in a particular way (Figure 2.1) (Havey, 2005). The truth is that most of the process models are disconnected from reality and only provide idealized versions (Van der Aalst, 2011).

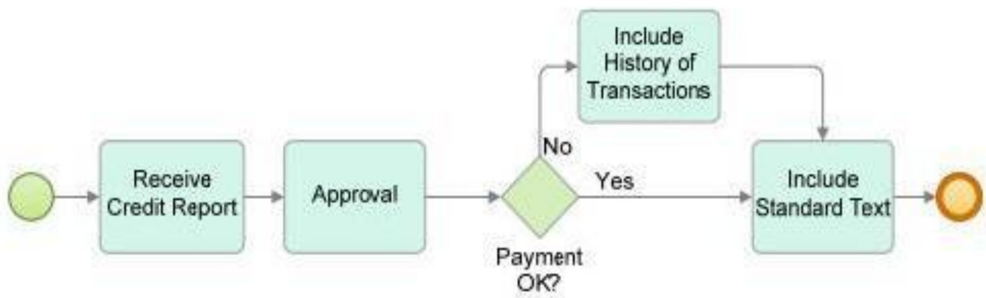


Figure 2.1 A simple business process model

The concept of process mining reverses the approach from “modeling and analyzing” to “extracting and modeling” by gathering information from workflow processes as they take place by using event logs recorded, and distills a structured process description from a set of real executions (Weiters & Van der Aalst, 2001). It is hard to extract and exploit event data in a meaningful way so that insights can be provided, problems forecasted or bottlenecks in processes identified. These are exactly the challenges that process mining aims to deal with. It is directed to the common problem of the lack of knowledge about what is actually happening in an organization. In reality, a significant gap exists between what has been predetermined to happen within a process (according to a predefined process model) and what happens in fact. Process mining tries to close this gap by analyzing business processes executions in reality with the information provided by event logs (Van der Aalst, 2011).

Figure 2.2 illustrates the idea of process mining. It shows the relations between information systems, event logs extracted from them, process models and real-life running processes. Taking an event log as a starting point and resource of information, three types of process mining can be carried out: *discovery*, *conformance* and *extension (enhancement)*, which will be discussed in section 2.3

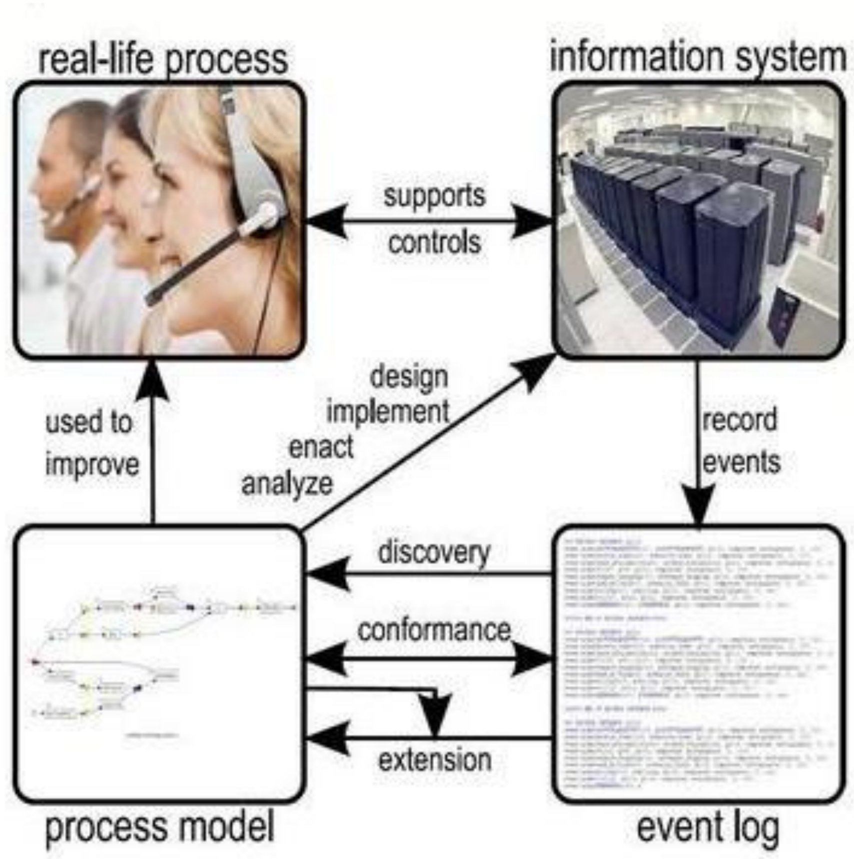


Figure 2.2 The concept of process mining (www.dataminingapps.com)

2.3 Perspectives of process mining

As already explained, event logs are an obligatory condition for process mining to take place and serve as its starting point. Three different perspectives can be distinguished: the *process perspective*, the *organizational perspective* and the *case perspective* (Van Dongen, De Medeiros, Verbeek, Weijters, & Van der Aalst).

The **process** perspective focuses on the ordering of the activities in a chain, i.e. the control flow. When mining this perspective, an expert can characterize all the possible paths for this process. This can be done for instance with the projection of the process on a Petri net (Van Dongen, De Medeiros, Verbeek, Weijters, & Van der Aalst).

The people involved in different activities and the relations between them are presented by the **organizational** perspective, i.e. the focus is on the originators of the actions executed. A social network can be created, showing the different roles and departments in a company and the connections and relationships between individuals within a process execution (Van Dongen, De Medeiros, Verbeek, Weijters, & Van der Aalst).

The **case** perspective focuses on the characteristics of a case. A case can be characterized by a path it follows or originators as well as the corresponding *values* of these elements, e.g. the throughput time (Van Dongen, De Medeiros, Verbeek, Weijters, & Van der Aalst).

In order to give an idea about the different possible perspectives, Table 2.2 below presents an event log recorded in a company and Figure 2.3 shows then a broad view of the respective mining results in terms of the three different perspectives (Van Dongen, De Medeiros, Verbeek, Weijters, & Van der Aalst).

Table 2.2An event log record

case id	activity id	originator	case id	activity id	originator
case 1	activity A	John	case 5	activity A	Sue
case 2	activity A	John	case 4	activity C	Carol
case 3	activity A	Sue	case 1	activity D	Pete
case 3	activity B	Carol	case 3	activity C	Sue
case 1	activity B	Mike	case 3	activity D	Pete
case 1	activity C	John	case 4	activity B	Sue
case 2	activity C	Mike	case 5	activity E	Clare
case 4	activity A	Sue	case 5	activity D	Clare
case 2	activity B	John	case 4	activity D	Pete
case 2	activity D	Pete			

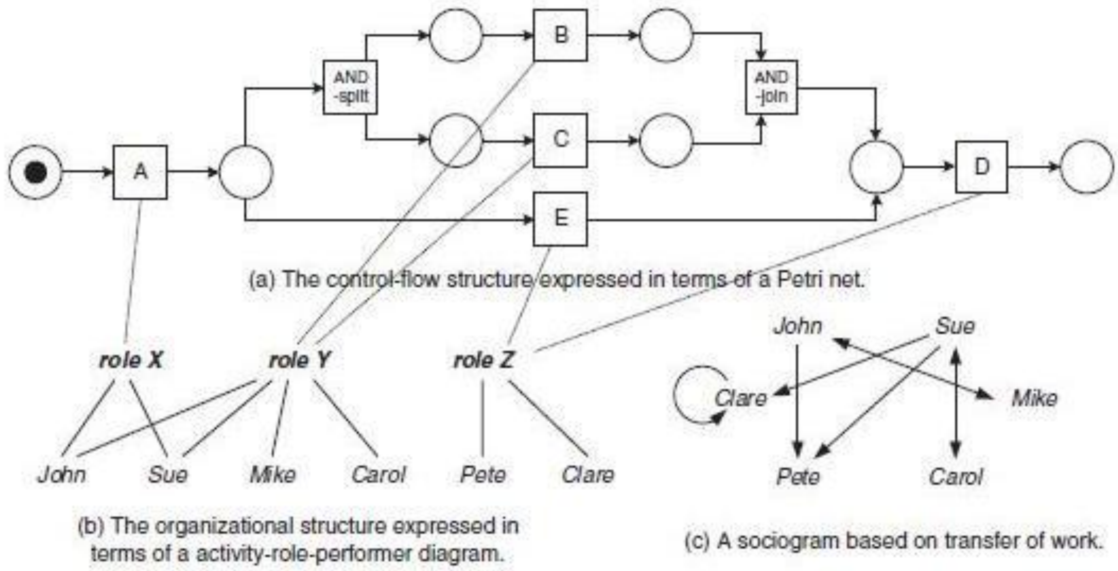


Figure 2.3 Process perspective (a), organizational perspective (b) and case perspective (c) based on the event log shown in Table 2.2

2.4 The three main types of process mining

Three different types of process mining can be performed by using event logs according to the purpose of the analysis.

2.4.1 Process discovery

The first type of process mining – *process discovery* – produces a model based on an event log without using any a-priori information. A lot of different process discovery algorithms exist. For example, an *α*-algorithm produces a so called Petri net that presents the real process behavior based on this event log. It maps a process model according to the path of activities captured by a log. With the help of the *α*-algorithm the behavior of the already explained simple log is sketched on the so called Petri net (Gunther, Rinderle-Ma, Reichert, Recker, & Van der Aalst, 2007), (Van der Aalst, 2011). This is a graphical tool for description and analysis of concurrent processes that arise in systems with many components (Petri & Reisig, 2008). A Petri net consists of *places (P)*, *transitions (T)*, and *arcs(☐)* (see Figure 2.4). It has exactly one start and one end place. Every other place or transition is situated on a directed path from the start to the end. Tasks are often represented by more than one transition with places in between. One transition represents one event (Van der Aalst & Van Hee, 2002).

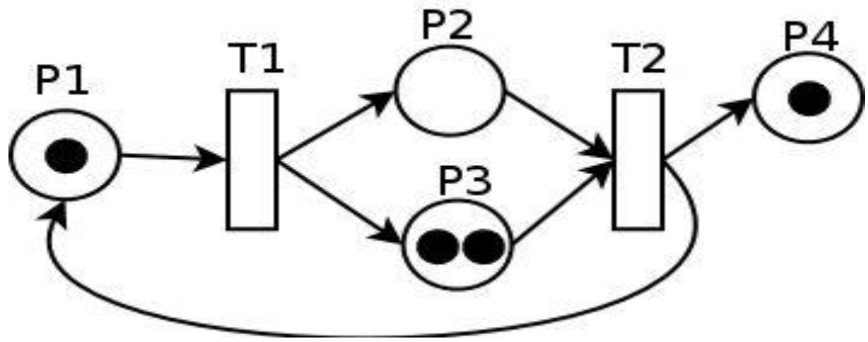


Figure 2.4 Example of a Petri net

What a Petri net actually does is to represent an event log as a process model. If it contains additional information like a resource for example, resource-related models can be discovered, too. Thus a social network can be examined, showing how people and groups work across an organization (Van der Aalst, 2011).

2.4.2 Conformance checking

The second type of process mining – conformance checking – examines if the reality as recorded in an event log conforms to the model or vice versa. It checks and analyzes deviations. By using conformance checking, an event log can be replayed on a process model to discover these deviations. Further, some business rules can be checked in the event log by using conformance checking. It can then be checked whether they are followed in reality or not. This type of process mining also keeps an eye on the execution of the so called "four-eyes principle" and detects deviations. When scanning the log, potential cases of fraud can be detected (Gunther, Rinderle-Ma, Reichert, Recker, & Van der Aalst, 2007), (Van der Aalst, 2011).

2.4.3 Enhancement

Enhancement, the third type of process mining, helps a process model to be extended or improved with a new aspect or perspective, on the base of actual process activities recorded in an event log. In this case an extension of an a-priori model is aimed. One type of enhancement is *repair* – modification of the already constructed model so that it can better reflect the reality. *Extension* is another type of enhancement. It adds some new prospect to a model by cross-correlating it with an event log. For instance, some performance data as time can be used and bottlenecks, throughput times and frequencies discovered (Van der Aalst, 2011).

2.4.4 Relations between event logs and process models

A key element of process mining is the realization of a strong relation between a process model and reality represented in the form of event logs. In the book of process mining three terms are introduced to reflect this relation: *play-in*, *play out* and *replay*. Figure 2.5 illustrates their concept (Van der Aalst, 2011).

Play-in represents the idea of the discovery technique explained above. An event log example is used as an input to create a process model. The *α*-algorithm is an example of play-in technique (Van der Aalst, 2011).

The **play-out** concept refers to the traditional use of process models. Behavior can be generated out of a Petri net, i.e. the initial requirement is a process model and out of it behavior can be originated. Play-out can be used both for anaysis and enactment of business processes. It supports conformance by analyzing and enacting the business activities. It is also used in simulations when experiments are conducted. Approaches like model checking are assumed to be play-out methods as well (Van der Aalst, 2011).

The **replay** technique is the one that relates to enhancement.

An event log and a process model are both used as inputs in it. The event log is therefore “replayed” over the preliminarily constructed model. Different reasons for “replaying” an event log exist:

- **Conformance check** – detection of discrepancies between the model and the log;
- **Extension of the model** – it can be extended with frequencies for instance, i.e. detection of which part of the model are mostly visited; discovery of bottlenecks;

- **Model construction** – different predictions can be made for different states of the model, based on the event log observed;

- **Operational support** – a case that is still in progress can be replayed in order for run-time deviations to be discovered if necessary; in this way forward steps to shape a case can be recommended (Van der Aalst, 2011);

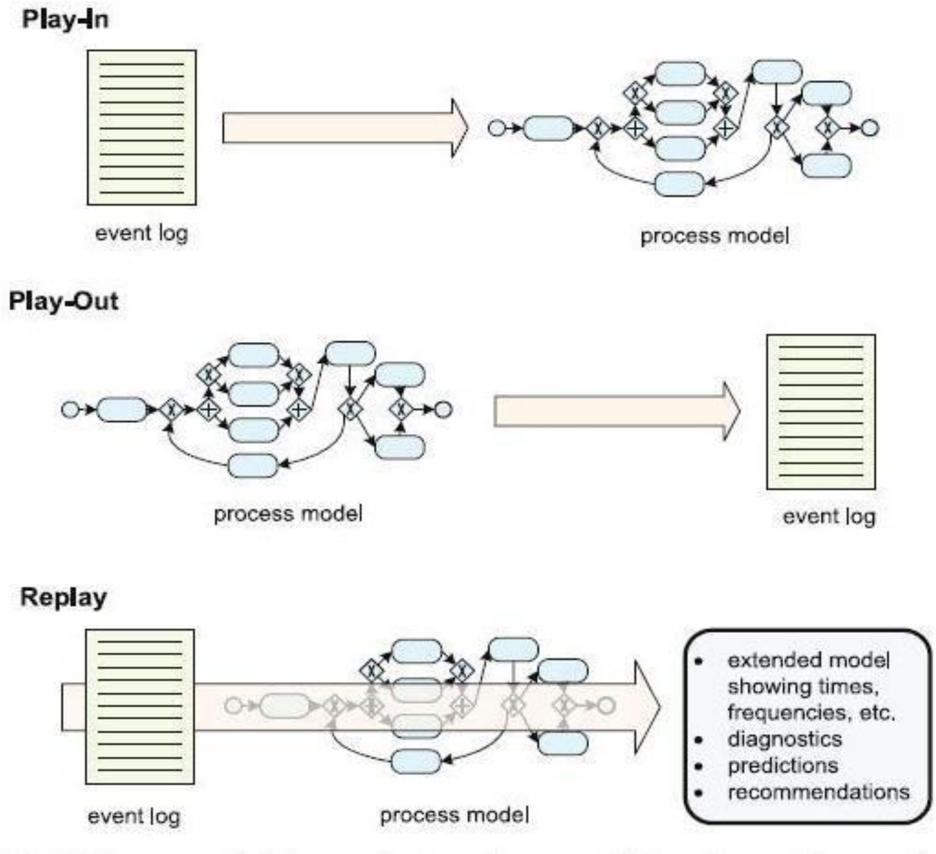


Figure 2.5 Ways to relate event logs to process models

2.5 Software

Different tools exist to read or store log files and present them in various ways. Therefore a difficulty appears in comparing mining results for one and the same data sets. The tools appear to be incompatible and non-combinative. This enforced process mining researchers to think of an appropriate infrastructure that could solve these issues – to be compatible and allow for the interaction between different mining tools (Van Dongen, De Medeiros, Verbeek, Weijters, & Van der Aalst).

As a result the ProM framework was created as well as some commercial tools that support the process mining technique and the extraction of files from different data sources such as XESame, ProMimport and Nitro (Van der Aalst, 2011). The ProM framework and Nitro are going to be explained in details as they are used in the case study analysis further in this master thesis.

2.5.1 ProM

The generic open source framework ProM was created at the University of Eindhoven to give an opportunity for the implementation of different process mining tools, including various algorithms and techniques. ProM is a “pluggable” environment, i.e. it allows addition of different functionalities by means of the so called *plug-ins*. A plug-in is a piece of software that can be added or removed from the framework without affecting its functionality. This shows the flexibility of the framework and its availability to receive and generate input and output formats. Different algorithms can be plugged in related to each of the process mining perspectives defined earlier. Figure 2.6 shows an overview of the ProM framework and the different groups of plug-ins that it contains (Van Dongen, De Medeiros, Verbeek, Weijters, & Van der Aalst), (Van der Aalst, 2011).

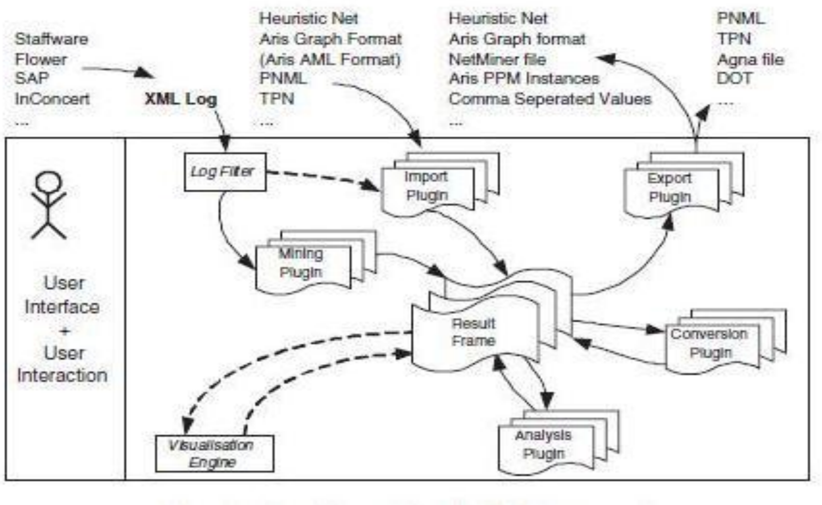


Figure 2.6 Overview of the ProM framework

As already mentioned, the main base for all process mining techniques is the event log which appears in different formats depending on the information system that had generated it. ProM can load both .mxml and .xes files which we already introduced in chapter 2. After the input file is imported, the different plug-ins available in the framework can be used, aiming to dig into the processes and analyze them in the different perspectives (Van Dongen, De Medeiros, Verbeek, Weijters, & Van der Aalst).

The first version of ProM (ProM 1.1) was released in 2004 and included 29 plug-ins: 6 mining plug-ins (generic miner, *α*-miners, multi-phase miner, social network miner, case data extraction miner), 3 conversion plug-ins, 4 import plug-ins, 9 export plug-ins and 7 analysis plug-ins (e.g. LTL checker, basic performance analysis and dotted-chart analysis). Through the years more and more plug-ins have been added to the software and the ProM 5.2 version released in 2009 reached 286 plug-ins in total: 22 import plug-ins, 45 export plug-ins, 44 conversion plug-ins, 32 filter plug-ins, 47 mining plug-ins and 96 analysis plug-ins. The improvement of ProM over time by adding more and more plug-ins available shows the realization of the main goal of the framework, namely providing a flexible and effective platform for development of different process mining techniques to help the analysis and further improvement of processes (Van der Aalst, 2011).

2.5.1.1 Types of plug-ins

In figure 2.6 the five different kinds of plug-ins available in the ProM framework are introduced. The *mining* plug-ins implement some mining algorithms which for example can construct a model in a Petri net notation from an event log recorded. The *import* plug-ins implement some functionality so that objects with a certain data format can be imported (“opened”). The *export* plug-ins make it possible to export (“save”) objects in different formats such as graphs, spreadsheets, Petri nets, etc. With the *analysis* plug-ins different kind of property analysis of mining results can be done. For instance, when a Petri net is constructed, there is a plug-in available that shows place invariants, transition invariants and a coverable graph. There are also some other plug-ins which give the opportunity to compare a log with a designed model . Finally, *conversion* plug-ins implement conversions between various data formats (Van Dongen, De Medeiros, Verbeek, Weijters, & Van der Aalst), (Van der Aalst, 2011), (Hornix, 2007).

2.5.1.2 ProM 6.1

ProM 6.1 is the latest version of the process mining framework. It was released in September 2011 and offers some new plug-ins as well as some new features compared to the previous variant. It is based on the .xes format rather than the .mxml. A new interface is introduced which can operate with many plug-ins, models and logs simultaneously. One of the new features introduced is the concept of packages which contain related sets of plug-ins and respectively the package manager which allows packages to be added, removed or updated. This helps against operating with irrelevant functionality as users can load only packages suitable for the particular tasks they want to execute. ProM 6.1 has the advantage that it can be customized for specific applications – domain and even organization ones. It contains a large set of analysis techniques which can be applied to processes that take place in reality. Different plug-ins support conformance checking and performance analysis and aim to help the improvement of process performance (www.processmining.org), (Van der Aalst, 2011).

Not all the plug-ins available in ProM 5.2 were added to the new ProM 6.1 software. Table 6.2 in Appendix 1 shows the main plug-ins included in ProM 6.1 (Van der Aalst, 2011).

2.5.2 Nitro

Nitro is a commercial tool released by Fluxicon[2] that aims to fill the gap between real-life logs and the ProM framework. As it is not possible to import a .csv or .xls information system file in ProM, Nitro is the one that helps and convert them into readily available for the framework .mxml or .xes event log format files (Fluxicon academic initiative for process mining).

The creation of an event log is highly supported by this tool but is not the only feature of the software. Nitro also allows for a general analysis of the event log. Firstly, it includes statistic views and instruments to drill down into various dimensions. In the last version of the software three new techniques were included that present case characteristics: case utilization (how much time per case is spent in executing activities, relative to the total case duration), mean activity duration (average time spent in an activity per case), mean waiting time (average time spent per case between two activities). Secondly, Nitro gives an overview

[2] www.fluxicon.com

of variants (cases with one and the same path) and process instances in the log. Finally, the last version of the functionality introduces the so-called *filters*which allow *to* clean up an event log, e.g. incomplete cases, error or inconsistent events, etc. Filtering is also very important for the drilling down into specific aspects of the processes and indepth and various analyses so that the insight into specific properties is dramatically increased. Six log filters are introduced: '*timeframe'*, '*variation'*, '*performance'*, '*endpoints'*, '*attribute'* and '*follower'*. They will be examined later as the Nitro tool is going to be included in the case study of the second part of this thesis (www.fluxicon.com).

2.6 Obstacles and challenges

To perform the three types of process mining, it is necessary that clear and structured event data should be used. This information is readily available in today's information systems but appears mostly in unstructured form as data is scattered over different kinds of tables or needs to be extracted from various subsystems in an organization. Part of the main efforts of process mining are directed to these problems as organizations today face the so called the *information paradox* – the more they automate the business processes, the less they are aware of their essence and behavior. Good understanding of activities is crucial for answering various business requirements and providing the enterprises the ability to redesign, control and optimize processes (www.dataminingapps.com).

In general, the mining of business processes has proven to be effective tool for analyzing the execution of operational processes and its techniques perform well on structured processes but still there are some problems when it comes to discovering less structured activity chains (Gunther & Van der Aalst, 2008).

Some additional obstacles and challenges exist in the context of process mining, proving it is still a young discipline. A major challenge is the balance between the so called *"overfitting"* and *"underfitting" of process models*. The overfitting represents a model that is too specific, it does not generalize process behavior and it is mostly driven by "accidental" information that appears in the data set. Underfitting relates to learned models that are too general, have poor precision and allow for very different behavior from what has been observed in the event log. It is mostly a problem when the data does not contain negative examples. So a conclusion can be made that overfitting is characterized by a lack of generalization and underfitting presents too much generalization. This appears to be a serious problem to be worked on and solved along the future research in the field of process mining (Van der Aalst, 2011).

Other common problem when mining processes is incorrect logging which can inevitably appear in information systems but is impossible to be distinguished by the discovery algorithm (e.g. wrong date or time recorded, future date recorded, etc.). Duplicate or missing tasks can appear as well. In simple processes it is possible to catch the absence of some “hidden” tasks and to detect them, but in more complex processes it is quite hard and approaches like observable behavior and bi-simulation try to support the solution of this problem (Van der Aalst & Weijters, Process mining: a research agenda). Another issue is the performance of incorrect tasks. An obstacle might be the limited information provided in the event log (it shows what had happened but not what could not happen or what should not have happened for instance). Sometimes a problem is that the log representation is just a fraction of all possible behaviors due to concurrency and loops as well. Further, a lack of knowledge about *“controlled decisions”* can exist (this is a situation where the execution of some tasks in the log depends on choice to execute or not some other task in another part of the log, i.e. the correlation between choices made cannot be detected; for example in the model shown on Figure 2.7: after executing task C there is a choice between task D and task E. However, the choice between D and E is ‘‘controlled’’ by the earlier choice between A and B; such constructs are difficult to mine since the choice is non-local and the mining algorithm has to ‘‘remember’’ earlier events) (Van der Aalst & Weijters, Process mining: a research agenda).

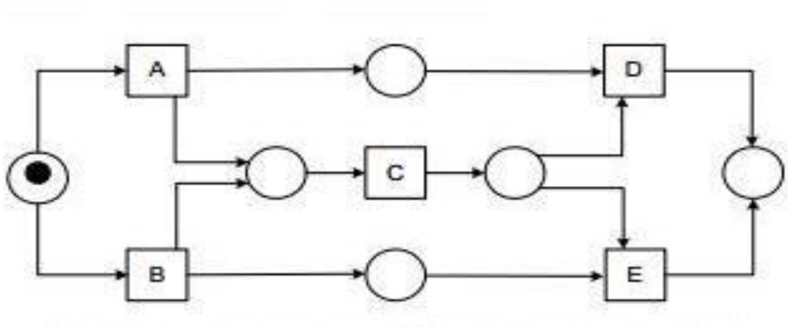


Figure 2.7 A process model with a non-free-choice construct

The discovery process mining technique meets some obstacles as follows:

- It is required that the behavior is representative (based on reality) but it is not quite clear what does this mean and what exactly the model is going to represent;
- Target format of the model is not specified; often it is defined to be Petri net model as it is simple, graphical and allows for the modeling of concurrency, choices and iteration if necessary;

Noise and *incompleteness* are other negative aspects that appear in the context of process mining. Noise in an event log refers to content of rare and infrequent behavior that is not representative for the typical process running (Van der Aalst, 2011). There may be some incorrectly logged data – an event that took place but was not recorded or an event that was recorded some time after it actually had taken place. A few process discovery approaches help in filtering the noise out as it is very essential for the proper conduction of different analyses and these are the following techniques: *genetic* mining, *heuristic* mining and *fuzzy* mining that are going to be considered later on (Van der Aalst & Weijters, Process mining: a research agenda), (Van der Aalst, 2011).

Incompleteness is related to the case when an event log contains too few events to discover control flow structures. Completeness in process mining is very essential and is related to noise. However, whereas noise means appearance of too much data, the incompleteness reflects the problem of having too little data (Van der Aalst & Weijters, Process mining: a research agenda), (Van der Aalst, 2011).

These problems can be solved by *cross-validation:* dividing the log into a *training* log and a *test* log. The training log learns a process model and the test log evaluates this model based on unseen cases. K-fold cross-validation can also help, which means splitting the log into a number (k) of equal parts and doing as many test as the number of the parts. Each test includes one part that is a test log and all the other parts serve as training logs. Still the lack of negative examples is a problem in these approaches because the data set provides only possible scenarios and not these that cannot happen (Van der Aalst & Weijters, Process mining: a research agenda).

Another important challenge of process mining is the visualization of the results as well. Considering the goal of process mining, it is clear that people in an organization have to be able to gain insight in the processes, understand them and understand the way process mining can be used to analyze them. The visualization of different aspects (e.g. flow time, work in progress, etc.) is not covered by commercial products related to process mining and needs further research and serious work in this direction (Van der Aalst & Weijters, Process mining: a research agenda).

2.7 Summary

In summary, process mining has been built on the base of data mining and business process management sciences. It can be assumed as a new set of business intelligence techniques. It is not restricted to analyzing some historic data, but also provides operational support, recommendations and expectations for real-time running processes.

Process mining gives the opportunity for simplifying and unifying the analysis of business processes running in an organization, evaluating their performance and developing them. The techniques it provides are directly applicable and supported by the *ProM* framework that is going to be a subject of the next chapter where the algorithms implemented are going to be pointed and their importance for the today’s enterprises is going to be explained.

Chapter 3 - Performance analysis

Nowadays, a lot of companies experience the problem of not understanding the business processes that flow through the whole organizational structure. The information technology has been offering a lot of opportunities for modeling, executing and recording different activities which makes it easier and faster for people to work but meanwhile a challenge appears in the so-called performance analysis of processes. There are some important questions like ‘How exactly is a process executed?’, ‘Are there any bottlenecks in the process?’, ‘Why do some processes fail?’, ‘Are there any gaps, any steps missing or missed to be completed?’, etc. All these questions can be answered by using a process performance analysis (PPA) (Hornix, 2007).

3.1 Process Performance Analysis

Process Performance Analysis (PPA) is closely related to disciplines like ‘Business Process Monitoring’, ‘Business Activity Monitoring (BAM)’, ‘Performance Measurement’ and ‘Corporate Performance Measurement (CPM)’. They are all connected to process behavior and aim to offer tools for monitoring operational business processes (Van der Aalst, 2011). Business process monitoring and BAM provide information about real-time running processes and support their analysis. It aims to find bottlenecks in the task chains and unreliable tasks. Moreover the two types of monitoring focus on time, cost and performance measurement of activities and there also exist BAM tools which support the visualization of the discovered information. Measuring of processes through these techniques helps identifying inefficiencies and incompleteness so that optimization can be put in place. Finally, the feedback provided by the tools and their impact on various situations is used in business solutions (Brandl & Guschakowski, 2007).

Performance measurement and CPM are a process of collecting and reporting data which refers to an individual, group, department or an organization as a whole. It includes the monitoring of processes and setting of strategies to achieve the outcomes which were planned and defined. The measurement of performance estimates the parameters under which programs and investments are reaching the results targeted (Gamble, Strickland, & Thompson, 2007).

The three perspectives of measuring performance according to Andy Neely in his book “Business Performance Measurement: Theory and Practice” are ‘accounting’, ‘marketing’ and ‘operations’ perspectives. According to him, measuring the performance can help managers to understand activities and specifically which of them generate revenues that exceed costs. The performance measures that he uses for this purpose are the following: financial, customer, business processes and innovation and learning measures. There exist measurement systems which provide tools for financial measurement and overall business performance measurement.They appear as means for motivation and control of the stakeholders in a company to reach the set goals and achieve high performance results (Neely, 2002). Companies realize they need to operate fast and effective, keep competitive and improve their performance by developing cross-functional activities. The role of PPA is to support enterprises in measuring and analyzing the performance of their business processes (Hornix, 2007). A few definitions exist which explain the meaning of the term “performance” such as “the accomplishment of a given task measured against preset known standards of accuracy, completeness, cost and speed” (www.businessdictionary.com) or “process or manner of functioning or operating” (www.thefreedictionary.com). The definition that is used in this work is the following – “the degree to which an organization carries its objectives into effect” – as it will best fit the purpose and the subject of this thesis, namely the performance analysis of business processes (Flapper, Fortuin, & Stoop, 1996).

3.1.1 Simulation

An important tool in the context of performance analysis is simulation. Simulation is used to imitate a real-life process or system over time. It is a commonly used method to understand the factors that influence performance characteristics. With the help of simulation a model is executed repeatedly. The goal is not only to state a single value, e.g. “the average response time is 07.07 min” but to discover a so-called confidence interval, e.g. “the average response time lays with 80% certainty between 07.00 and 07.30 min”. Hence, a few subruns of a process are normally executed instead of a single simulation run. In order that experiments can be conducted, the number and length of subruns have to be specified. This information, together with data about resources that appears in workflow models, helps simulation tools to provide details for performance properties such as:

- *response time* (the time a generic systems or functional unit takes to react to a given input),
- *waiting time* (a period of time which one must wait in order for a specific action to occur after it was requested),
- *service level* (measures the performance of a system; after certain goals are defined, the service level gives the percentage to which they should be achieved),
- *throughput time/flow time/lead time* of different activities or the process itself (the average time that a unit requires to flow through the activity/process from the entry point to the exit point),
- *frequency (the number of occurrences of a repeating event per unit time)*,
- *utilization rates of departments* (the rate at which potential output levels are being met or used; percentage of the capacity actually used during a month, quarter, year, etc.) (Van der Aalst, Nakatumba, Rozinat, & Russell), (www.netmba.com), (www.investopedia.com), (www.businessdictionary.com).

On the way of modeling and analyzing business processes by the use of simulation, management in organizations may get ideas on how to reduce costs when improving service or utilization levels or lowering flow times. After a detailed performance analysis of running processes is executed, different ways of improvement in cost, quality and flexibility can be established (Van der Aalst, 2011).

3.1.2 Performance indicators

The performance of an organization can be expressed by quantitative, measurable indicators commonly referred to as *key performance indicators* (KPI's). They can be either time-related (e.g. throughput times, service times, waiting times, time spent between two different activities), cost-related (e.g. process costs, material costs) or quality-related (e.g. frequencies, error rates). According to the mission and the main goal of an enterprise, different KPI's are perceived to be important. They are closely connected to the critical success factors of a company. KPI's measure the contribution of business processes to the main objectives of the organization (Hornix, 2007).

The process mining tools support PPA by observing the actual values of the above mentioned KPIs of business processes. In this way deviations can be detected that reveal the difference between planned and actual values of KPI's. This allows for the correction of errors and improvement of processes so that they can help achieving the main goals of an enterprise. Performance analysis turns out to be very useful in risk analysis, prediction of results and different scenarios and prediction of potential problems and making of plans for their solution (Van der Aalst, 2011).

The performance analysis is in a relation with all of the types of process mining but mostly with the enhancement type of process mining and is supported mostly by the replay technique explained in Chapter 2. Executing the replay method, different kinds of performance-related information can be derived, some of which main concepts already mentioned.

- *Service and waiting time visualization* – some statistical information as average waiting time or difference in service times can be projected and highlighted on the model.

- *Flow time (lead time, throughput time)* – the time it takes for a process execution between two specified times. Two main points of time in the process can be taken such that different statistics can be made like the average flow time between these two points or some other case properties (e.g. average lead time over all cases; service level such as percent of cases with lead time lower than some threshold value) that can help for monitoring of the so-called Service Level Agreements (SLA's). For instance, the order in which activities should be executed, taking into consideration time dimensions like complete times, waiting times for an activity to be executed after another activity can be defined. .Some aspects of flow time analysis will help here to measure the consistency of reality with the predefined goals and agreements set. Non-conformance with these SLA's can be also highlighted in a process model.

- *Identification of bottlenecks, analysis and measures for their elimination* – in some cases some activities take more time than expected which causes the appearance of bottlenecks that block up the whole chain of activities. These bottlenecks needs to detected and further investigated in order to improve the process. These places that take too much time can be highlighted on the model. In the situation of bottlenecks identification in a few cases, they can be separately analyzed from the other cases with the purpose of root causes for the delays to be investigated.

- *Frequencies analysis* – through replaying a model, all times are recorded and frequencies are collected. For instance, in the following situation, some frequency statistics can be extracted: within a process there is a choice either to execute activity B or activity C after activity A. With the help of the analysis, a conclusion can be made in how many cases activity A was followed by activity B or C.

- *Resource behavior* – when combining the above mentioned frequencies results with average service times, utilization of resources can be also detected, e.g. in a model resources and waiting times can be projected to give a broad picture of the process and its resource behavior. (Van der Aalst, 2011), (Van Dongen & Adriansyah, Process mining: Fuzzy clustering and performance visualization).

3.2 Performance analysis and process mining

Companies often have very limited information about what is actually happening in their everyday processes. This is a common problem in many organizations and usually a significant gap exists between what was predefined to happen and what happens in reality. Process mining can be used as a powerful tool for the "process owners" to understand the activity chains – monitoring their behavior, developing and improving them on the base of observations and analysis that can be performed by the techniques that process mining offers (Gunther, Rinderle-Ma, Reichert, Recker, & Van der Aalst, 2007).

When process mining takes place in the context of the performance analysis of business processes, there is a compulsory requirement to be met, i.e. each event has to refer to a timestamp, i.e. the time when it was recorded. In cases where the time between two actions for example is remarkably long, process mining can help for identifying a problem and reasons for this delay. When analyzing performance through process mining, different conclusions can be made about the participants in a process, decision points and roles, trends showing which activities tend to fail most or which resource tends to delay actions, etc. The techniques of process mining and the plug-ins included in the ProM framework give a precious opportunity to stakeholders in organizations to monitor and control the process flow, observing various relations and dependencies in a net of activities, resources and time dimensions (Van der Aalst, 2011).

Process mining delivers reliable information concerning the way process models have to be changed in order that processes can become more flexible and developed. With the help of this discipline and the ProM open source framework created to support it, as well as the commercial tools available, the reasons for process failures can also be detected and left out in the future. All these are possible after a thorough analysis with the help of the mining techniques and the plug-ins available to assist their realization (Calderon-Ruiz & Sepulveda).

Process mining can be adopted as useful and helpful technology to support the goal of disciplines like BPM, BAM and CPM mentioned earlier. As already explained, process mining aims to analyze the real process behavior. By using an event log, a process model can be created to see how the process is actually going. This is not only limited to the projection of performance data such as average flow time or average response time for example. The focus is also on the relationship between the different activities that take place during a process and how they affect each other (Van der Aalst & Weijters, Process mining: a research agenda).

As a result of the deviations in some cases indicated by information systems and the respective performance problems that can occur, organizations wonder frequently whether their systems are working in the appropriate way, i.e. the way the organization think they work (Van der Aalst, 2011).

With the help of process mining, the recorded event logs can be analyzed and further details for the process performance can be discovered. Various questions can be answered regarding the actual process that takes place, cases which appear, the presence of bottlenecks, social network relations and others that give an idea about the performance characteristics of a process. This helps companies to monitor the consistency of the information systems they use and the way they reflect the flow of activities (Van der Aalst, 2011), (Van der Aalst & Weijters, Process mining: a research agenda).

PPA expresses the study and evaluation of the current running processes to measure performance, find problems and solve them. Further, PPA also aims at optimizing processes and showing weak places where they should be developed and improved. With the help of process mining techniques and the framework to support them, performance analysis results can be projected on a process models where bottlenecks can be quickly discovered and resolved (Celino, Alves de Madeiros, Zeissler, Oppitz, Faaca, & Zoeller), (Van Dongen & Adriansyah, Process mining: Fuzzy clustering and performance visualization).

3.3. Performance analysis with the ProM framework. Process diagnostics

The ProM framework and particularly some of its plug-ins were created to support the analysis of business processes. In this part of the chapter, plug-ins that allow for the performance analysis of organizational processes will be examined and their contribution to the development of the processes will be explained.

The functionality presented in the ProM framework consists of qualitative analysis of processes, i.e. the logical correctness of processes (absence of anomalies like deadlocks or livelocks) and also affords an opportunity to help the performance analysis, i.e. a quantitative analysis of processes (Hornix, 2007).

In order to explain the opportunity of process overview and performance analysis of business tasks, an extend version of the process diagnostics methodology is used. This technique is also going to be the base of the case study in chapter 5. Process diagnostics is closely related to performance analysis. The goal of process diagnostics is to discover a process for a short period of time. The technique is supported by the tools available by process mining. There are five phases included in the diagnostics: *log preparation* (extraction of the event log from the information system), *log inspection* (process overview), *control flow analysis*, *performance analysis* and *role analysis*. The phase of performance analysis focuses on problems and bottlenecks that appear in a process, i.e. weak places which limited capacity reduces the capacity of the whole chain (Bozkaya, Gabriels, & Van der Werf).

The different phases of process diagnostics and the useful plug-ins of ProM and Nitro are explained in chapter 4.

3.4 Summary

Performance analysis of business processes is a very important step in the overall activity of an enterprise as it shows where the organization is supposed to direct its forces to and where it would be necessary for problems to be identified, analyzed and left out. A company should also focus on development and improvement of the everyday activities and find a way for the realization of this. A possible and effective option to exercise process observation, discovery, development and improvement is the process mining technique. Through the different methodologies it includes and the tools that support it (academic and commercial), it gives a great opportunity for the examination of processes or parts of them, activities that take place within them and people who execute them. Thus, a broad view of the overall organizational activity can be projected and positive effects for the company can be attain, turning its initial goals into real achievements.

Chapter 4 Process diagnostics

In this chapter, the different phases of the process diagnostics methodology will be explained into details. Further, the different plug-ins of ProM and Nitro which are important for each phase will be discussed. These plug-ins will be used in the case study of chapter 5.

4.1 Log preparation

It was already mentioned that information systems provide scattered and unstructured data that does not appear in a readily available format to serve as an input in the process mining framework or another tool available. Therefore, extraction of information and preprocessing of the log is necessary. XESame and Nitro were already introduced in chapter 2 and 3 to convert an information system data format to an .mxml or .xes event log which can be imported in the ProM framework. With the help of these two softwares, various activities and their events can be identified. They have to be ordered and grouped under different columns titled such as: *activity ID*, *resource*, *time stamp*, *other*, etc. If the log includes multiple timestamps for example, their semantics need to be clarified, e.g. is it a timestamp of the start of the event or of the case. Also the timestamp format has to be specified as it can appear in different forms (e.g. dd/mm/yy hh:mm; mm/dd/yy hh:mm; dd-mm-yy hh:mm; yy/mm/dd hh:mm, etc.) After all the steps are completed, a conversion of the file can be put in place (Bozkaya, Gabriels, & Van der Werf), (Van der Aalst, 2011).

4.2 Log inspection

The log inspection phase is used to get familiar with the event log and the process which it reflects. With the help of Nitro (see Figure 4.1) statistics about the log can be observed that present information about number of cases and events, different activities (events) that take place, number of task executed per case, start and end events as well as their occurrences, resources and other attributes included in the event log, etc. All this information appears also in the form of different graphs and tables so that various insights can be obtained. Also filtering can be done with Nitro in order to remove incomplete cases for instance. These are cases which start before the start of the log or are still running after the end of the log. The Fuzzy Miner plug-in of ProM can further be used to discover the real process model by using the information in the event log (Bozkaya, Gabriels, & Van der Werf).

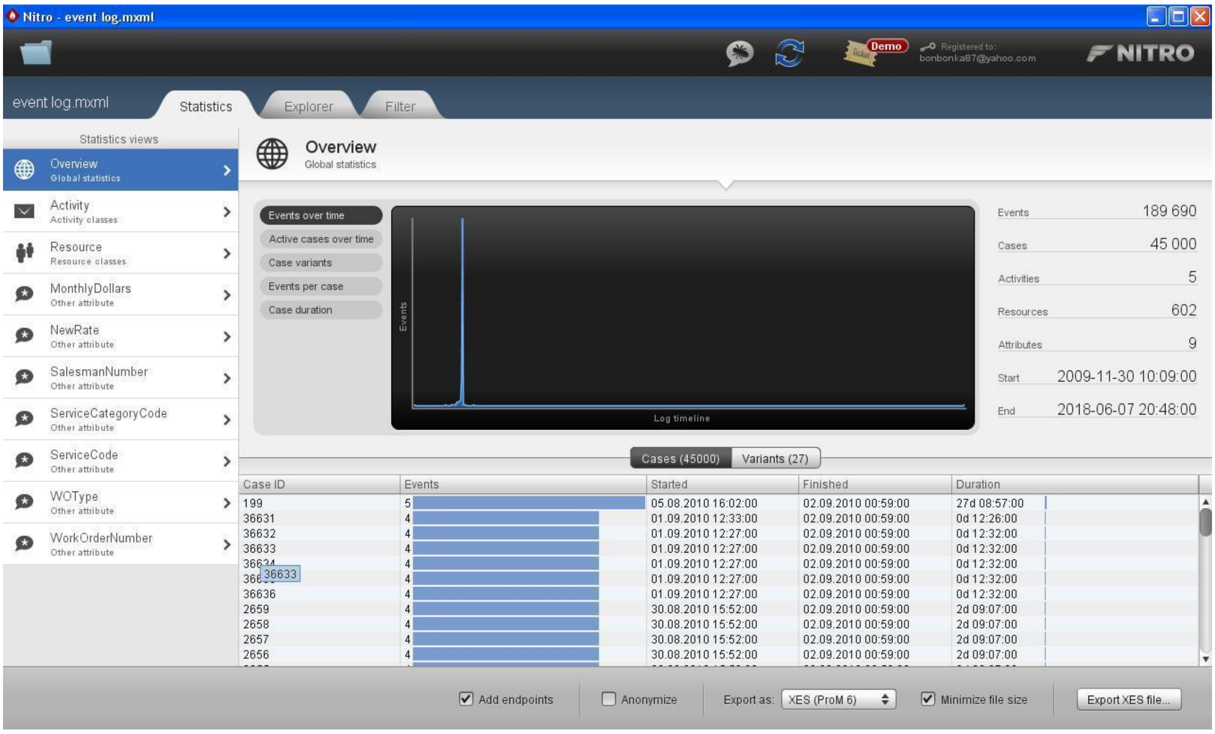


Figure 4.1 Nitro statistics overview of an event log

4.2.1 Fuzzy miner

The *fuzzy miner* serves well the performance analysis of business processes in an organization and especially those which are less structured and reveal unstructured, contradictory and inconsistent behavior (Bozkaya, Gabriels, & Van der Werf).

When loading an event log in the plug-in, first a measurement table appears that gives the option metrics that can be seen to be specified (*trace, significance, correlation*). Setting the *weight* value of the metrics determines how strongly it taken into consideration when mining. Thus if an emphasis on a specific metric is aimed, the weight of the others can be reduced. The weight of different significances and correlations can be set. Next, some attenuation properties can be set after which a graph (fuzzy model) appears that reflects the process (Figure. 4.2) () (Van der Aalst, 2011).

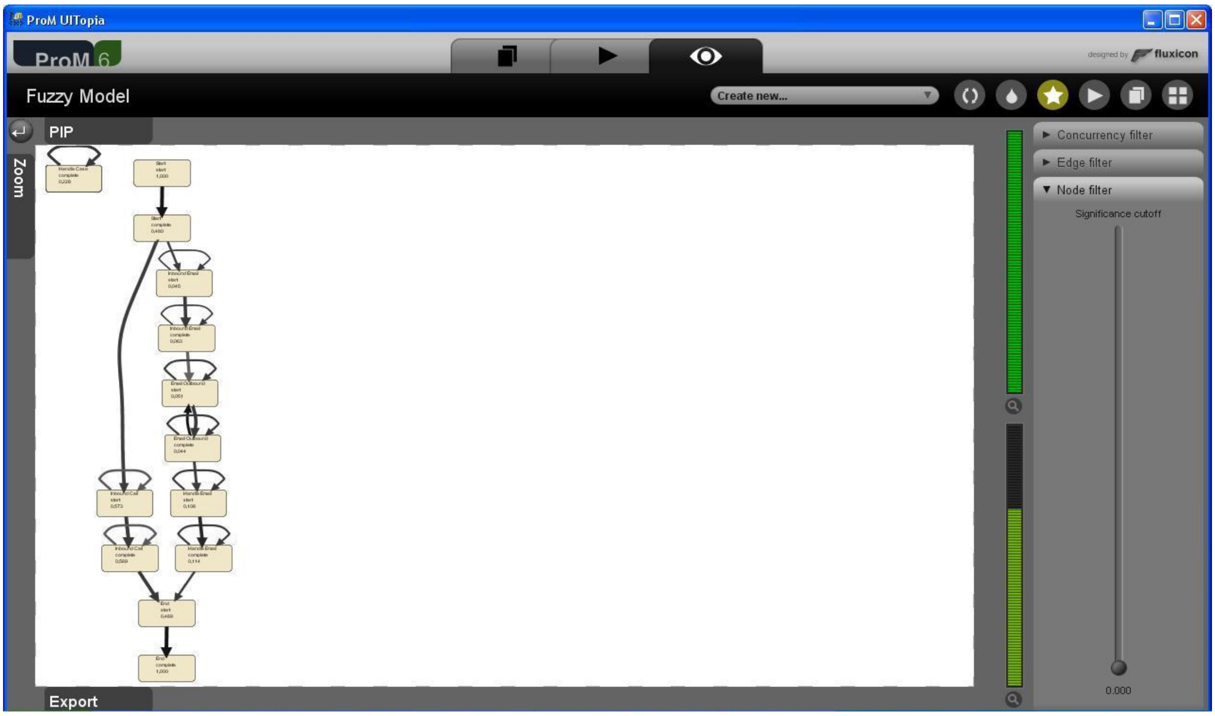


Figure 4.2 Fuzzy miner

It presents the sequence of activities in the process examined. Event classes appear in yellow boxes together with their significance value (written under the name of the task). Normally, less significant and low correlated behavior is excluded from the graph but if it appears, it is displayed in green octagons on the graph. Arcs on the graph represent the relations between activities and are colored in grey, the lighter the grey , the lower the significance of the relation. There are also three filters available in the fuzzy miner plug-in so that unnecessary elements can be left out. These are the node filter (controls the amount of event classes that are going to be included in the graph), edge filter and concurrency filter (resolves conflicts between two event classes, e.g. parallel events (Bozkaya, Gabriels, & Van der Werf), (www.processmining.org), (Van der Aalst, 2011).

The fuzzy miner can construct hierarchical models where activities that appear infrequently can be moved to subprocesses. The discovered process modelneeds to be easily understood by the user, providing information on the frequency and importance of tasks and paths and supporting the performance analysis of business processes by empowering the user to interactively explore activity chains with the help of event logs. *(PM book)*

4.3 Control flow analysis

In this phase of process diagnostics, the accent is on the ordering of activities within the process. The way the process actually looks is examined here. If a process description is predefined, a conformance check is executed to reveal whether the process that appears in reality conforms to the process model constructed. An event log is for 100% conform with a process model if each case of event log can be replayed in the process model. This phase also includes the exclusion of infrequent sequences, so that at least 80 % of the log is taken into consideration but still it has to be comprised of cases with only high frequency occurrence. The *Performance Sequence Analysis* helps for the identification of the most frequent paths in the log (section 4.3.1). An alternative is the use of Nitro (section 4.3.2) to filter out the most frequent behavior (Bozkaya, Gabriels, & Van der Werf), (www.fluxicon.com).

4.3.1 Performance sequence diagram

The performance sequence diagram (PSDA) is another tool available in the ProM framework that aims to observe process performance in an enterprise. It follows the behavior of the activity chains to discover common or rare behavior of processes as well as results that occur in extreme situations (for instance, cases with very high throughput times). The Performance sequence diagram analysis supports the investigation of possible reasons that cause such behavior. Identical sequences recorded in a log during a process execution are grouped together, i.e. if in a log the paths (A, B, C, D, E), (A, B, D, E), (A, C, D, E), (A, B, D, E) are recorded (where A, B, C, D and E are different steps to take place in a process), PSDA will group together the second and the fourth sequences in one pattern as they follow the same succession of steps executed. The execution patterns are drawn in the shape of blocks and arrows where blocks correspond to the duration of tasks and arrows to connection between activities (Figure 4.3). The plug-in gives an opportunity to focus on different data elements such as task ID, originator, etc. and to determine how transfer of work between instances takes place for each case. The sequences that reflect the transfer are compared to eachother and discovered if they follow the same pattern. These patterns are displayed in different diagrams – the pattern diagrams. The throughput time information available in the plug-in allows for the examination of patterns that appear often or rare and indicates the causes for undesired behavior which supports the eventual future prevention of such problem performance (www.processmining.org), (Calderon-Ruiz & Sepulveda), (Hornix, 2007).

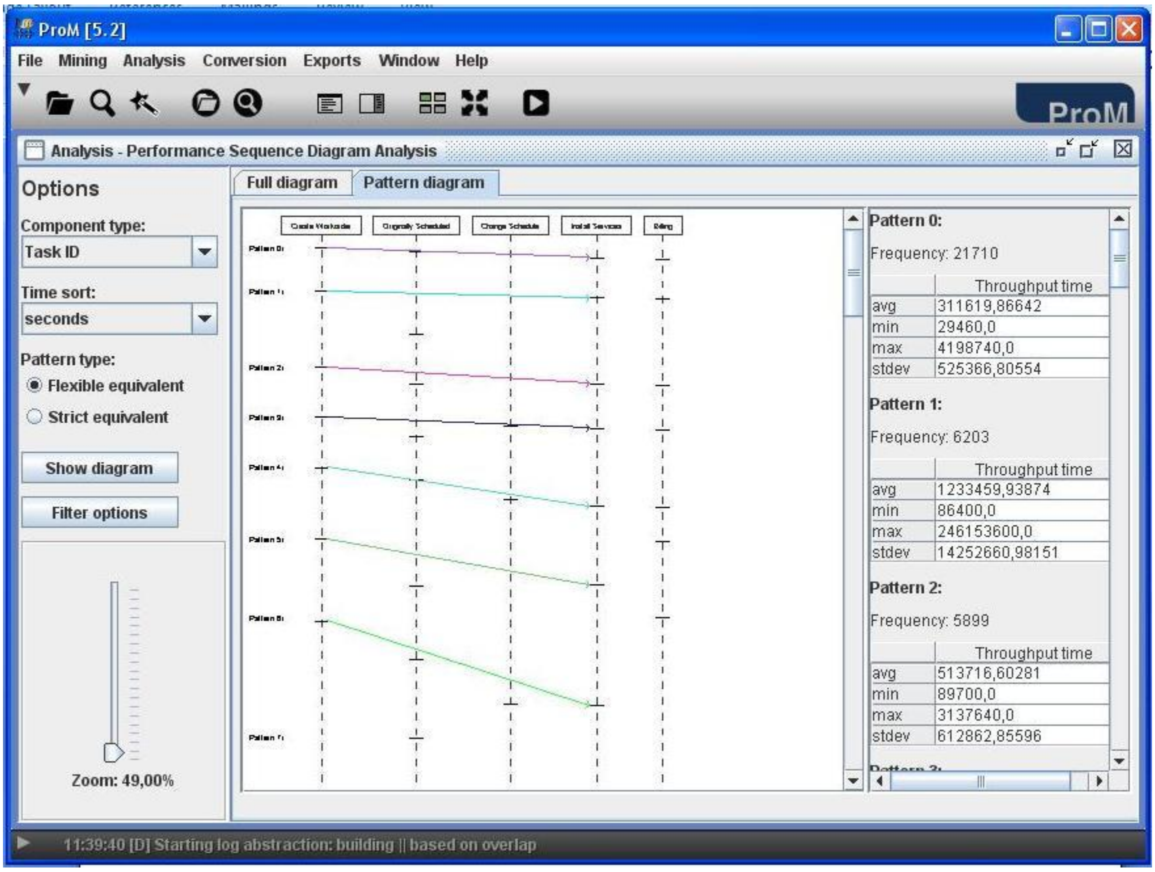


Figure 4.3 Performance sequence diagram

A few settings can be applied while using the performance sequence diagram. First a ‘data element’ (component type) can be selected according to the information recorded in the event log. This can be a task ID, originator, salesman number, service code, etc. Second, ‘time sort’ is to be set such as seconds, minutes, hours, days, etc. The last option available is the pattern type which can be either strict-equivalent or flexible-equivalent. By choosing the strict one, sequences are assumed to be different in case the blocks shown do not start and end in exactly the same order. When selecting the flexible equivalent, for each block a requirement is set such that all blocks that begin after this block ends, have to begin after it ends in the compared sequence as well (www.processmining.org).

The diagram contains detailed information about sequences, data elements and times which can be discovered by moving the mouse cursor over a sequence, a block or an arrow displayed on the full diagram. At the pattern diagram (the right tab on the top) the same information related to specific patterns can be obtained (www.processmining.org).

The filter option allows to limit the number of process instances in case there are too many and the user wants to analyze particular ones.

PSDA is sometimes used to discover why processes fail. Successful cases executed and recorded in an event log can be compared to unsuccessful ones and reasons for collapse to be investigated through an extensive research. Activities that are never or unnecessarily performed can be found as well as behavior that leads to wrong business process outcomes. The technique allows organizations to reduce time spent on tasks and associated costs which reflects the significance of this algorithm and its important support considering the business process improvement (www.processmining.org).

4.3.2 Nitro and the control flow

The *Explorer* option in the Nitro software can also be useful when examining the process and the ordering of activities in it. The different variants of the process, i.e. the different patterns appearing in the event log, and the number of cases of each variant helps the user to realize which of the sequences are common and frequent and which of them appear rarely in the process. Looking at the screenshot on Figure 4.4 for instance, we can see that the range of Variant 20 to Variant 27 sequences appear very rarely (0.00 – 0.05 % frequency of the total event log). The number of cases following a variant is given as a percentage and an absolute number (in this case there are 45 000 cases recorded) The cases on their place are presented together with the number and kind of events that take place within them.

Furhter, with the help of the *Filter* option that Nitro includes and specifically the *Variation filter*, the rarely appearing variants can be easily removed from the log so that further analysis can be focused on the usual behavior of the process. The opposite can be also performed – if the user wants to focus on infrequent cases in order to try to investigate the reason of their occurrence for example – he can easily filter out the frequent variants and leave the other ones in the log for further analysis.

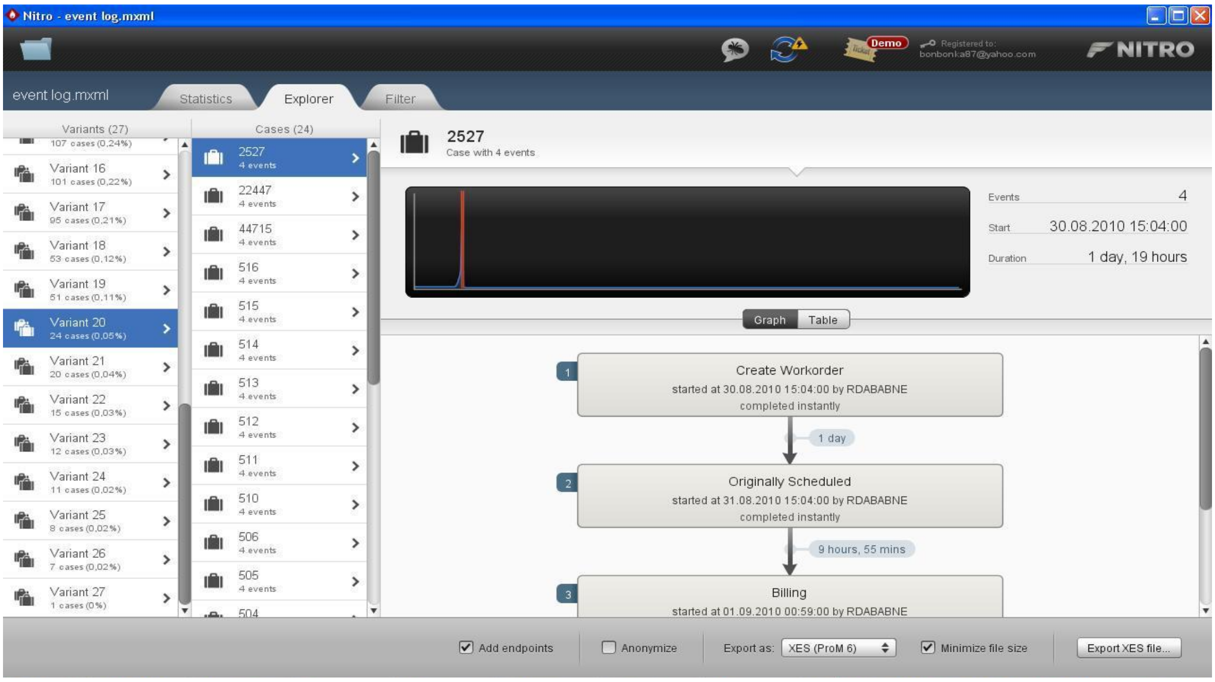


Figure 4.4 Nitro Explorer of an event log

4.3.3 Conformance analysis

As already mentioned a few times, it is clear that reality often deviates from predefined models. This is the reason why it is very important for the management of the company to check the conformance of the process itself to the hand-made in advance model. This is where the replay analysis technique helps to measure the conformance of an event log to the model. During the check *skipped* or *inserted* activities are possible to appear. Skipped activities are such that have to be performed according to the model but do not take place in the log. The inserted tasks occur in the log but should not appear according to the model (Adriansyah, Van Dogen, & Van der Aalst)**.**

There is a plug-in available in the ProM framework to support this – “replay a log on a Petri net for conformance analysis”. The inputs necessary for the analysis to take place are an event log and a Petri net. When they are loaded in, a “mapping Petri net – log” window appears where the transitions in the net have to be set to correspond to the events in the log. The result is an object that contains all alignments between the log and the net, visualized by projecting to the original traces as shown on figure 4.5.

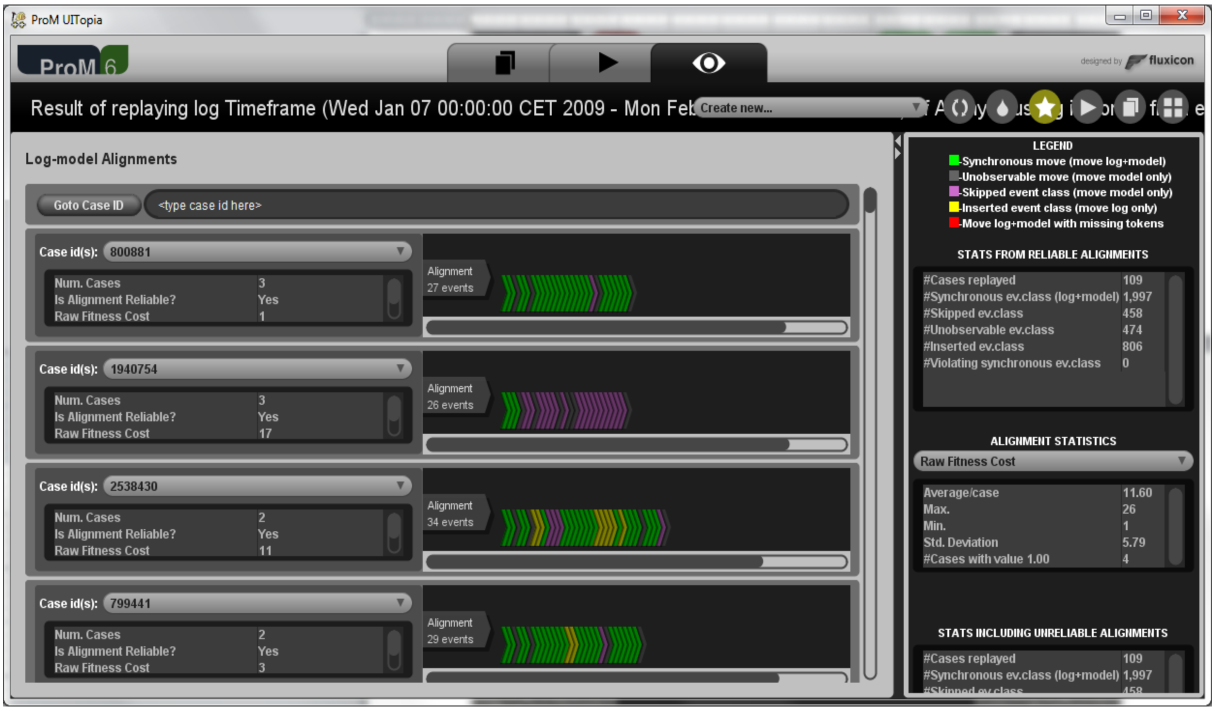


Figure 4.5 Replay a log on a Petri net for conformance analysis

4.4 Performance analysis

The performance analysis phase of the process diagnostics method searches for bottlenecks in the activity chains. It aims to find the weak places in the processes and to discover possible reasons for problems that appear. Nitro can help for throughput times of individual activities and the process itself to be calculated and analyzed (Bozkaya, Gabriels, & Van der Werf).

Among all of the analysis plug-ins that are included in the ProM framework and those which can be of high importance for the performance analysis of business processes, the most important plug-ins will be explained and used in the case study, i.e. the basic performance analysis (section 4.4.1), the dotted chart analysis (section 4.4.2), the basic log statistics (section 4.4.3), the LTL-checker (section 4.4.4).

4.4.1 Basic performance analysis

The basic performance analysis plug-in is supported by ProM 5.2 and measures parameters such as the execution time of an activity, the waiting time, etc. as it can project them on a few different kinds of graphs. Therefore, an obligatory condition is that the event log needs to contain timestamps as already mentioned earlier. When loading the plug-in, first the performance measures need to be configured and then the results appear in the form of various graphs. Figure 4.6 shows the configuration panel.

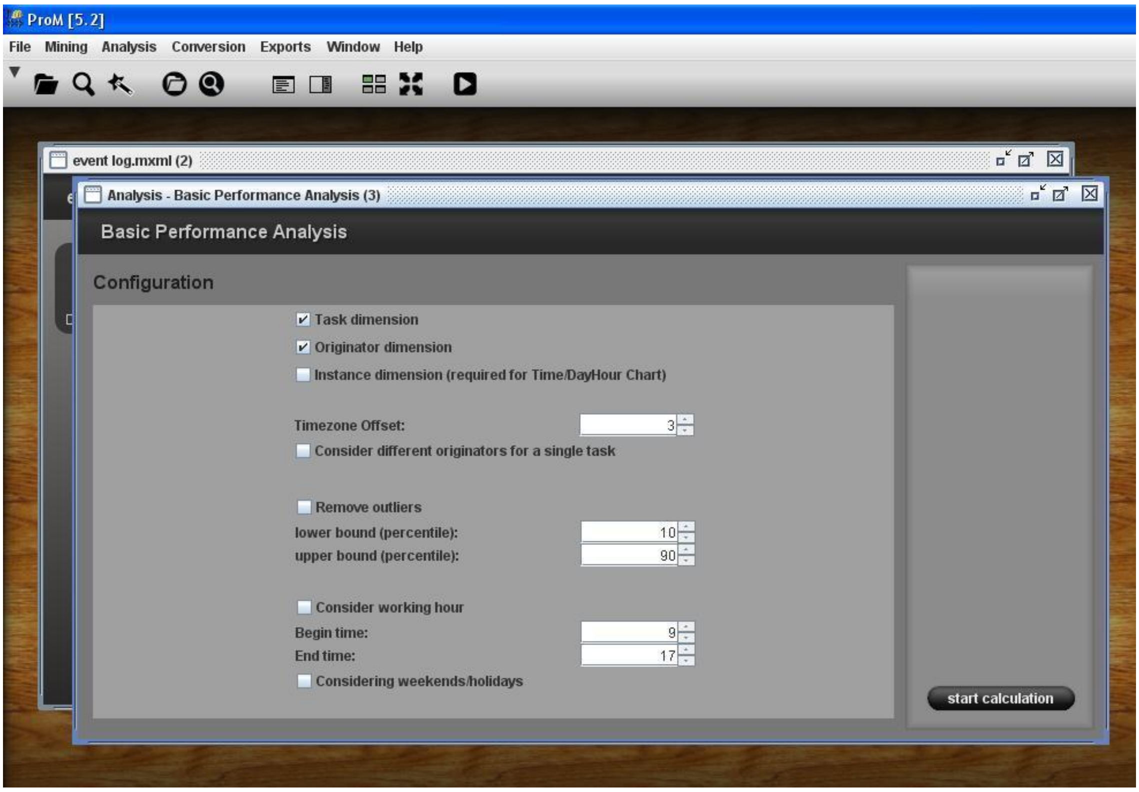


Figure 4.6 Configuration panel of the basic performance analysis plug-in

First, three different **dimensions** for the performance measures of tasks (execution time, waiting time, etc.) can be selected. If the user selects the originator dimension for example, the execution times are shown along with the respective originators. More than one dimension can also be selected and then the result is displayed considering all the dimensions selected (www.processmining.org).

The **timezone offset** is the timezone of a place when the log was created. This data is usually indicated as the timestamp in a log and is important when working hours are considered at different locations.

The **remove outliers** field allows for removing outliers after the performance measures are calculated.

The **working hour** reflects the office hours in an organization. If not considering working hour, the execution time of an activity which has started on thursday at 16h and has ended on friday at 11h is going to be wrongly calculated 19 hours instead of 3 hours for instance (in case the office hours are 09:00 – 17:00h). Therefore it is very important to take into account the working hours of a company so that there will not occur any misunderstandings regarding the amount of time for which a task had been completed (www.processmining.org).

Weekends and **holidays** can also be taken into consideration. Checking this option, if there are days free of activities, they are assumed to be either weekends or holidays (www.processmining.org).

When all the appropriate options are chosen, the **start calculation** can begin in order different result graphs to appear. There are 9 different result charts, including two main performance measures – measures for tasks (processing) and measures for instances (throughput). The different types of charts are explained in Appendix 2 (www.processmining.org).

4.4.2 Dotted chart analysis

The dotted chart analysis plug-in provides a broad picture of a process. It presents a helicopter view of the activities that take place within a process. The chart is used to compare cases and throughput times. The x-axis represents the time of the event and the y-axis shows the *classifier* (component type) of an event where a classifier is a function that maps attributes of the log. Examples of classifiers are: ‘name of activity’, ‘case ID’ of an event, ‘resource’ (originator) of the event, etc. Lines in the chart correspond to the classifier chosen, i.e. if the component “originator” is selected in the options, every line will correspond to a resource which executed an activity. A dot on the chart represents a separate event of the log (www.processmining.org), (Van der Aalst, 2011).

The graph illustrates metrics related to the events and their distribution over time. The two kinds of performance metrics here are metrics for the overall process log and for each component. This already provides helpful insights in the performance of the process. Concerning the overall event log, the *position of the first event in the log*, *position of last log*, *average spread*, *minimum spread* and *maximum spread* can be calculated. Regarding the component types, *the position of the first or last event in a component*, *average*, *minimum* or *maximum interval* between events can be calculated. Figure 4.7 illustrates an example of a dotted chart (www.processmining.org).

The dotted chart analysis is a powerful tool that ensures view of a process from various angles. All events can be seen at a glance and meanwhile different perspectives present – color, shape, time, etc. The chart is accepted to be an example of a visual analytics technique. With its help conclusions can be made about the performance of processes or causes for problems to be investigated. Weak or high performing originators can be detected according to the examined aspects. For instance, a high average throughput time of a certain resource can be captured that can be perceived as weak performance of a task of this person (Bozkaya, Gabriels, & Van der Werf), (Van der Aalst, 2011), (www.processmining.org).

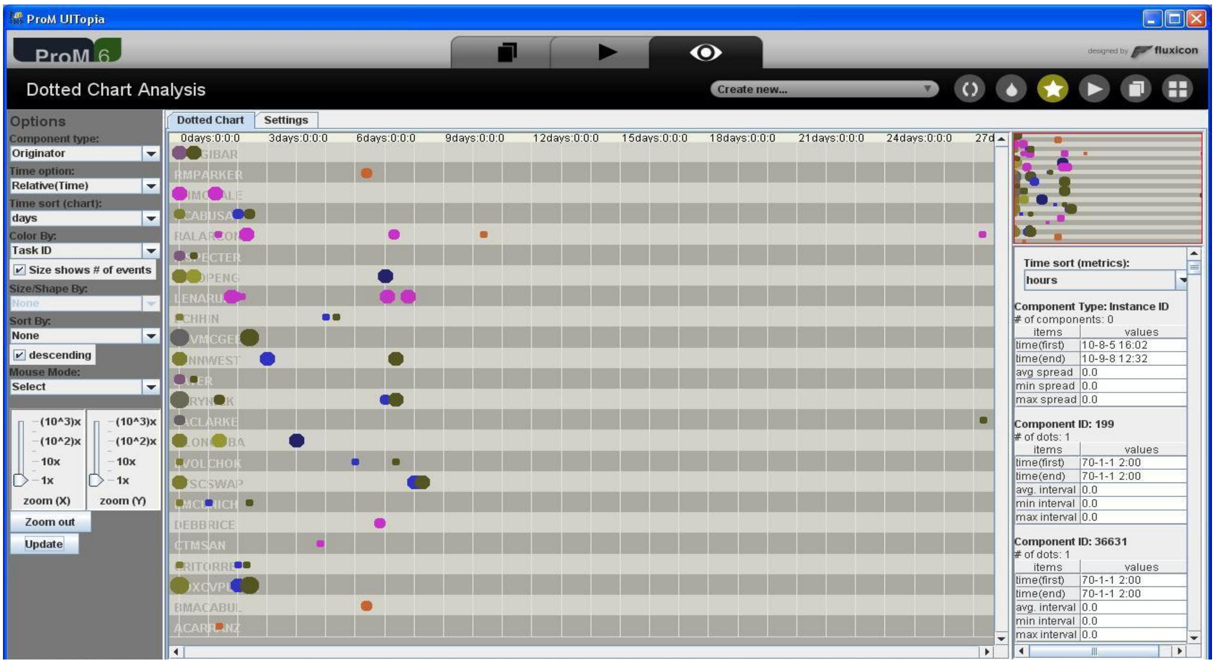


Figure 4.7 Dotted chart Analysis plug-in

4.4.3 Basic log statistics

The basic log statistics is a plug-in available in ProM 5.2 that presents basic statistics concerning the execution times and values of data attribute (figure 4.8). The results from the mining of execution times can be presented either as a textual overview or in graphics. Either results for the whole log (*Global Activity Statistics*) or for individual process instance (*Instance-wise Activity Statistics*) can be observed (www.processmining.org).

Values of data attributes included in the event log are evaluated and measures like frequencies of nominal values, minimum or maximum values are calculated.

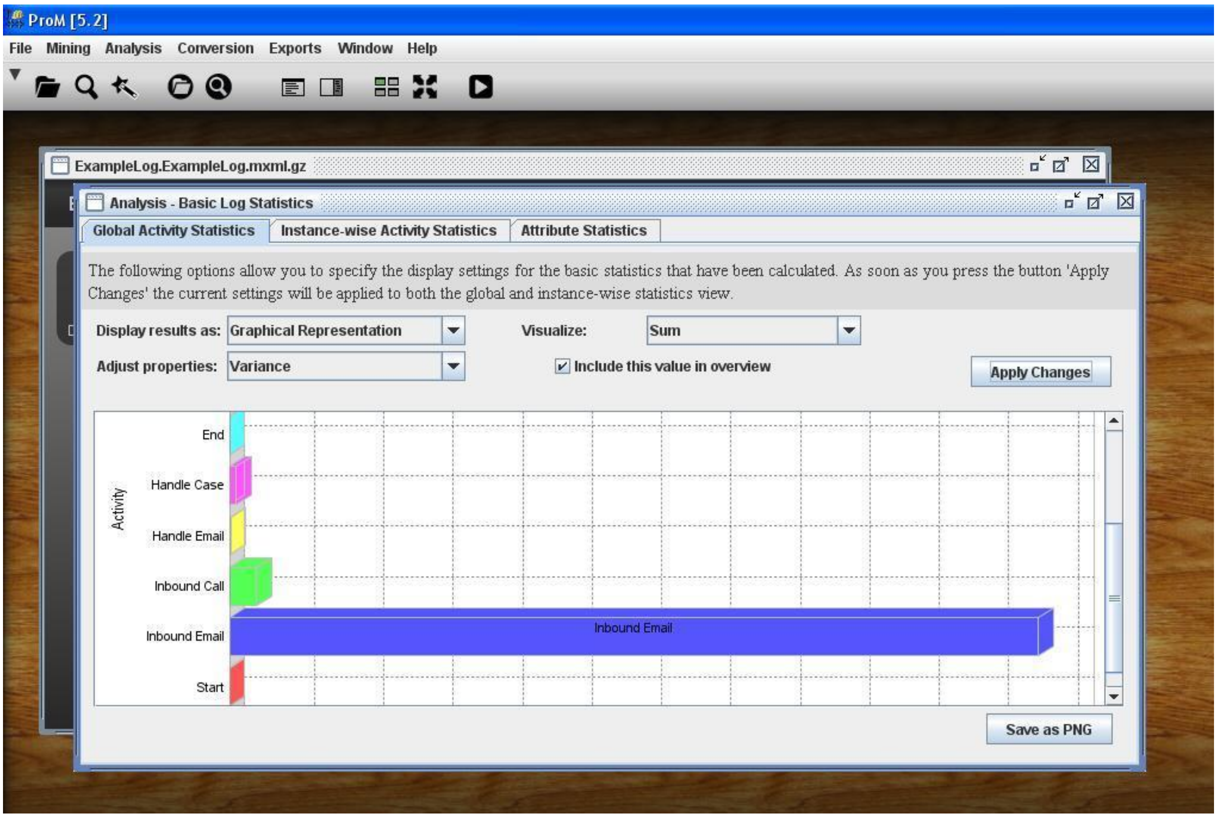


Figure 4.8 Basic log statistics plug-in

4.4.4 LTL checker

Executing a great number of processes through an organization and willing to be aware of their behavior, it is interesting and sometimes even necessary to be checked whether certain properties hold or not. Thus, a necessity had appeared for a concrete language which formulates dynamic properties to be modeled. The language (Linear Temporal Logic) aims to specify properties of events (Van der Aalst, De Beer, & Van Dongen).

The *LTL checker* is a plug-in that has been developed to focus on verification. Given an event log and some property, it should be verified if this property holds. It is therefore considered to be an analysis plug-in of the ProM framework (Van der Aalst, De Beer, & Van Dongen), (www.processmining.org).

About 60 properties exist that focus on activities performed, people that executed them (originators) and event attributes. Here the realization of the already mentioned earlier "four-eyed principle" can be followed (www.processmining.org).

The LTL checker, like a lot of the ProM plug-ins available, also gives the opportunity for a single case to be checked in isolation if the user wants to focus on a specific process instance (www.processmining.org).

Figure 4.9 express satisfied and unsatisfied rules (first two columns), set with the opening of the plug in, where the cases that correspond to them are shown and a case itself can be examined (last column on the right).

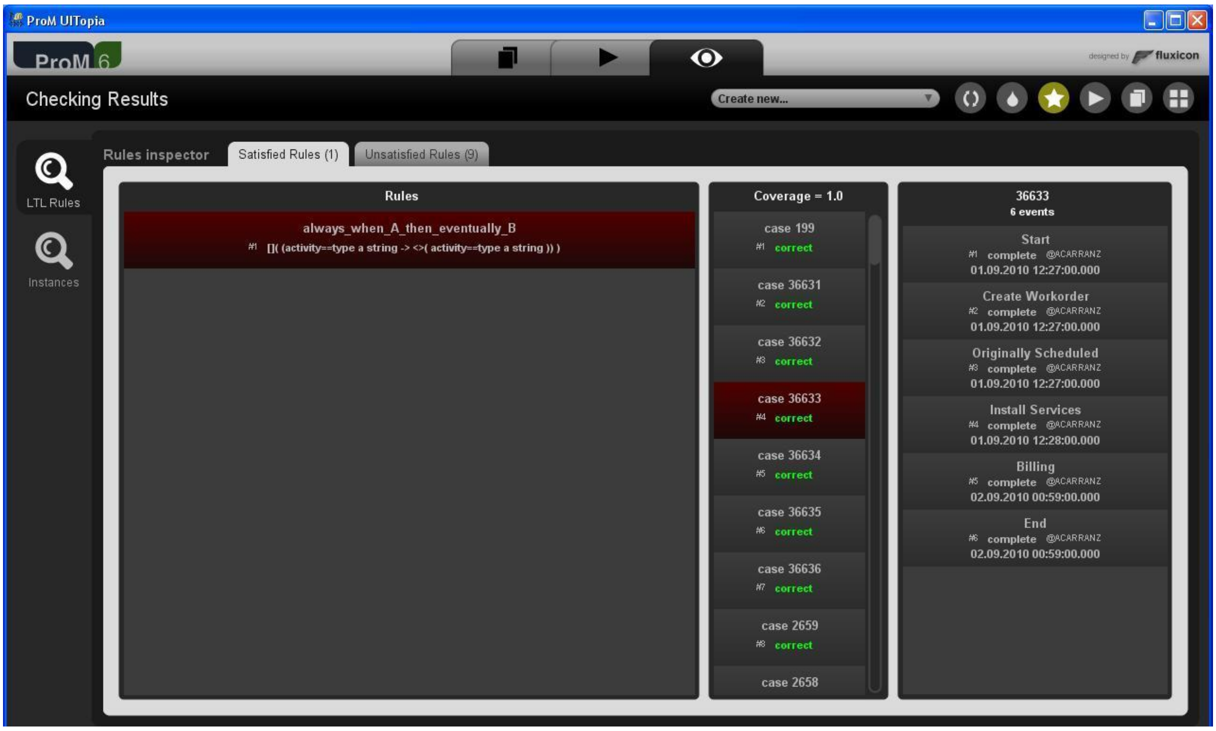


Figure 4.9 LTL checker

4.5 Role analysis

Finally, the role analysis can be performed as a last step of the process diagnostics. The requirement for this phase is that the event log includes resource data about the originator of each activity that occurs in the process. With the help of a few plug-ins available in the open-source software, different information can be obtained such as the division of work between departments, execution of similar activities in different departments, role hierarchy, relationships between different roles in the organization, team working, etc. Here, the social network miner (section 4.5.1) and the role hierarchy miner (section 4.5.2) are explained.

4.5.1 Social network miner

Social network analysis (SNA) of business processes can be performed if an event log includes the resource attributes (originators of the tasks in the process). The network reflects the work between people in an organization. Resource-related models can be constructed by mining the organizational perspective of process mining explained earlier in this thesis. Nodes in a social network graph correspond to organizational entities like roles, groups or departments and arcs reflect the relationships between these entities. Nodes and arcs might have weights that indicate their importance in the process (Van der Aalst, 2011), (www.processmining.org).

The *social network miner* (Figure 4.10) is a mining plug-in that was constructed to serve as a base and support the SNA. It provides a few techniques which functionality helps the analysis of social networks. Five different ways of mining are available in ProM 6.1 that are connected to the SNA. These are ‘handover-of-work social network mining’, ‘reassignment social network mining’, ‘similar-task social network mining’, ‘subcontracting social network mining’ and ‘working-together social network mining’. For all of them, different options can be set, e.g. layout projection, ranking, etc. Shape or size dependencies can be specified as well as choice to show edges or weights for instance (www.processmining.org).

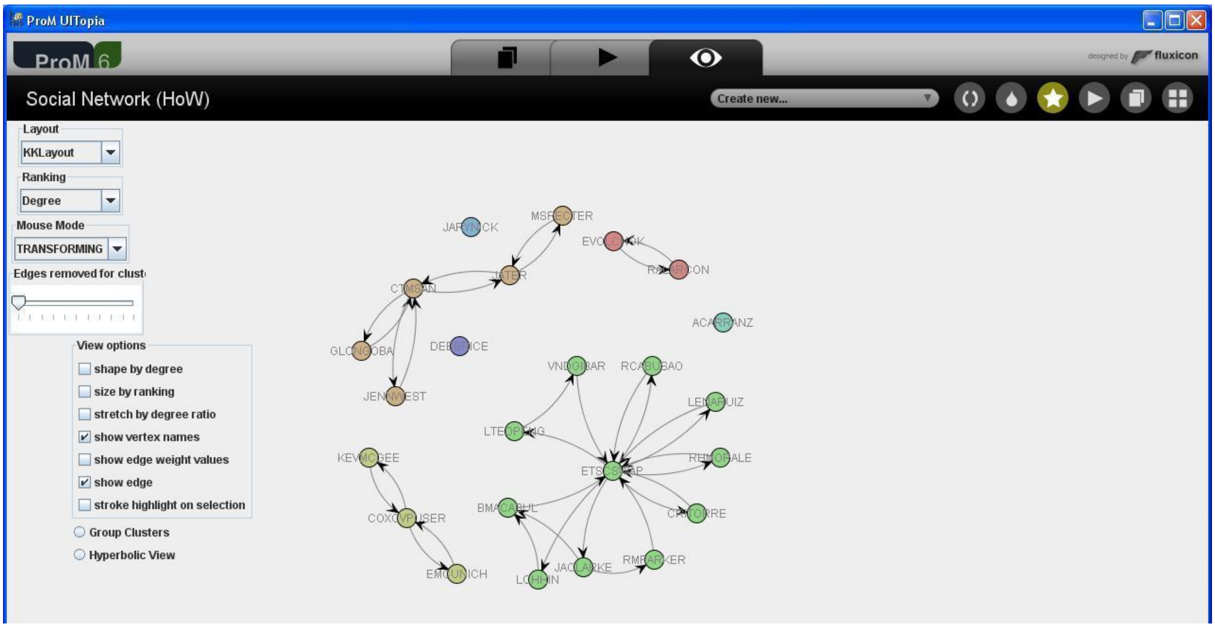
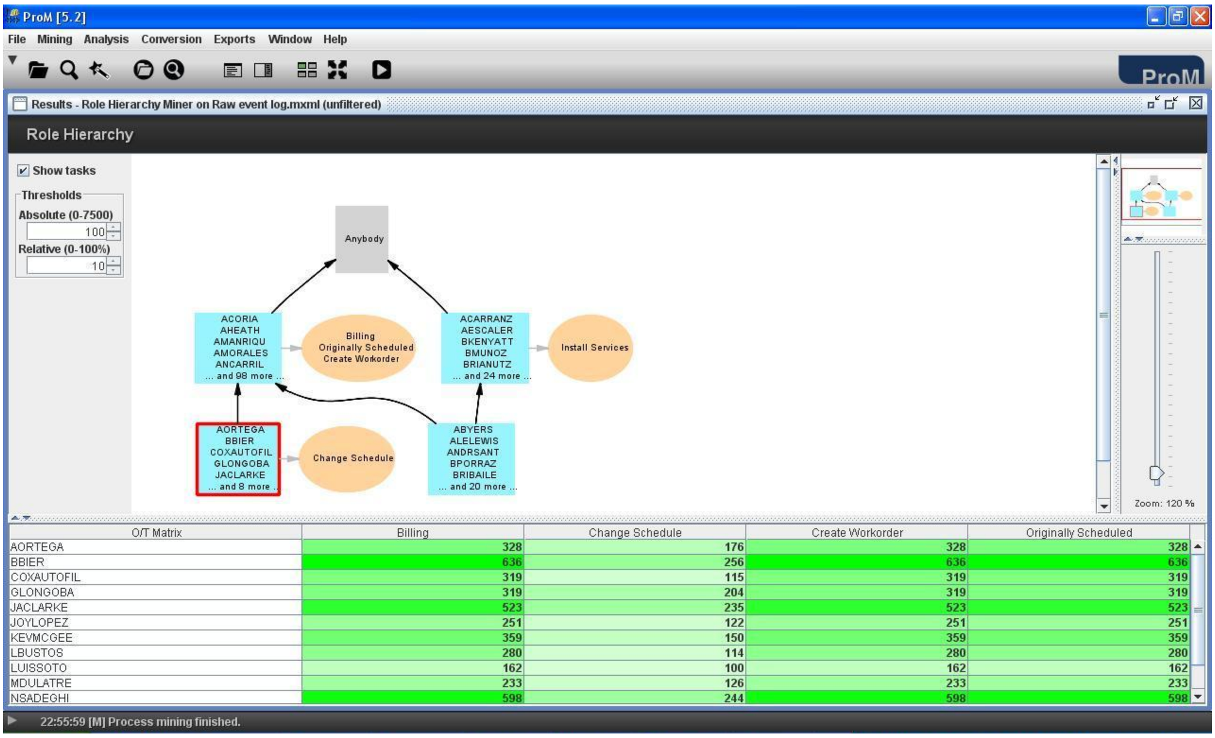


Figure 4.10 Handover-of-work social network

4.5.2 Role hierarchy miner

The role hierarchy miner is a plug-in available in ProM 5.2 that allows for the role analysis of the processes running within the enterprise. The hierarchy is based on the various activities which are executed by different roles. The graph represents which activities are performed by whom. For instance there are resources executing just a few activities but frequently. These are considered to be some specialists in the organization. Other originators may execute a lot of different tasks that take part in the process. This means these people are probably generalists in the company. By observing the different activities and people that performed them, a hierarchy can be drawn reflecting the execution of tasks and also the number of each activity initiated by every person is indicated in a table under the graph (see Figure 4.11) (Bozkaya, Gabriels, & Van der Werf).



O/T Matrix	Billing	Change Schedule	Create Workorder	Originally Scheduled
AORTEGA	328	176	328	328
BBIER	636	256	636	636
COXAUTOFIL	319	115	319	319
GLONGOBA	319	204	319	319
JACLARKE	523	235	523	523
JOYLOPEZ	251	122	251	251
KEVMCGEE	359	150	359	359
LBUSTOS	280	114	280	280
LUISSOTO	162	100	162	162
MDULATRE	233	126	233	233
NSADEGHI	598	244	598	598

Figure 4.11 Role hierarchy diagram

Chapter 5 – Case study

In order to explore the powerful force of process mining in digging business processes and giving the opportunity for their detailed analysis while looking at different aspects and perspectives, a case study will be put through. With the help of different ProM plug-ins and Nitro, the added value of process mining is tested in the context of performance analysis of a certain business process. The framework that will be used through this chapter is the process diagnostic framework explained in the previous chapter. Some basic statistics will be observed together with detailed analysis over the various relationships that take place in a process concerning different activities, originators, time, etc. The base of the case study will be event log information, granted from a real-life operating company in the USA.

5.1 Short description

Cox Communications, Inc. is a subsidiary company of Cox Enterprises which provides digital cable television, Internet, telecommunications and wireless services. The company operates in the United States as a whole but for the purpose of this thesis the San Diego area, California operations will be examined and only the Residential Service business processes will be analyzed.

As many other companies, Cox Communications aims to implement well-performed business processes. To achieve this, the company accomplishes several activities to support its daily operations. These are divided into primary and support activities and are transformed into business processes. The core business of the company is the *work order* process (see fig. 5.1) which includes primary activities (sales, system operations centre, customer care, etc.)

5.1.2 Work order process

The core process starts in the ‘sales’ department where orders are received and created. A date for installation is then set and scheduled on the work order.

In the ‘dispatch’ department a technician is appointed and if the customer wants to reschedule the date, the work order is updated. Otherwise it directly goes to the ‘field sales &

service’ department for installation. A technician is send to install the hardware and cable at the customer’s home and after an engineer in the ‘system operations center’ department opens a signal from Cox Network for the customer. The technician makes sure everything works properly at the end and the work order status is turned to “completed”. Finally the billing is initiated every month and continues until the disconnection of the service.

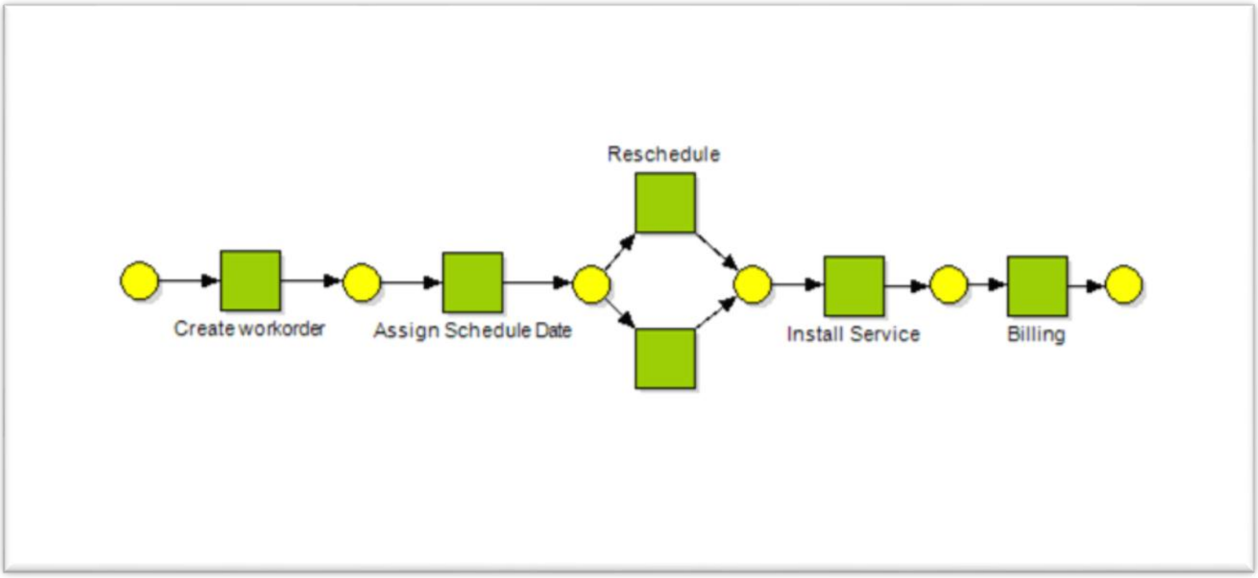


Figure 5.1 Work order process at Cox

To maintain optimal performance and customer-service levels, Cox has developed a huge data warehousing application where the work order data is stored in a single table within the database.

In managing the Work Order process, Cox has assigned an outsourcing IT consulting company to customize a Work Order Management System called ICOM. The software is used to simplify the work order assignment. The system is also equipped with a billing system to serve the monthly customer billing.

The data file granted from Cox Communications is generated from the system explained above in a Microsoft Excel spreadsheet format. It contains around 65 000 instances. Each record contains information about a unique identifier number indicating the case, the activity executed, the originators who executed it and the timestamps. Further, some additional attributes are given.

5.2 Log creation

As already mentioned earlier the data file does not come in a readily available event log format to be load in the ProM framework. Once extracted from the information system, it needs to be cleaned and converted from a data file to an .mxml or .xes event log file. After analyzing the data in the event log from Cox, it is discovered that not all of the fields available are relevant for process mining and some of them do not even contain any values and have no meaning. Therefore, some attributes are removed from the initial file.

In the current case the Nitro software is going to be used to convert the .csv file from the US company into a .mxml format that can be further used in both Nitro and ProM.

The first thing to do is to import the excel file in Nitro. Figure 5.2 shows the information included in the log in a table with a title for every column (activity, timestamp, user ID, etc.). Each title of the columns need to be linked to a label (the identification of an attribute). In Nitro there are a few labels already set (*case*, *activity*, *timestamp*, *resource* and *other*) and for any other additional information about each activity that the user wants to include, the "other" label can be put. In this case the first column is going to be set as "case ID" by selecting the column and the corresponding label "case ID". This automatically refers all the rows of the first column to case IDs that are going to be included and ready for analysis in the event log from now on. The second column of the table "activity" is respectively marked with the "activity" label available in Nitro. The work order number, WO type, monthly dollars, new rate and service category code (all included in the .xls file) are given the "other" label. The salesman number and user ID both are automatically included in the "resource" label and the timestamp is put under the label of "timestamp". When the user wants to set the timestamp under this label, a window appears that asks for setting of the time pattern used in the log (see fig. 5.3). In the current case the time appears as "dd/mm/yy HH:mm" so this is the pattern that is going to be set for the time.. A user can always choose not to include some of the information in the log by deleting them in this phase. In this case the "service code" column is chosen not to be used as it will probably not bring any added value for the purpose of the case study.

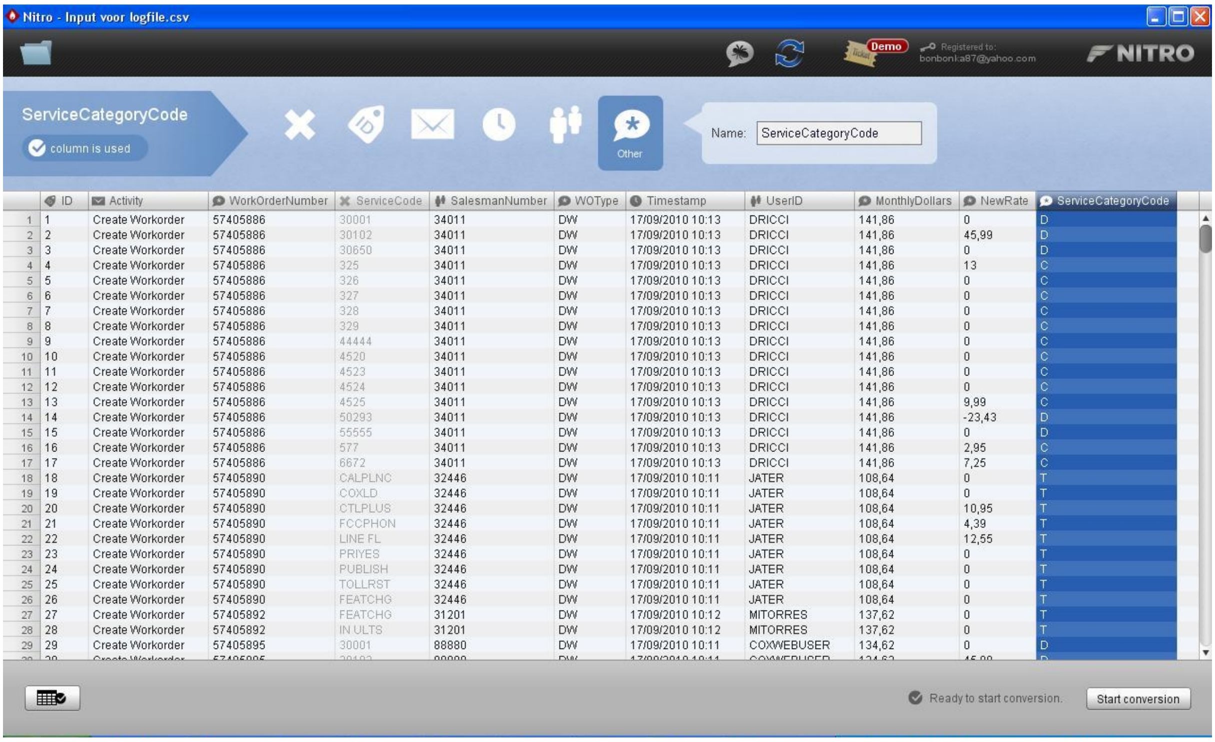


Figure 5.2 Attributes information table in Nitro

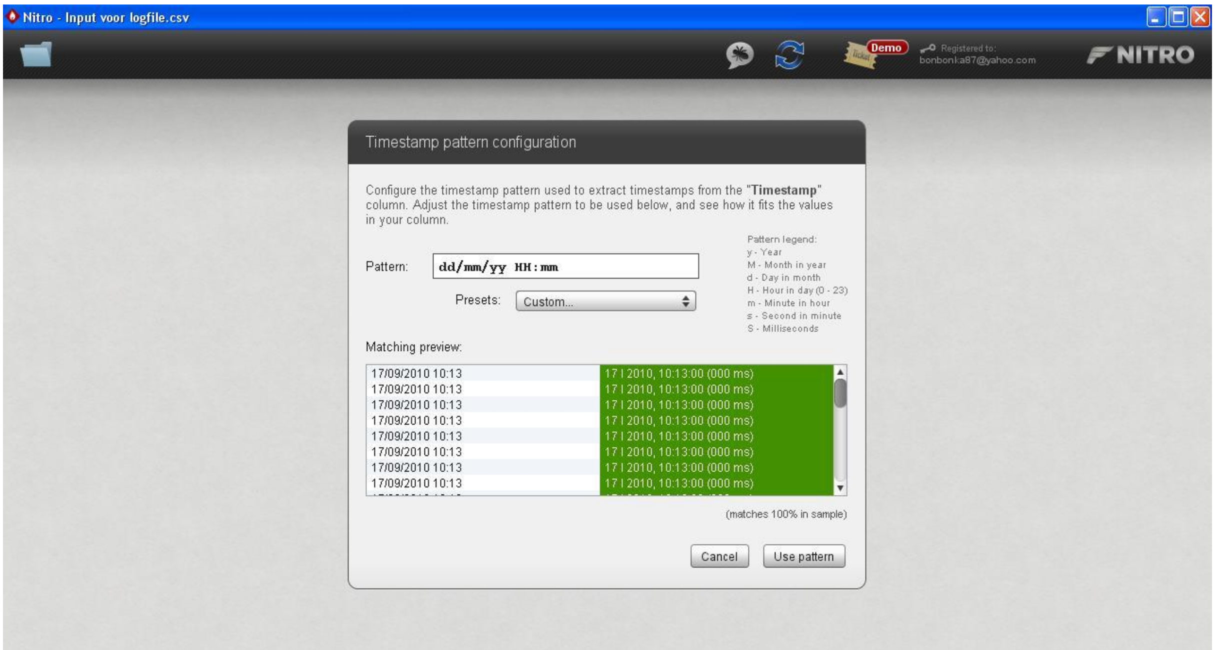


Figure 5.3 Timestamp pattern configuration in Nitro

The file is now ready for conversion. After the creation of the event log, the second part of Nitro opens which can be used to analyze the created event log. On the bottom of the window a user can find a few options for saving the file in the desirable format. The output can either be .mxml, .xes or .csv format. In this case an .mxml file is going to be extracted as further in the thesis, plug-ins from ProM 5 are also going to be used and this edition of the framework only supports the .mxml and not the .xes format. In ProM 6.1 both an .mxml and a .xes file can be used.

5.3 Log inspection

In this phase the event log is going to be initially examined and the process it reflects is going to be discovered. As already mentioned the initial examination can be obtained by the use of the Nitro commercial tool and its basic statistics functionality and the process discovery is done by using the fuzzy miner plug-in included in the ProM framework.

5.3.1 Nitro

After the conversion of the .csv file to an event log, the statistic overview appears immediately in Nitro. The general statistics can be found in Table 5.1 The event log consists of 189 690 events which are executed for 45 000 different cases. Further, there are 5 different activities in the event log as already explained in section 5.1 (‘Create WO’, ‘Assign a schedule’, ‘Reassign a schedule’, ‘Install services’ and ‘Billing’). There are 602 resources that have executed the events and 9 attributes are included for each activity.

Table 5.1 General statistics of the event log

Events	189 690
Cases	45 000
Activities	5
Resources	602
Attributes	9
Start date	30/11/2009
End date	07/06/2018

Looking at the start an end date of the event log (i.e. 30.11.2009 and 07.06.2018) the first log error is indicated. This is the end date that appears to be in the future which is impossible. Another issue that makes impression is the fact that since the start date (30.11.2009) till half a year later no events have been recorded per day. This also indicates an error in the recording in the corresponding information system because it is not realistic that no data was recorded for 6 months. For both these reasons, the event log is going to be filtered based on the time of the events. This can be done by using the timeframe filter in the "Filter" tab of Nitro where a start date is chosen to be 1st December 2009 and the end date – 1st December 2010. In the timeframe filter besides the start and end date, different cases can be indicated to be kept. In this situation, cases "contained in timeframe" will be saved in the event log. So, only cases started after 1/12/2009 and finished before 1/12/2010 will be kept. In this way, it is sure that only complete cases are used for further analysis.

After the filtering, the number of events decreases to 186 541. They appear in 44 366 cases and still 602 resources are indicated to perform the activities (table 5.2). The cases mostly take place in July, August and September 2010.

Table 5.2 General statistics of the event log after filtering

Events	186 541
Cases	44 336
Activities	5
Resources	602
Attributes	9
Start date	04/06/2010
End date	01/11/2010

When observing the "Activity statistics" in Nitro, a positive indication is spotted considering the *relative frequency* of the activities in the different cases. It seems that all of the cases (44 366) include the main activities "create work order", "originally scheduled", "install services" and "billing". In almost ¼ of the cases, the customer has decided to reschedule the originally scheduled date. The appearance of all the necessary activities in every single case of the log shows right business activity and supposes completion of all the cases that have appeared during the working process. Table 5.3 presents the frequency of all the activities, included in the process. On the contrary, the “first in case” and “last in case” columns (tables 5.4 and 5.5) that refer to respectively first and last activities that appear in the cases, indicate strange issues, e.g. “billing” and “change schedule” appearing as first activities in a case and “originally scheduled” and “change schedule” figuring as last activities in a case. When recording events in information systems, sometimes the time is wrong noted by the system which means that for some activities that take place in one and the same day and at close hours, the system can change places (the sequence) of these two activities. This is a possible reason for the issue above mentioned but can also be a result of wrong entering of information in the system by the user or the appearance of unforeseen circumstances in the last minute. Further analysis of the sequence of tasks is going to be performed in the next step – control flow analysis of the process.

Table 5.3 Frequency of the activities in the work order process

Event class	**Frequency**	**Relative frequency**
Create work order	44 366	23.78 %
Originally scheduled	44 366	23.78 %
Install services	44 366	23.78 %
Billing	44 366	23.78 %
Change schedule	9 077	4.87 %

Table 5.4 First activity in a case

Event class	**Frequency**	**Relative frequency**
Create work order	44 366	23.78 %
Billing	44 366	23.78 %
Change schedule	9 077	4.87 %

Table 5.5 Last activity in a case

Event class	**Frequency**	**Relative frequency**
Originally scheduled	44 366	23.78 %
Install services	44 366	23.78 %
Billing	44 366	23.78 %
Change schedule	9 077	4.87 %

Exploring all of the rest of the statistics overview, an interesting fact appears regarding the price and rate of the services provided. In a lot of cases, the “monthly dollars” are introduced with “0” value which indicates no fee in some cases. On the one side, this can be a result of some partnerships between Cox and some customers as well as some “loyalty” packets or just an option of “prepayment” for a few months. On the other side, this can be some weak point in the process and specifically the billing activity that may need to be traced. There is a bad possibility of a resource to have forgotten to activate the service or the billing itself. What else, there may be a problem with the billing system adopted and if so, measures have to be taken regarding this.

In Nitro, also the different variants in an event log can be explored. A variant can be seen as a different sequence of activities. According to the “case variants” information, there are 25 variants of the sequence of activities in a case and approximately 5 of them are mostly taking place in the log. This aspect of the log is going to be examined in details in the “control-flow” phase of the process diagnostics.

Important information that can also be obtained by the Nitro software is the duration of a case. This data shows the average time, necessary for the company to handle a customer’s order which gives key insights to the management and all the stakeholders. In cases where the order needs more time to be handled, reasons for low performance has to be searched for and analyzed. According to the event log information, the case duration of the process in Cox Communications usually lasts for one to four days. There are also a lot of cases that are completed for six to thirty six days but they are significantly less.

The inspection of the resource information presented in Nitro reveals participation of 602 originators taking part in the work order process in Cox. Table 5.6 presents the basic frequency statistics of the resources.

Table 5.6 Basic resource statistics in Nitro

Values	602
Minimal frequency	2
Mean frequency	309.87
Maximum frequency	8 356

About 13 of the originators that take part in the process have relative frequency 1-5 %. All the others show appearance less than 1%. The most often appearing originators together with their frequency percentage can be seen in table 5.7.

Table 5.7 Resource frequencies

Event class	**Frequency**	**Relative frequency**
CTMSAN	8356	4.48 %
ETSCSWAP	7287	3.91 %
COXCVPUSER	5579	2.99 %
MSPECTER	5311	2.85 %
BHANDLEY	3048	1.63 %
GOTHOMAS	2830	1.52 %
BRMOORE	2475	1.33 %
BBIER	2151	1.15 %
RDABABNE	2126	1.14 %
NSADEGHI	1996	1.07 %
MGREENST	1926	1.03 %
JACLARKE	1888	1.01 %
HOANTHAI	1859	1 %

5.3.2 Fuzzy miner

To get an idea of the real process behavior, the process is discovered by using fuzzy miner. The *"Mine for a fuzzy model"* plug-in is going to be used for this purpose

Figure 5.4 shows the process model extracted from the event log.

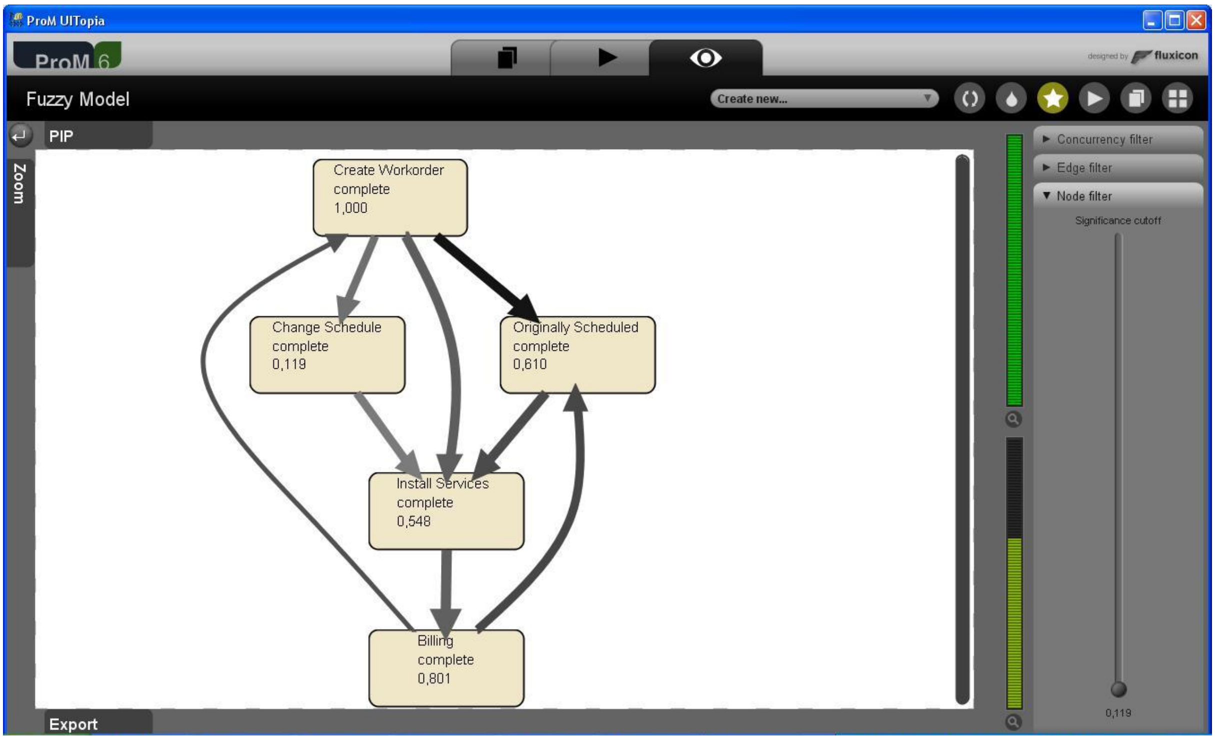


Figure 5.4 Fuzzy model of the Work Order process in Cox

The node filter is set to the minimum such that the detailization and complexity of the model to be at the highest level.

At first sight the process model seems right and compatible with the predefined work order process model (Figure 5.1). However, examined in details, some strange behavior is detected in the process model obtained from the log compared to the predefined one. The occurrence of creation of a work order after the billing task for example is a strange phenomenon as normally the billing has to be based on an already created customer order. In order to be followed how many cases appear where the billing takes place before the work order was created, a rule is going to be set in the LTL checker plug-in. The rule states "*always_when_A_then_eventually_B*". "A" and "B" refer to activities that can be defined by the user in the settings window of the plug-in. In this case A is defined as "billing" and B is defined as "create work order". After loading the plug-in, the result appears (figure 5.4) showing that 1116 process instances are correct, i.e. in 1116 cases the creation of a work order follows the billing. This amount does not represent a large percentage of all the instances which means it should not be considered highly embarrassing in the process analysis. Another sequence that is not typical is the appearance of the “originally scheduled” task after the billing. According to the LTL checker this arrangement takes place in 14 086 cases (out of 44 336) which is more than one fourth of all the instances and this becomes worrying and has to be further analyzed and searched for a reason. A strange issue is also that installation of services sometimes follows directly the creation of a work order without a date has been scheduled before the installation.

The thickness of the arrows represents the frequency of the relations between activities and according to the analysis of the obtained process model from the log, the strange behavior does not take place as often as the regular sequence of tasks that was determined to take place in advance, i.e. *create work order – schedule/reschedule – install –services – billing.* According to the color of the arcs, the most significant relations in the process are these between a creation of a work order and originally schedule of a date as well as the one between “originally scheduled” and the “install services” tasks. Less significant relations seems to appear where the “change schedule” task takes place in the process.

Following the weight values shown under each task, the most significant activities for the process are the creation of a work order, the “originally scheduled” step, the billing and installation of services. The change of schedule date has low significance in the activity chain as its weight is just 0.119 out of 1.000.

Some specification that can be done so that all the possible behavior in the event log can be shown is setting of the *cut-off* option of the *edge filter* to the value of 1.0. The default settings of the plug-in are programmed to show the most common behavior of the log. Figure 5.5 illustrates all the possible relations after fixing the cut-off value to 1.0. Now additional strange sequence can be seen such as appearance of billing activity after changing schedule. However, it can be easily spotted that the arrow is a thin one and its brightness is too high which means this relation does not represent high occurrence and significance in the process.

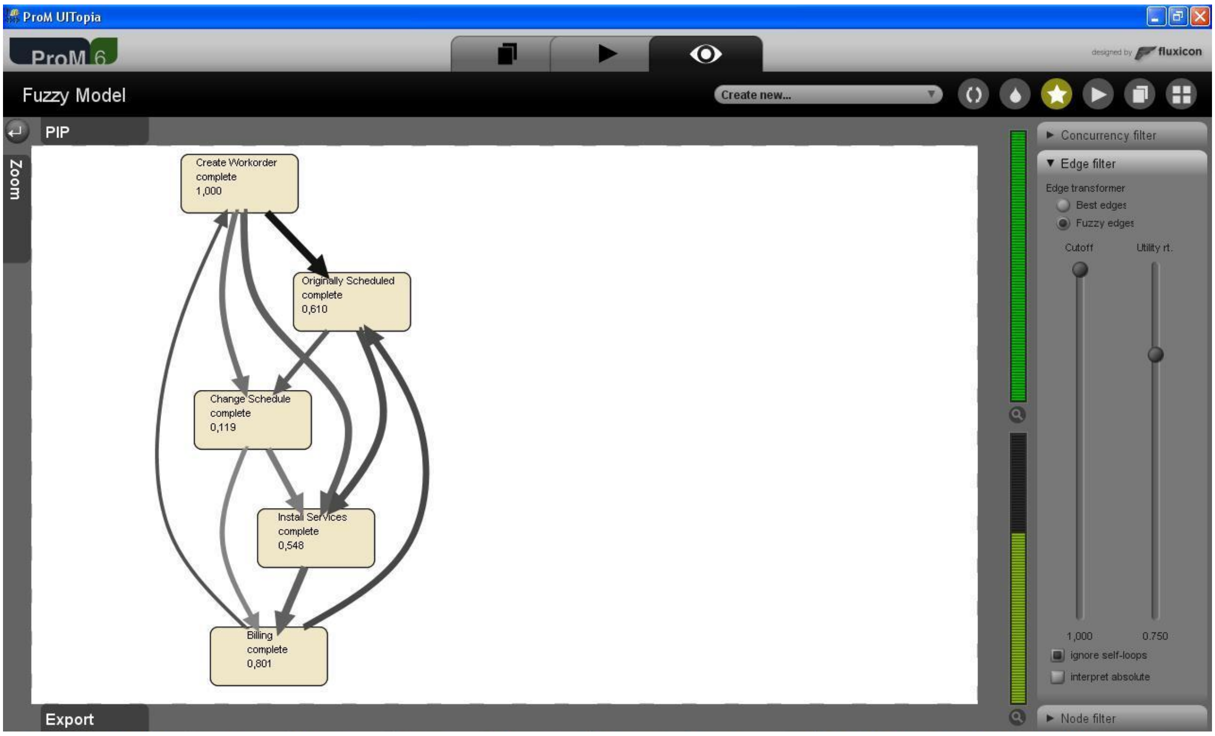


Figure 5.5 Fuzzy model of the work order process (edge filter cut-off set to 1.0)

5.3.3 Summary

The initial inspection of the event log of Cox Communications work order process reveals some infrequent behavior and deviations from the originally constructed process model. This can either be a result of wrong recording of the information system in the enterprise and poor consistency of the applications or may be a consequence of irregular execution of the different steps in the process or negligent attitude by a resource concerning the right completion of a process. The next phases of the process diagnostics aim to deal with the deviations analysis and assume some reasons for their occurrence.

5.4 Control flow analysis

In this phase of process diagnostics, the most frequent behavior (activity paths) will be discovered according to the event log concerning the Work Order process in Cox Communications. The “explore” feature available in Nitro and the performance sequence analysis ProM plug-in are the tools being used in this phase. Infrequent behavior or cases whose sequence of activities does not occur too often are also going to be followed and left out according to the Pareto principle (roughly 80 % of the effects come from 20 % of the causes), i.e. at least 80% of the event log will be covered but it will only contain cases with frequent behavior. This means that only major flows (most often variants) between activities will be shown and used in the event log, enabling easier interpretation.

5.4.1 Nitro

It was already noted that 25 variants (activity paths) appear in the log. This can be easily observed in the "Explorer" tool of the Nitro software. The four most appearing variants are presented in table 5.8. The most frequent variant that appear (48.93%, 21 709 cases) goes after the following path: *create work order – originally scheduled – install services – billing* (Figure 5.6)*.* This indicates correct behavior of the real process flow since these are the four main activities in the predefined model in the correct order. In variant 2 (13.93%, 6 182 cases) the tasks take place as follows: *create work order – install services – billing – originally scheduled.* This does not seem to be typical behavior as it seems strange that the installation of services and the billing appear before the fixing of the installation date. This feature points to some mistake in the process. This can be a delay that a worker had made concerning the registration of the installation date. It seems that in all these cases, after the appearance of the order, the technician who installs the services was sent (probably after setting of date and time with the customer) to the house of the customer, the billing was effected after the activation of the service and finally the originally scheduled installation date was introduced to the information system. A few assumptions can be made concerning this issue: 1) the initiating of an originally scheduled installation date was obstructed by the information system, 2) the system or the originator who executed the task recorded a wrong date of the event or 3) the resource completed the schedule task on the last place in the process.

Table 5.8 Four most frequent variants in Cox Communications work order process expressed by Nitro

Variant	**Sequence of activities**	**Number of cases**	**Percentage**
Variant 1	*create work order – originally scheduled – install services – billing*	21 709	48.93 %
Variant 2	*create work order – install services – billing – originally scheduled*	6 182	13.93 %
Variant 3	*create work order – install services – originally scheduled – billing*	5 900	13.3 %
Variant 4	*create work order – change schedule – install services – billing – originally scheduled*	4 670	10.53 %

Variant 3 represent 13.3% (5 900) of the cases that took place for the period examined (3 months). The sequence of activities in this variant takes place as follows: *create work order – install services – originally scheduled – billing*. When exploring the execution date and timestamps of all of the instances included in this variant, it seems that the "install services" and "originally scheduled" tasks were both recorded on the 1st of September 2010. The reason for the sequence turn of these two activities can be again a result of the fact that sometimes the registration hours in an event log do not appear exactly in the way they were actually recorded. The information system may probably indicate some wrong timestamps or the induction of the data by an originator could be again delayed concerning the "originally scheduled" task and first the activation of the installation had to take place and was registered. There might be a few reasons for all these issues and they have to be observed and analyzed by the management or some people responsible for the process flow in the company. In fact this does not seem to be a major deviation of the originally defined process in the enterprise but still represent different behavior and has to be kept into consideration.

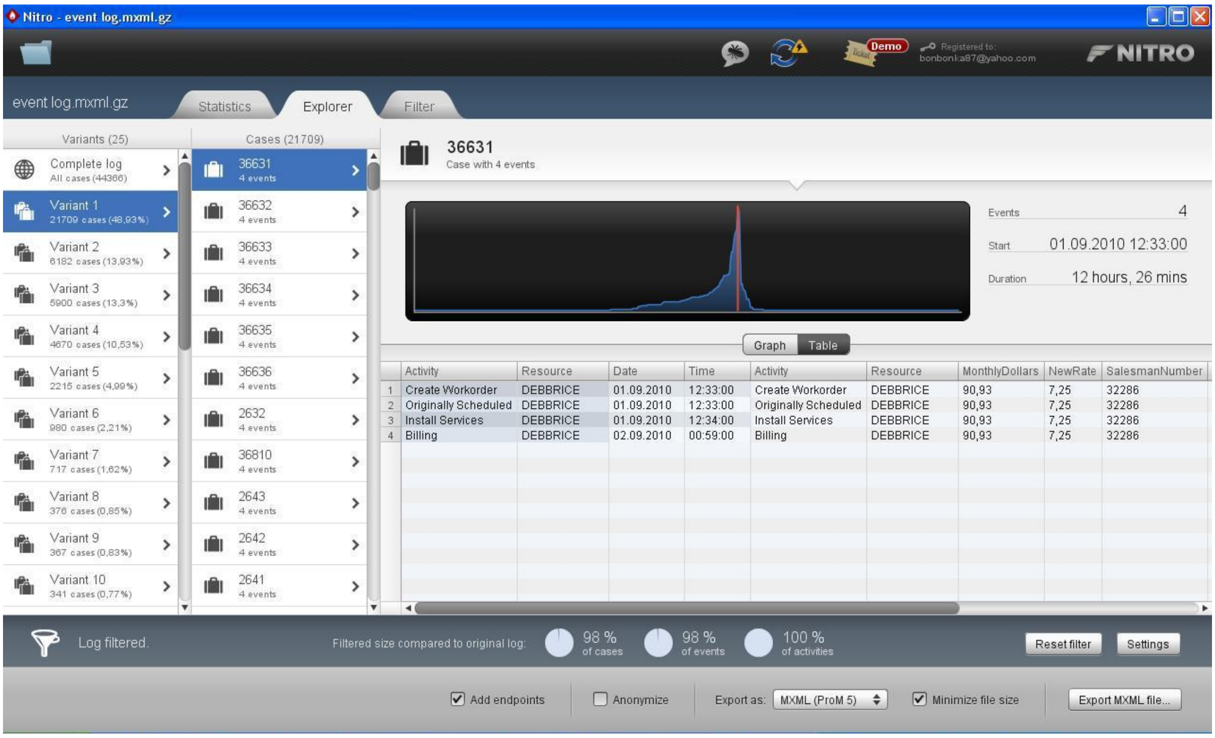


Figure 5.6 Cases variants represented by the Nitro tool

The next variant already includes 5 activities as the "change schedule" also appears there. The strange feature here is that this task takes place immediately after the creation of a work order which is impossible as first a date has to be originally scheduled so that it can be rescheduled after.

Some variants of the sequence of activities represent less than 5% of the whole event data which represents infrequent behavior and are not going to be analyzed in details. Variant 7 for instance (*billing – create work order – originally scheduled – change schedule – install services*) starts with the billing of a service which is actually impossible as the billing has to take place on the base of an already created work order. In these cases, there could have been some reservation of work order numbers in advance for example and the customer could have asked to pay in advance and then install the service. This can already be a result of some internal rules and options in the Cox enterprise (e.g. prepayment of the service).

However, according to the Pareto principle the first four variants are enough to be considered for the control flow analysis to take place, representing 86.69% of the whole event log data.

The check of this event log shows a correct termination of 48.93 % (21 709 cases) starting with the creation of an order, passing through originally scheduled date and installation of services and ending with the billing activity. If only the initiation and completion of the process are taken into consideration, the third variant can also be accepted as showing correct termination of the process. It represents 13.3% of the data so regarding the right start and end of a process, it can be concluded that 62.23% of all the cases represent correct completion.

5.4.2 Performance sequence analysis

After loading the event log file in the ProM 5.2 framework and opening the performance sequence diagram plug-in, the different sequence paths of activities are shown, grouped in the so called *patterns*. This is shown in figure 5.7. The component type is set to be "Task ID" and the time calculated is set to be shown in days.

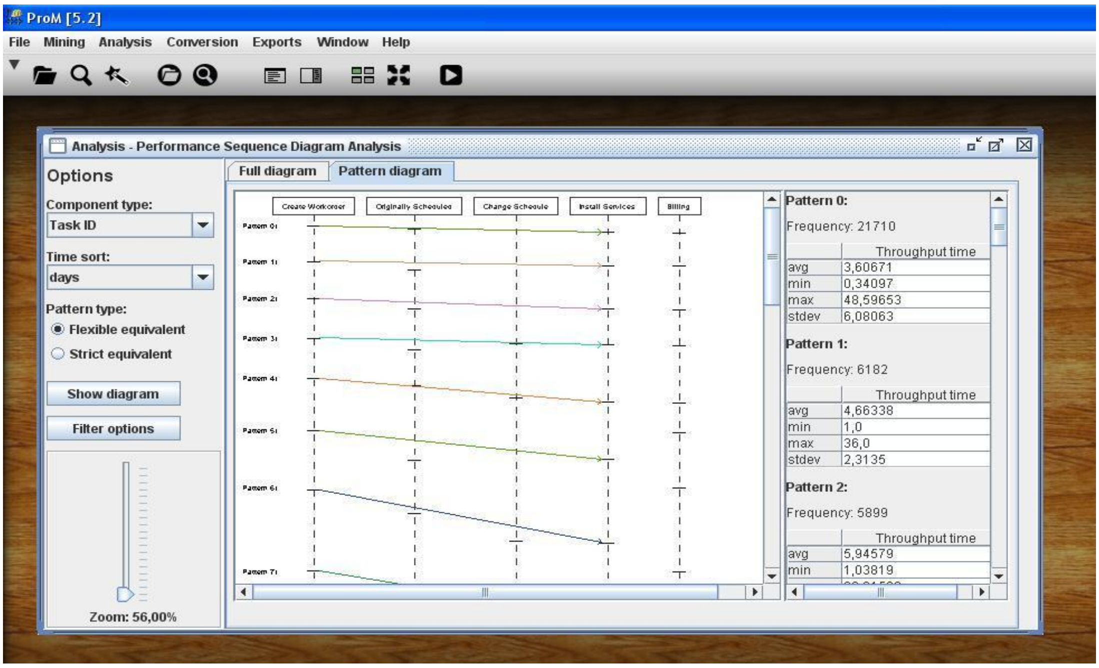


Figure 5.7 Performance sequence diagram

The performance sequence diagram also gives throughput time information, i.e. the time taken for completion of a case. In order not realistic long lasting cases to be left out, the log data is filtered. In the filtering option only process instances with throughput time below 250 days are set to appear. Thus 44 366 cases (out of 45 000) are going to be explored (as they were filtered in the previous section with Nitro).

Exactly as already discovered in Nitro, here it can be also seen that 25 different sequences take place within the process executions. The most usual one (pattern 0) appears in 21 710 cases (almost 50% of all the cases). It is followed by pattern 1 that can be detected in 6 182 cases.

It seems that in most of the cases, the average throughput time is around 3 days and a half. The next sequence usually takes around 4 days and a half. The average time of the third pattern (5899 cases) is 6 days. Table 5.8 presents the time data about the most appearing patterns and their *average*, *minimum* and *maximum* throughput times according to the performance sequence diagram analysis.

Table 5.9 Performance sequence diagram statistics

Pattern	**Frequency**	**Average**	**Minimum**	**Maximum**
Pattern 0	21 710	3.6	0.34	48.6
Pattern 1	6 182	4.7	1.0	36.0
Pattern 2	5 899	6.0	1.0	36.3
Pattern 3	4 670	6.2	2.0	65.0
Pattern 4	2 215	12.9	1.2	52
Pattern 5	980	15.4	1.0	30.0
Pattern 6	717	28.7	27.3	55.5

These seven patterns represent 95.5% of the cases examined. In 86.7 % of all of the instances, 4 main patterns appear (pattern 0 – pattern 3) with average time to complete a customer order of three and a half to six days. As shown in the table, some cases can be completed for a day and even less but there are also some situations in which the process takes two months to be executed. On the one hand, these are not necessarily non-typical cases because their continuation depends on the scheduled with the customer installation data. It is not impossible that a customer would like to schedule a date in two months. On the other hand the termination of a process in two months is quite worrying when the average throughput time is a few days. In a situation when the scheduling of an installation date in 2 months is not the case, the process has to be carefully examined as there may be some obstacles for it to be finished on time, e.g. unavailability of technicians to take care of all the coming orders, delay of the installation registration or of the billing task, late registration of the creation of a work order (when it appears as a last activity of a case) or late registration of an installation task. In cases like these, a responsible person can be appointed whose responsibility would be to control the process flow, e.g. if a technician is sent for an installation and during the whole day he does not report a successful fixing of the service, this should be indicated somewhere in the system as a signal that something is wrong. Thus the running process will become clear and the process data will not indicate such deviations as the above mentioned. Another example is the enormous system traffic and non-possibility for the information system to work properly because of appearance of bottlenecks through the process and not availability of the information system to reflect all the tasks completed in reality on time.

The observation of all of the patterns and their throughput times indicated in the performance sequence diagram shows that none of all the cases in the different patterns lasts more than two months (including minimum, maximum and average throughput time). It also shows that no case lasts less than 8 hours (0.34 days). So the overall range of the throughput times for all the cases is 8 hours to 65 days.

The performance sequence diagram also gives the opportunity for exploration of the resources and other classifiers. However the originators are discussed more detailed with the help of other plug-ins in the next phases – *performance analysis* and *role analysis*.

5.4.3 Conformance analysis

The conformance analysis serves for the check of conformance between the event log and a predefined process model. In this phase the filtered event log will be compared to the process model in Cox Communications shown in the beginning of chapter 5 on figure 5.1. The results are presented on Figure 5.8 below. They show 21 709 cases in compliance with the process model, containing 4 events i.e. "*create workorder – originally scheduled – install services – billing*". The other trace corresponding to the model, containing 5 events, is "*create workorder – originally scheduled – change schedule – install services – billing*". It appears in 2 215 process instances. So the overall conformity between the log and the predefined model is 53.92 % (23 924 cases).

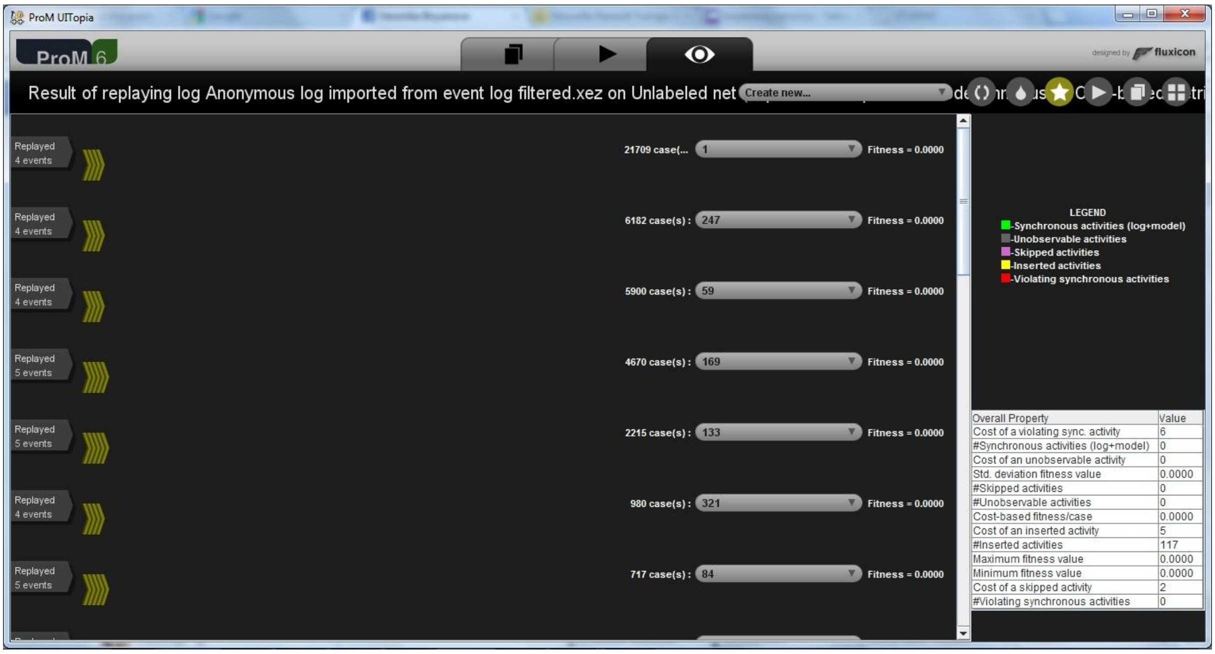


Figure 5.8 Replay an event log on a Petri net for conformance analysis

5.5 Performance analysis

This is the most important phase of the process diagnostics method considering the purpose of this master thesis. Four plug-ins are used to analyze the performance of the work order process in Cox: the basic performance analysis, the dotted chart analysis, the basic log statistics and the LTL-checker. In this step bottlenecks and problems are aimed to be discovered and analyzed.

5.5.1 Basic performance analysis

The basic performance analysis plug-in gives opportunity many different aspects of the process to be analyzed depending on the configuration that is made in advance and the graphs selected to be shown. The two main components “task” and “originator” are examined from different angles. When analyzing the event log of the work order process in Cox and setting different measures and graphs, a few insights come out.

5.5.1.1 Bar charts

Looking at the bar chart diagram on figure 5.9 the two dimensions (task and originator) are presented. On the horizontal *x* axis, the different activities in the process are shown and on the vertical *y* axis the time unit (specified to be days) is pointed. The dimension is set to be 3D for better visualization. The colored vertical lines on the chart represent the different resources involved in the execution of the different tasks. According to the bar chart, the most number of originators is involved in the change schedule task execution. The “originally scheduled” activity is also performed by a lot of different resources. Then the installation of the services and the billing take place with lower but still large number of participants and finally the “create work order” task seems to be executed by significantly lower number of people compared to all the other activities. This is a little bit strange as the creation of a work order is normally the first task to take place and should be instantly performed by a person in the company. The small amount of resources can hold up the process initiation and respectively the whole process performance after. The reason for the comparatively small amount of originators of this task might be that only some people are authorized to do this as it is important activity, being a first step of the process. The creation of a work order also includes giving the order a unique identification number that is also given to the customer further and always related to this very customer and exactly this order. So if a mistake is made concerning the relation between order, unique number and a customer, a big problem can occur, e.g. sending technicians on the wrong date or at the wrong place, etc. This demands high attention when an order is created and this might be the reason for the authorization of less number of resources regarding the initiation of this step. All these assumptions can be also related to the previously made observations where it was discovered that there are cases in which the process does not start with the creation of a work order as it was preliminarily defined according to the work order process model. This non typical sequence may be caused by occurrence of situations in which the creation of a work order cannot take place as a first step of the process and has to be postponed because all of the people authorized to do this are busy or cannot react exactly at this moment. However, the service has to be registered in order not to be missed and this might be the reason someone to have executed another step in the process first to be sure that it exists in the system and the customer request is not going to be forgotten or missed after this.

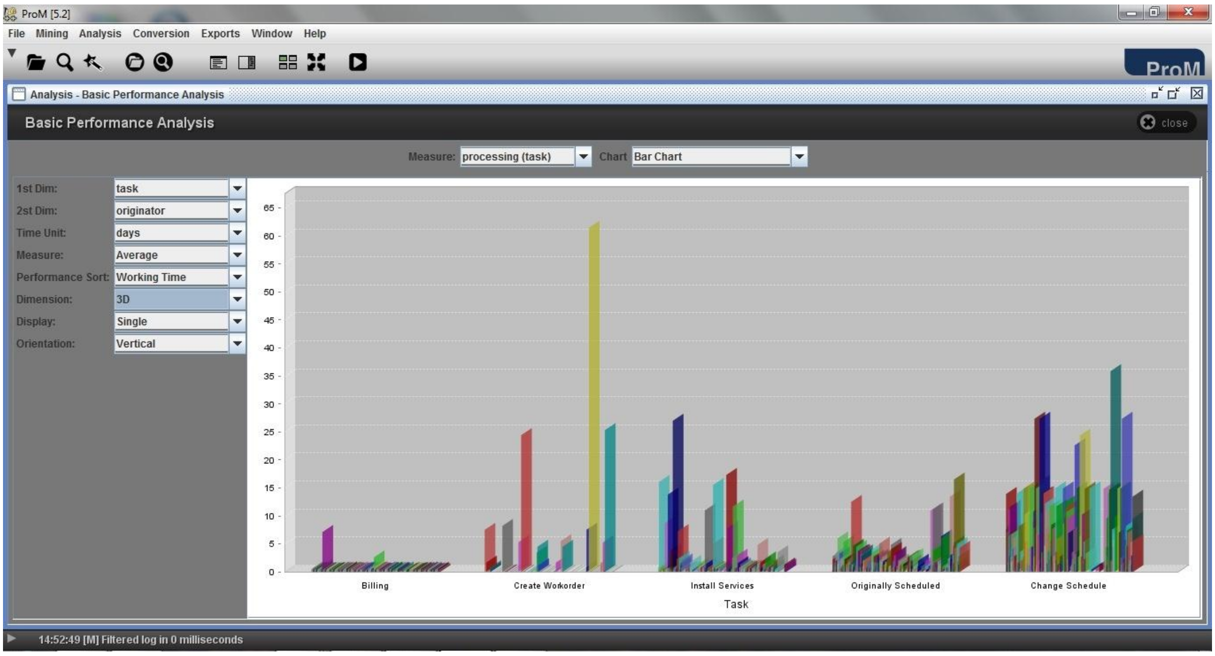


Figure 5.9 Bar charts of the basic performance analysis plug-in

Observing the time measures of the chart, it is obvious that the billing activity takes the least time to be completed. This is probably a consequence of the billing application, integrated in the Work Order Management System (the platform used for the process flow support). It is visible that the billing do not take more than a day when executed by 99 % of the people. The “originally scheduled” task mostly takes one to five days. The installation of services is varying (0-25 days) because it is dependent on the arrangement between the customer and the company. Factors like availability of the customer and availability of a technician play sgnificant role here. Strange behavior represent the bar charts of the creation of a work order – there are some sources that executed the activity for 25 or 60 days which is probably some system mistake as this seems impossible. It can be also possible that an account of a person was left opened and he went to a holiday for example and until he came back the system was indicating the activity as not completed (in cases where the creation of an order appears on the last place).

5.5.1.2 Pie charts

If looking the event log from another perspective using the pie chart diagrams, a few conclusions can be reached. Every chart on figure 5.10 represents a different resource and the tasks included in his work. The different pie pieces show how the average time it takes for an activity to be executed (in days).

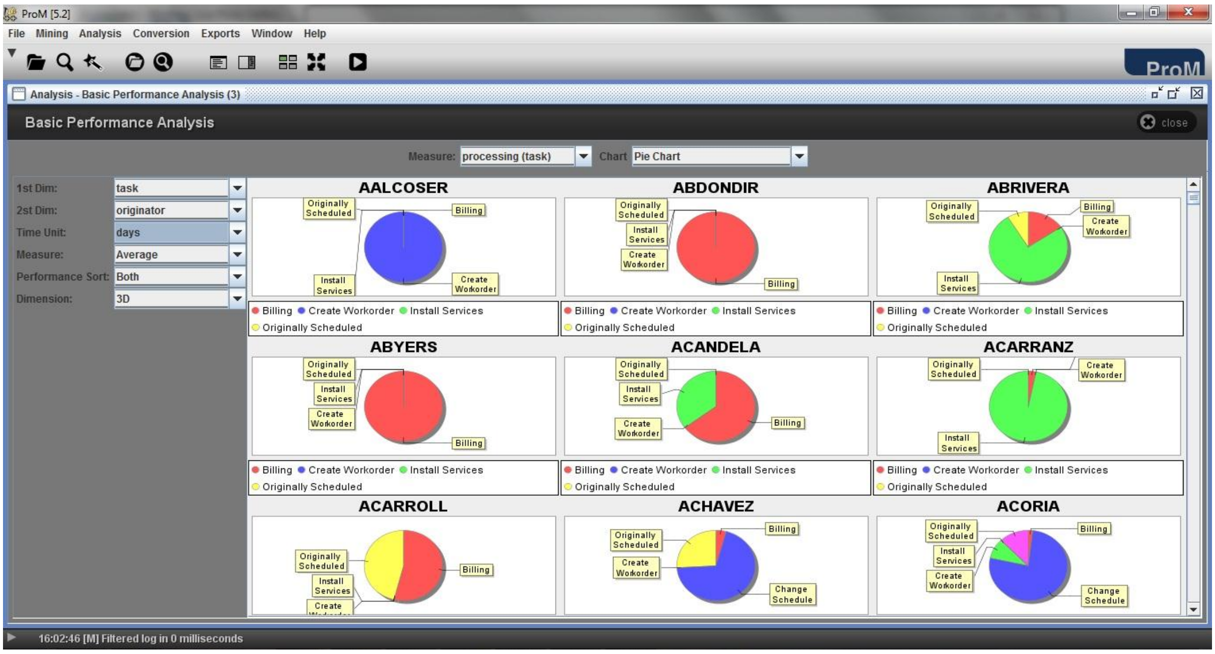


Figure 5.10 Bar charts of the basic performance analysis plug-in

In the work of some employees, the installation of services includes only and represents 100 % of the pie chart that represents their work. This is explicable as a technician is the only specialized person capable to do this and respectively this task takes all his time and has the highest average time possible for completion, i.e. 100 %.

Another activity that seems to take a lot of time for the people to execute is the creation of a work order. This is not typical as normally this should be the first step of the process but as it was already spotted before (control flow analysis), there are cases where this activity takes place in the middle or at the end of the process. In these cases it is possible that this activity represents the highest value of average time for some originators. The billing task also appears as taking the most time in some employees work but as a whole its execution time in the general picture seems to be quite short (1-5 % of most of the pie charts that it appears in).

The schedule and re-schedule activities appear as two halves or more than half a piece of a lot of pie charts. This points that a lot of people spend most of their time setting a date for installation and re-scheduling it if necessary. These could be complex and time-taking activities because probably a person has to find the unique order number, enter specific field, set all date, month, year and hour and maybe confirm. These are also tasks which are related to appointment of a technician and discussion with the customer so many participants are included and it is not surprising that they are the most time-taking for some people in Cox.

For many people in Cox the two activities of assigning installation date and billing appear as the most time-consuming. The possible reasons for the scheduling task were already mentioned above. Regarding the billing, it was already mentioned in the short case study explanation that there is a special application integrated in the software platform. It is possible that not a lot of people are trained to use it and for some of the employees it is harder than for other ones. In cases like this, the management has to monitor who are the people whose job includes the billing process and some training programs have to be organized for all of them, so that the time consumption of the billing activity will decrease its value.

It is not surprising that there are also many people whose time is equally spread on the 4 or 5 main activities of the work order process. Obviously these people are prepared to work with the software system and probably use it for a long time and manage to execute each task for a certain time that is neither too long not too short.

5.5.1.3 Meter charts

In order to check the average time that every activity takes in the work order process, the meter charts are going to be observed. The initial configuration settings take into consideration the working time of the company which is set to be 9.00-18.00h. The 1st and the 2nd dimension are set to be one and the same, i.e. “task”, the measure is “average” and the time unit is “hours”. This means that figure 5.11 shows the average amount of hours that are necessary for each activity to be executed. According to the charts the billing and the creation of a work order are the activities that need the least time to be performed. Their execution time appears in the “normal” green area of the meter which points stable performance of these tasks. The creation of a work order takes about 1.52 hours in average and the billing takes up around 2.5 hours to be completed. Then the “originally scheduled” task comes with an average execution time of around 17 hours and it also stays in the green area of the meter chart which indicates “normal” performance. The installation of servies takes 23 hours in average still staying in the green field. The most worrying task seems to be the change of schedule. Its average execution time (44 hours) appears in the red “critical” zone of the meter chart. This means this task indicates low performance and measures has to be taken in relation with it.

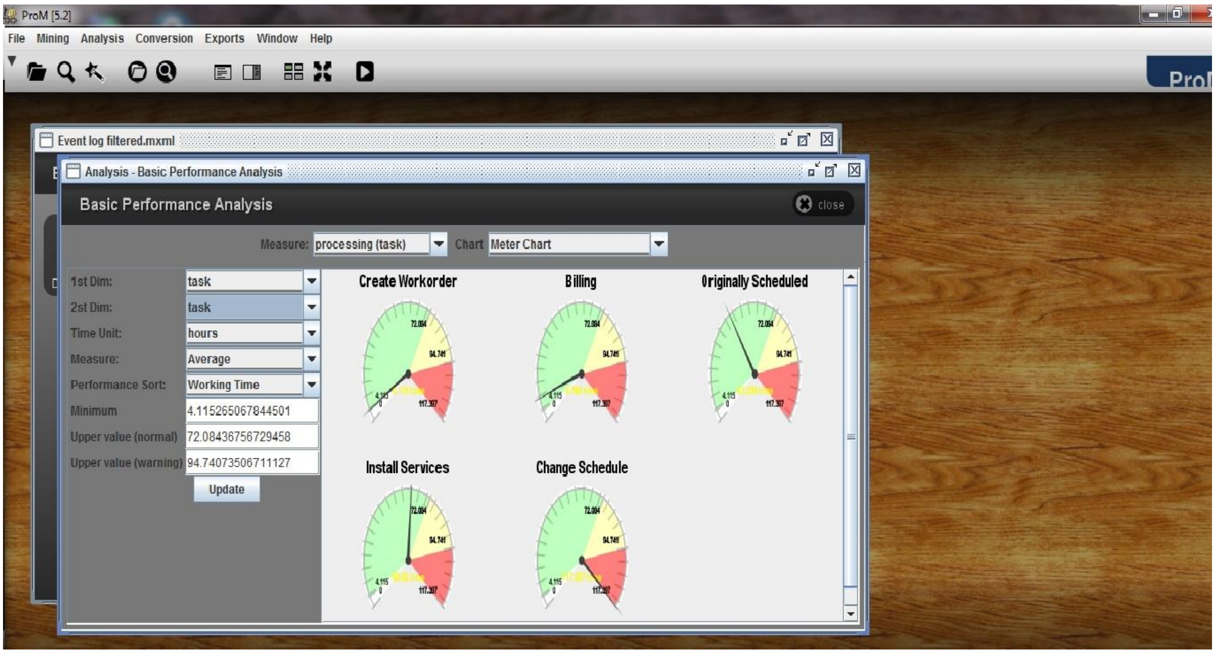


Figure 5.11 Meter charts of the basic performance analysis plug-in

In case a manager or a stakeholder from the company needs more detailed information concerning the performance of tasks of every single resource included in the process, the meter charts are interesting to be used again but setting the "originator" as a second dimension in the results.

5.5.1.4 Conclusion

The basic performance analysis plug-in includes a lot of ways to represent results about execution time of activities in the company as well as resource performance. In this study case it has given interesting insights for the work order process in Cox Communications identifying activities that take too long to be completed or resources that need too long to execute a task. After analysis of these weak points, corrective measures are to be taken by the management so that the process can be optimized leading to fast working rates and respectively more satisfied customers and employees.

5.5.2 Dotted chart analysis

The dotted chart diagram on figure 5.12 gives a helicopter view of the work order process. The options are set as follows: the component type is "instance ID" so cases are spread on the *y* axis, the time option is "actual" – it will be indicated on the *x* axis, set to be in weeks. The dots are colored according to the type of task: the red color corresponds to the *billing* activity, the orange – *change schedule*, green – *create work order*, purple – *install services* and blue – *originally scheduled*.

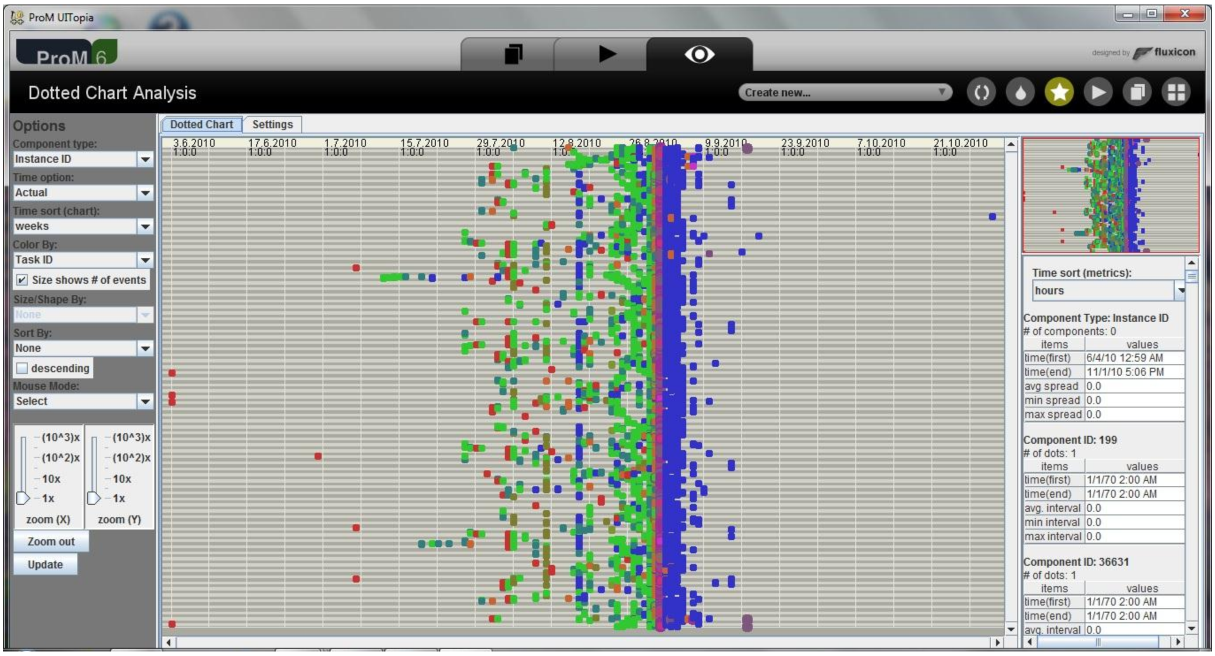


Figure 5.12 Dotted chart of the Cox Communication work order process

The throughput time of every case can be easily observed on the dotted chart. It seems to be various but in average a case takes up around 4-5 weeks to be completed (according to the graph). The sequence of activities in every case can be also detected. The cases tend to start differently and tend to end mostly with the "originally scheduled" task. The installation of services normally appears just before the end of the process in most cases and the creation of a work order is mainly in the beginning or the middle of the processes. These are the most basic conclusions that can be made by analyzing the graph when the instance ID is set to be the component type.

The dotted chart analysis can also focus on the originators. The component type can be set to be a resource or the dots can be set to be colored according to the resource, leaving the "Task ID" to be the component type (as in figure 5.13). As it was already discovered that there are 602 resources participating in the process flow, it is difficult to analyze the different work of every employee in details but some major comments on the relation "*task – resource"* can be made about this. On the graph the different activities appear on the rows, the time dimension is shown in weeks on the *x* axis and the dots represent different employees of the company executing the different activities. The size of the dots corresponds to the number of events completed. The bigger the dot, the more events this person has executed. According to the analysis of this graph, the installation of services is obviously performed by only a few people in the company. As already mentioned earlier, this is a normal phenomenon as this is a special technical work that can only be executed by a specialist. The billing, creation of a work order and assignment of an installation date seem to include a lot of different resources who are probably all administrative people trained to work with the software platform and execute all the other activities in the process. The "change schedule" task does not appear so often in the process instances as any other activity but also includes the participation of various performers. According to the size of the dots, representing the different employees and how many events they had executed, it seems that all of the activities were performed a lot of times by a lot of people but also some activities represent a small appearance in an employee work so there is a great variation in the relation between resources and type and number of events they execute.

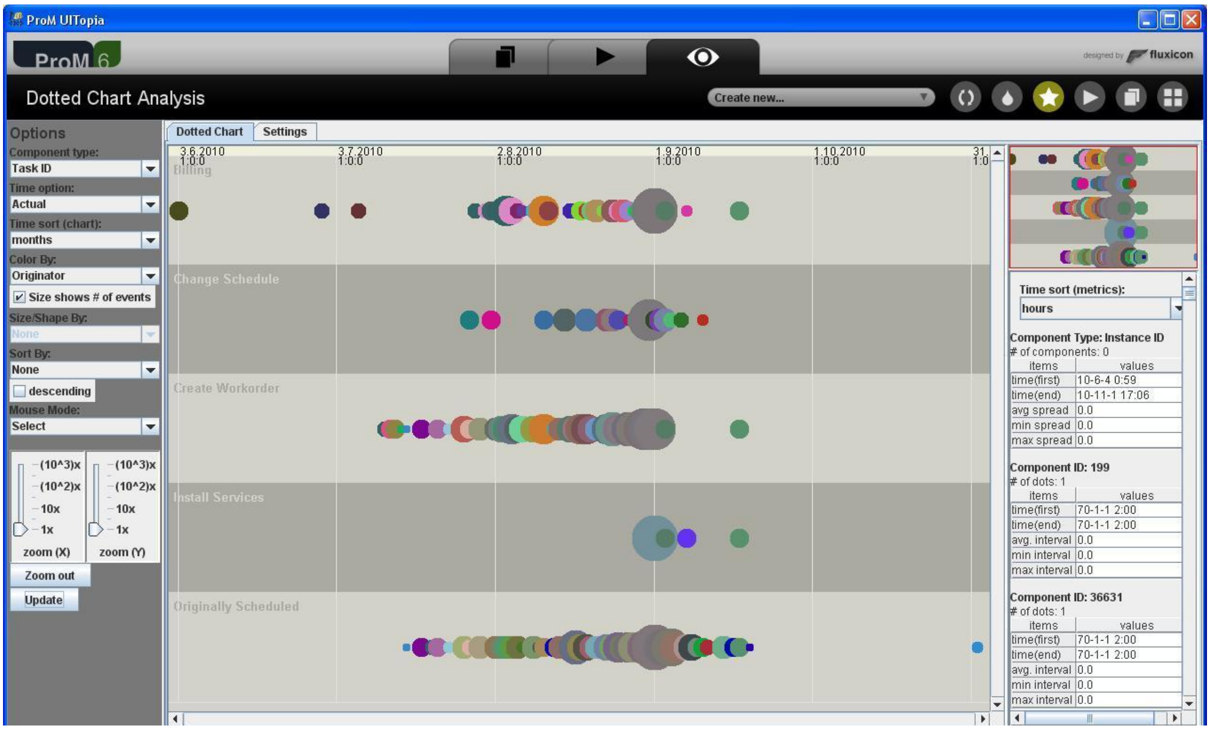


Figure 5.13 Dotted chart analysis (component – Task ID, colored by originators)

5.5.3 Basic log statistics

The basic log statistics plug-in resembles the Nitro statistics. It shows different statistical data such as overall duration of the event log recorded that is 3617 hours (150 days). It also reveals the duration of every single case. Picking up randomly around 50 different cases to check their duration, cases with throughput times of 27 days, 0 days (10, 17 hours) and 4 days appear. This indicates the variety of the time necessary to complete a process instance.

The basic log statistics also presents attribute statistics which are very useful for deeper analysis of the log. The frequency of the resources participation is pointed. The observation shows that this parameter varies from the value of three times through hundreds and reaching five to eight thousand times frequency regarding the different originators. People who had participated just a few times in the process are probably managers whose work is not this but in urgent situations they were the ones to execute an activity or some people from other departments that replace the responsible worker for a few times in case he cannot handle the request or some problem with his username and pass occurs. Frequencies with low values may also be a result of internships – young people come for one or a few months and then they leave. Sometimes they might be given access to the main software platform and be allowed to execute a step in the process. There might be also a possibility for someone to has left the company during the period observed in the log. The opposite is also possible – a new employee could have been appointed and to have used the system only a few times.

The frequency of activities that was already discovered in Nitro can also be obtained. After the filtration done in the control flow phase of the process diagnostics method, the frequency of activities is as shown in table 5.10 below:

Table 5.10 Frequency of activities in Cox Communication (filtered event log)

Value	**Frequency**
Create work order	44 366
Billing	44 366
Originally scheduled	44 366
Change schedule	9 077
Install services	44 366

These values confirm the conclusion made at the beginning that every case includes the main activities that should appear in the process and almost 1/5 of all the cases also includes the change of schedule for installation. This is an evidence for the occurrence of every task of the process in all the cases no matter their sequence which was discovered to be various.

The “monthly dollars” attribute statistics of the event log granted from the Cox company show enormous variety in prices which can be slightly related to the performance of the process. In some cases the customer might ask an express execution of the process (e.g. within a day) and this will be probably charged higher than a normal order. Sometimes very high prices like 600 dollars and more appear which may be a result of prepayment by the customer which was already assumed earlier in the analysis. In a lot of cases a null value appear as a price which can be interpreted in a few ways – the order request may be wrong detected by the system and finally calculate 0 value because of an error; process steps may be wrong recorded and followed by a resource, making it impossible for the billing application to find a base for calculation of a price – here a reason for low process performance has to be searched for. The reason for appearance of 0 values for monthly rates also can be a result of discounts, promotions or just prepaid services for a few months or more.

5.5.4 LTL checker

With the help of the LTL checker plug-in different properties of the log and different rules (e.g. sequence of certain activities) can be checked if they hold for the process. For instance, it can be discovered in how many cases the right sequence of the activities one after another appears in the process run.

Analyzing the event log available from Cox Communications with the LTL checker, some different analysis will be performed compared to the one until now. Different aspects of some cases will be observed and analyzed.

There are a lot of interesting rules to be checked if right for the process. Here the "*exists_activity_done_two_times*" and "*eventually_activity_A_next_B*" rules will be used.

After the check if any activity occurs two times in the work order process, the event log shows that in any of the 44 366 cases analyzed there is not even one that contains same activity executed twice. This indicates that there are not duplicated tasks and respectively illustrates good system and resource performance.

The exercising of the next rule is made for the two schedule tasks. Through the "*eventually_activity_A_next_B*" rule some impossible sequence is checked , i.e. the execution of the "change schedule" task before the "originally scheduled" installation date. This should not be possible as a date first has to be fixed so that it be rescheduled after. However, the event log shows the appearance of this order of the two activities in 294 cases. This indicates low performance of the process flow in these cases and a reason has to be searched for this non-typical behavior. Fortunately the portion of this strange behavior is very small (under 1 %) and can be considered unrepresentative for the whole event log.

The same rule is used to follow the sequence *billing – creation of work order.* It reveals that there are 1 116 cases where the billing task comes before the creation of an order (see figure 5.14). In every of these cases these are the first two events that appear. So in 1 166 cases the start is billing and just after this the customer order is created in the system. This is very low percentage (under 1 %) of the event log and should not be considered a big problem but still indicates problem areas of the process flow. As already assumed, this can be a result of prepayment option or may also be an effect of problems with the software.

Another possibility is the billing to have taken place before the creation of an order because the billing application has to be run at certain periods (not every day) and in order not to miss the billing of this service, the request was input in the system with some special ID number that the billing application can detect and after this, an ordinary request number could have been created that indicates a creation of work order in the system.

When the same rule is checked for the sequences *install services – create workorder* and *originally scheduled – create workorder*, no deviations are found. There is not even one case that includes this order and this is sufficient for the good performance of the process.

Figure 5.14 LTL checker plug-in results

5.6 Role analysis

During the role analysis diagnostics phase of the Cox Communications work order process, two main performance plug-ins are used – the social network miner and the role hierarchy miner. In this final step of the process diagnostics, the focus is on the people that work in the organization and their relationships and dependencies. The analysis is hard because of the great amount of resources that take place in the process but still some major comments are made concerning this aspect of the activity chain.

5.6.1 Social network miner

The *work together social networking* plug-in used for the social analysis of the Cox Communications event log data indicates that some resources exist that do not have any relation with any other originator in the company. These are all the resources (circles) positioned in the left half (and not only) of figure 5.15 that have no edges connected to other resources (e.g. *PLSSHNFI, GUYEE, KOEERGFE*, etc.). This mean the people represented by these circles are executing some activities totally independent from any other participant in the work order. An explanation of this phenomenon can be the existence of external employees in the company that execute tasks independently. Another explanation might be the executing of a whole work order process just by one person. In such situation it is not possible the system to make any relation between such an employee and any other worker in the company. Thus, no relationship is detected in both ways (no edge), e.g. neither this originator transfers the process continuation to some other nor does he receive a request for execution of a next step in the process by another resource. It was already discovered that there are some people executing all the activities in the work order process. The *working together* diagram implies that in some situations this probably happens within a single process instance, i.e. the person executes all the activities in a single case. There is also a possibility that the information system does not detect right the flow of activities between some resources and they appear to work alone in the activity chain.

However, it is obvious from the picture that enormous part of the working staff in Cox is in a very complex relationship network, including connections both incoming and outgoing, between hundreds of resources. This might lower the performance of the work order process because the process thread and sequence of work can be easily lost somewhere in all the channels that it flows through. A possible solution for increasing of the performance is formation of teams that work together on the process only within the group and do not collaborate with other teams that do the same but separately.

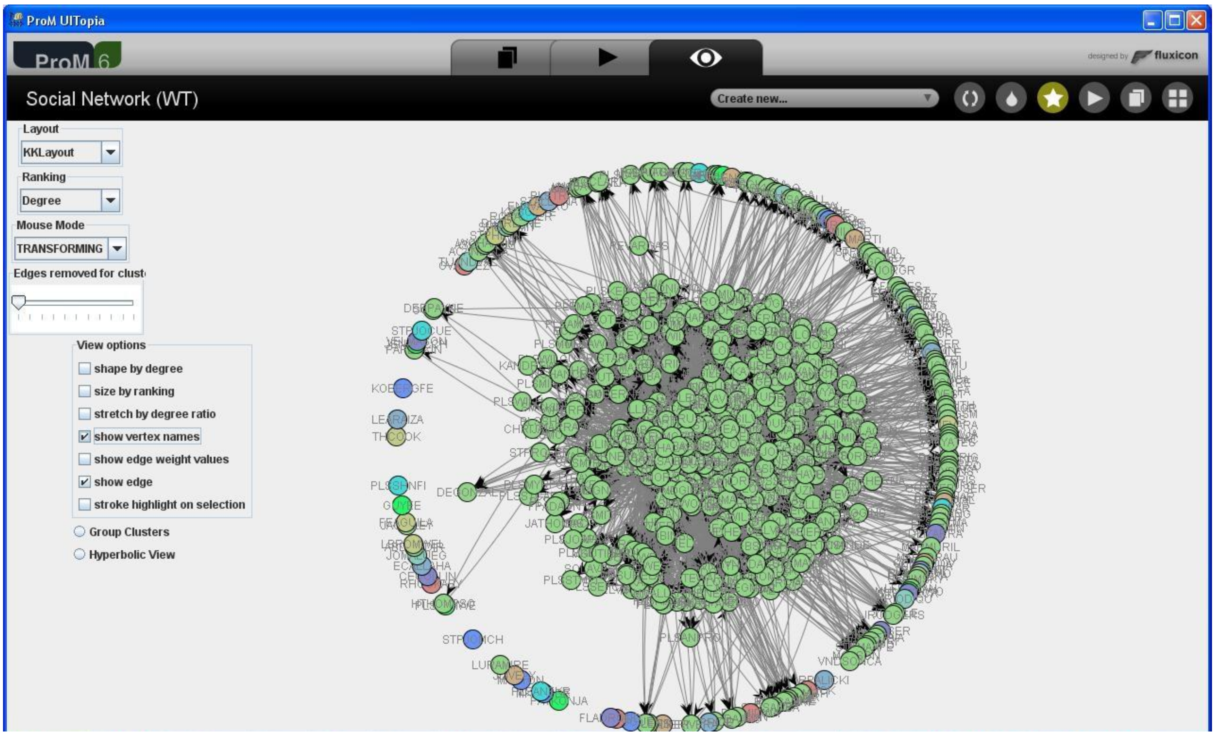


Figure 5.15 Work together social network

The social network is also observed from another perspective, i.e. *subcontracting*. The main idea is to follow how many times an employee executes an activity in-between two activities performed by another individual. This might indicate that the work is subcontracted from one worker to another. Figure 5.16 presents the complex structure of subcontracting in Cox Communications contained in the filtered event log data.

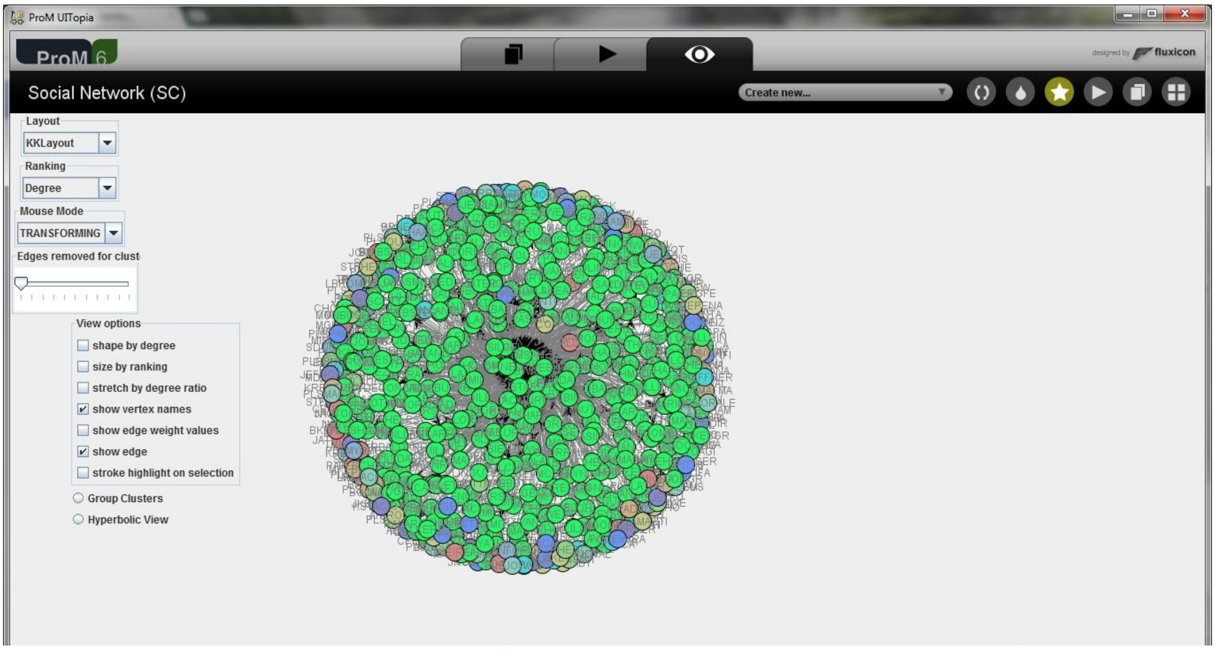


Figure 5.16 Subcontracting social network

The existence of 602 resources in the work order process in the company explains the complex network of sub-contracting. The circles on the periphery of the diagram are obviously less involved in the subcontracting and the more they go the center of the whole network, the more complex the relationships become and the more subcontracting there is. The weights of the relationships between these resources are higher than those appearing on the outer part. As a whole the analysis of the figure concludes that there are a lot of originators involved in a subcontracting which is dangerous for the high performance of the process but if executed successful, it shows very good performance and stable organizational process resource relations and respectively process flow.

5.6.2 Role hierarchy

The role hierarchy miner is the last plug-in to be used in the Cox Communication work order process analysis. It focuses on the organizational mining and the resources in the company. The results of the mining are shown on figure 5.17.

The hierarchy model shows how many resources participate in the execution of every activity and the table below the model indicates how many times each resource has executed any type of activity that is included in the work order process in the enterprise. The relationships between the originators and the work flow between them is also indicated by the arcs on the figure. Clicking on every element of the model, the relationship *originators – number of tasks executed* is shown in the table that appears under the model. For instance when clicking on the *billing – originally scheduled – create work order* balloon, all the people that have executed the three activities the same amount of times are shown in the table. The maximum performance of these three is 956 times (by *BHANDLEY*).

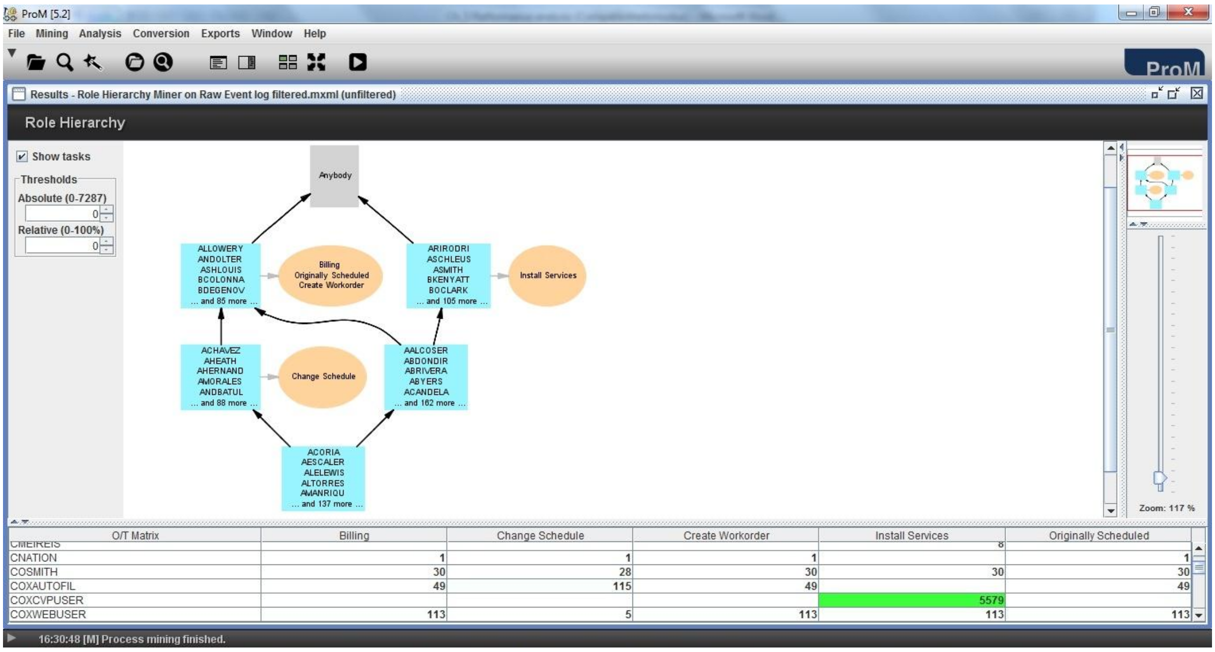


Figure 5.17 Role hierarchy miner results for Cox Communication

The analysis of the log reveals participation of resources of exactly the same amount for every activity that has to be executed in the process. This are probably the already discovered in the previous section independent working employees that were assumed to complete a case alone from its start to its end (e.g. AALCOSER, ABDONDIR, ABYERS, etc.). Some of them are listed in table 5.11 below.

Table 5.11 Number of tasks executed by some originators in the work order process in Cox Communication

O/T matrix	***Billing***	***Change schedule***	***Create workorder***	***Install services***	***Originally scheduled***
AALCOSER	12	0	12	12	12
ABDONDIR	2	0	2	2	2
ABRIVERA	1	0	1	**22**	1
ABYERS	216	0	216	216	216
ACANDELA	11	0	11	**17**	11
ACARRANZ	51	0	51	**242**	51

Some of the resources in the table tend to execute the installation of services more than any other task in the process (e.g. *ABRIVERA*, *ACARRANZ*) . However, it seems that they are authorized to also perform any other step of the activity chain.

In the table below the model, large numbers of execution of a specific activity are highlighted (see figure 5.17). The biggest amount of tasks executed is also pointed in the *thresholds* field which indicates the performance of 7287 times installation of services by an employee (*ETSCSWAP*). Another example is the originator *COXCVPUSER* who has also exclusively performed only the "install services" task 5579 times. *CTMSAN* did it 6054 times but also executed dozens or hundreds of times of all the other activities. These statistics indicate high performance of these resources and enormous work load. Taking the *COXCVPUSER* example and having in mind the working period reflected by the event log, the work load of this employee can be calculated in order to follow his productivity. The analyzed period of the work order process is around 3 months. This originator is registered 5579 times to install services. Leaving out the weekends, there are 22 working days in every month. So for 66 working days this employee executed the “install services” activity 5579 times which means 84 times in average per day. It is obvious from the table that there are a lot of resources that performed only a few activities just a few times. Their productivity and performance are enormously low compared to the resource analyzed and reasons for this have to be searched for by the management if the executives want to increase the process performance of their employees. There might be some explanations (as assumed before) such as absence of some resources in part of the time frame analyzed or internship programs and participation of some people just a few times in a process, external participants in the process, management included in the process performance by exception, etc. However, if low performance is detected, additional intercompany research has to be performed to reveal why some people show very high work load service and others tend to execute just a few tasks for months.

Chapter 6 – Conclusions

6.1 General conclusions

In the daily working activity business organizations need to keep a competitive advantage on the market. They need to maintain stable processes, operate fast and leave out bottlenecks and problems in the everyday activities. Customer satisfaction and fast working rates are one of the main goals in order to keep the business alive.

Managers and stakeholders predefine processes across the organization to be able to keep the process stick to workflow models. Unfortunately, what happens in reality is rarely the same as what is constructed in models in advance. Process mining is a relatively young research area that may contribute to the optimization of processes in an enterprise and help elimination of bottlenecks, detection of possible problems and development of activities. Process mining offers different techniques for digging into processes and analyzing the results which come out of the mining. Processes can be mined in three different perspectives: process perspective, organizational perspective and case perspective. Three types of process mining exist: process discovery conformance checking and enhancement. The subject of this master thesis, i.e. performance analysis, can be mostly categorized under enhancement. The enhancement includes the adding of different perspectives to the process activities such as time and resource (person who has executed the task) for instance.

A special framework was created in the Technical University of Eindhoven to support the process mining techniques. This is the **ProM** open source software that contains different plug-ins related to the process mining methods. The input necessary for the framework is a so-called event log which represents data extracted from the information systems in a company that includes information about activities of a process, timestamps, resources and some additional attributes added to the different process instances. After importing an event log in ProM, various analyses can be performed concerning the process reflected by the log. The software includes mining, import, export, conversion and analysis plug-ins.

Nitro is a commercial tool that also supports the field of process mining and its main goal to extract meaningful process information from event logs in order to improve the processes. Nitro presents some major statistics such as the number of events and cases included in the log, information about the time period (start and end date) of the process observed and activities included in the process. The explorer function of Nitro gives insights to the different cases variants, i.e. the sequence of the various tasks in the different cases that appear. Throughput times of tasks and cases can be identified as well. Finally, Nitro focuses on every attribute, included in the log – timestamps, resources, cost-related information, etc. Some summaries for any attribute are given on the base of which different conclusions can be made.

There are some **obstacles and challenges related to process mining** such as the occurrence of incomplete cases, missing cases, duplicate or missing tasks. Basic problem is also the scattered and unstructured data which is extracted from the information systems in an organization and often does not reflect reality accurately. Overfitting (representation of too specific process model) and underfitting (construction of too general model) are also among the problems related to process mining. Other weak points are noise (content of rare behavior not representative for the process flow) and incompleteness (having too few events) in an event log.

The **process performance analysis** helps organizations to measure and analyze the performance of their business processes. A definition of performance that this thesis sticks to is 'the degree to which and organization carries its objectives into effect'. It is considered to fit best the purpose of the current research and the context of this project. The so-called key performance indicators serve for measuring the performance of an enterprise. These are different for the different companies and can be time, cost or quality related.

Performance of business processes can be analyzed with the help of process mining. Different conclusions can be reached concerning the participants in a process, their roles, activity characteristics such as which activities tend to fail most or take the most time, etc. Process mining techniques is powerful to control the processes of companies, revealing various relations between activities, resources and time dimensions. Through the use of process mining the event logs recorded by information systems concerning particular processes, can be analyzed in details and further information for the performance can be discovered. Questions regarding activities, cases, social networks and bottlenecks can be answered and problems tried to be solved.

Process diagnostics is a methodology which can be followed to analyze a process with process mining. It includes five steps: log preparation, log inspection, control flow analysis, performance analysis and role analysis. The first phase, **log preparation,** is a compulsory requirement for the process mining to start. The information appearing in information systems has to be gathered together, filtered if necessary and structured so that it can be converted to an event log as this is the input for the framework to be used after. The log can be created with the help of converting tools,. In this thesis Nitro is used to create an event log. After the conversion of the data to a .mxml or .xes format which can both be used as input for the ProM framework, the phase of **log inspection** can take place. Through the use of Nitro and some ProM plug-ins (e.g. fuzzy miner), basic statistics of the log can be observed. These are start and end dates of the log, events, cases and activities included in it, resources that execute the activities, etc. In this way, a general overview of the process can be given.

In the **control flow analysis**,the accent is on the ordering of the activities within the process examined. As mentioned before it is rarely the case that the process runs as predefined in a process model. In this phase, the right sequence of tasks is meant to be discovered as well as the conformance between the process and the log. In this step most frequent traces in a log are also observed and analyzed. The performance sequence diagram, replaying a log on a Petri net in ProM and Nitro can be used for the control flow analysis.

The most important step for the subject of this thesis is the **performance analysis phase**. The ProM framework has various plug-ins which can be applied for performance information to be extracted and analyzed. In this stage, bottlenecks and problems can be identified, throughput times of activities and cases to be discovered and analyzed and resources to be examined. Possible plug-ins to be used are basic performance analysis, dotted chart analysis, basic log statistics and LTL checker.

The final phase of the process diagnostics **– role analysis –** mostly focuses on the resources in a process, i.e the people involved in the execution of the different tasks in the work flow. The originators form social networks and the hierarchy in a company can often be discovered through the use of process mining algorithms. Social network miner and role hierarchy miner are possible analysis plug-ins to be used for observing the resource behavior, constructing social nets and revealing of dependencies between stakeholders in an enterprise.

6.2 Case study conclusions

The process diagnostics methodology shortly explained above serves for a framework of steps to be followed during the case study performed in this master thesis. The work order process in Cox Communications company (USA) concerning its residential services is analyzed with the help of all the mentioned algorithms and plug-ins. Thus, process behavior is discovered as well as problem areas and some assumptions are made regarding the work flow in the company. This case study is used to discover what the performance analysis capacities of process mining are and which plug-ins are useful for this purpose.

In the first step, **log preparation** , an event log is created with Nitro. A .csv file is granted from the organization and turned both into .mxml (ProM 5.2) and .xes file (Prom 6.1)..

The **log inspection** gives an overview and reveals the main activities included in the work order process. These are ‘create work order’, ‘originally scheduled’, ‘change schedule’, ‘install services’ and ‘billing’. According to the predefined process model, this is the sequence in which these activities have to appear. The log consists of 189 690 events included in 45 000 cases. Every event represents a task executed and includes the originator, the timestamp, the work order number, the monthly fee and a few more attributes. Every case refers to a customer request for installation of services. Observing the start and end date of the event log, an error is indicated. The end date of the log appears in the future (year 2018) which means there is a mistake in the recording of some case(s). For this reason time filtering is executed in Nitro so that the event log includes the correct time period. After the filtering, the number of events and cases decrease respectively to 186 541 and 44 336. The number of resources stays the same as before, i.e 602. The period observed is approximately 5 months (start date: 04/06/2010; end date 01/11/2010). However, the highest density of events appears for a period of 2-3 months.

The observations in Nitro show that all of the cases executed include the main four activities that have to be completed within the process, i.e. ‘create work order’, ‘originally scheduled’, ‘install services’ and ‘billing’. Almost ¼ of the cases indicate the appearance of the fifth (possible but not compulsory) task – ‘change schedule’. This indicates right business activity and supposes completion of all the cases. However, there are a lot of cases including strange issues concerning the sequence of the tasks in the chain and involvement of the employees in the working process and these have to be further analyzed. In the case study some assumptions are made concerning this.

The price value for the different customers varies a lot and it is strange that in many cases this value is 0. Again some assumptions are made concerning this, e.g. loyal customer discounts, information system problem, prepayments, etc.

Nitro shows the different variants, i.e. sequences of tasks, which appear in all the cases recorded. After the log filtration there are 25 different traces. The exploration of the log through Nitro also shows that the mean throughput time for a case to be completed normally takes 1-4 hours. In situation of longer-lasting cases, possible reasons have to be searched for in order the delay to be explained.

Regarding the originators in the work order process in Cox, it seems that 13 people mostly take place in the activity chain. Their frequency is 1-5 %. All the others represent frequency of work less than 1 %.

In the log inspection phase, the real behavior of the process is discovered with the help of the fuzzy miner plug-in in ProM. The fuzzy model analysis gives insights for the process flow and supports the conclusions made with the Nitro tool. At first the model looks right, the expected sequence of the activities according to the predefined process model appears on the fuzzy model but some strange sequences are also discovered that do not conform the work order process model. These need further analysis and attention.

In the **control flow phase** related to the Cox event log data, the most frequent process behavior is presented. The ‘explore’ feature in Nitro, the performance sequence diagram in ProM and ‘replay an event log on a Petri net’ plug-in show the most frequent variants in the event log. There are four main traces that represent 86.69 % of the whole log. The most often variant takes place in approximately 50 % of the cases. The trace it follows (*create work order – originally scheduled – install services – billing*) is in conformity to the process model and shows correct behavior of the process in reality.

There are a few more main variants that mostly appear in the process but the ordering of activities differs from this in the process model. Possible reasons for this might be wrong recording of events by the information system in the company, negligence of employees leading to late registering of activities in a case or just change of order of tasks because of impossibility to be executed in the predefined way. The observation of the process helps the management to monitor the performance of the tasks and take measures to improve the activity chain and adapt it to the internal and external environment. The aim is fulfillment of the main objectives of the company, following its strategy and operating in the right way that will keep the business alive and moving in the right direction.

The conformance analysis of the process shows 53.92 % (23 924 cases) conformity between the event log and the process model. This is not a positive indication for the work order process in Cox as only half of the period observed indicates correct completion of the process instances. For all the other cases, possible reasons for the deviations have to be searched for, analyzed and be worked on.

The **performance analysis** phase is the most important regarding the purpose of this master thesis. The analysis of performance of the activity chain in the company gives a few insights about the work flow.

First, the basic performance analysis is executed to reflect the performance of the process. According to the bar charts included in this plug-in the most number of originators is involved in the ‘change schedule’ task. The ‘originally scheduled’ activity is also included in the work of a lot of people. Then the installation of services and the billing take place and the creation of a work order seems to involve the lowest number of originators. This might be a result of limited authorization for people to execute this task as it is normally the start of the process and respectively has high importance. . The hierarchy is illustrated on figure 6.1.

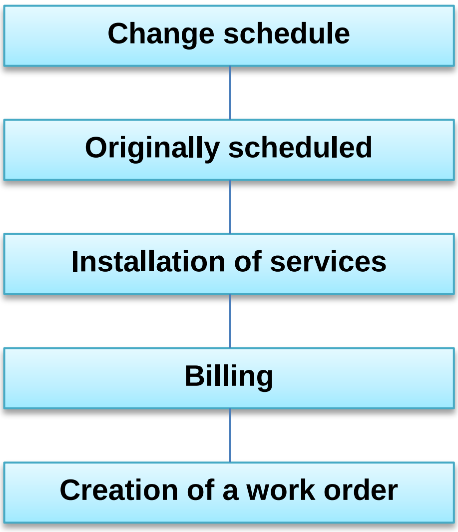


Figure 6.1 Task hierarchy according to the involvement of resources in execution of activity (from highest to lowest number of originators involved)

The pie charts of the same plug-in reflect the throughput time of every activity for every person in the enterprise. Some employees execute only one task, e.g. 'install services' and this takes all of their working times. The throughput times differ for every activity according to the different originators. This can be related to the training of the people to execute different tasks and the level to which they are used to operate with the system.

Meter charts are used to calculate the average throughput time for every step in the process. The results indicate that creation of a work order and billing take the least time and the change of schedule takes the most. The latter appears in the so called critical red zone of the meter chart indicating that measures have to be taken for the throughput time of this task to decrease.

The dotted chart analysis gives a helicopter view of the process and can help for the analysis of throughput times of each case included in the process for the period observed. The focus may also be on resources. A few conclusions can be made as a result of the dotted chart analysis:

- In average a case needs 4-5 weeks to be completed;
- Different cases start with different activities but mostly tend to end with the ‘originally scheduled’ task;
- The installation of services is performed by small amount of people in the company;
- The creation of a work order, the ‘originally scheduled’ activity and the billing include a lot of different originators;

As a whole there is great variation in the relation ‘*resource – event’.* People execute different types and combinations of events as well as various amounts of tasks.

The basic log statistics plug-in gives some major statistics about the process, e.g. overall duration of the event log – 3617 hours (150 days). It also shows the different throughput times for each case. A case in Cox Communications can take from 0 to 30 or even more days to be completed. The frequency of the participation of people in the process is also shown. It indicates values from three to thousands of times execution of an activity by one person.

The LTL-checker in the ProM framework helps for the investigation if certain rules are followed in a process. Applied to the Cox Communications event log it is revealed for instance that there are not any duplicated tasks (executed twice) in the whole work order process event log. This indicates good performance of people and technology. Another rule checked in the case study is the “eventually_activity_A_next_B” rule. It is applied to a few sequences of activities. The most non-typical result is the appearance of changing of a scheduled date for installation of services before the assignment of such a date. Fortunately this sequence is detected only in 294 cases (lower than 1 % of the whole event log) and this behavior is not considered representative for the process.

The last phase of the process diagnostics methodology used in the case study, i.e. the **role analysis** reveals information for the resources in a process. Social networks are drawn and relations and dependencies between people are investigated.

The social network miner and the role hierarchy analysis used in this final step of the process diagnostics draw a few conclusions for the work order process in Cox. Resources that do not have any relation with any other person in the company are found. Possible reason can be the involvement of external employees in the work order process or just execution of the whole process from its beginning to its end handled just by one independent originator. It was already discovered that there are people whose work includes performing of all the tasks in the process.

In general, the enormous number of employees detected by the event log to generate activity (602 people) leads to complex relationship social network. It includes both incoming and outgoing connections among hundreds of resources.

The role hierarchy miner shows the frequency of different tasks included in the work of each employee. Some of the people participate a few times in a certain activity for all the period analyzed and others initiate thousands of tasks for the same time tract. The productivity of every resource can be measured according to the data projected by the hierarchy miner and assessment of the performance of all the people in Cox can be obtained. The appearance of some originators less than others is a result of reasons which have to be additionally discovered if the management wants to focus on this aspect of the process, i.e. the resource performance and productivity.

6.3 Main Conclusion

The process mining algorithms and the use of different plug-ins available in the ProM framework to analyze the work order process in Cox Communications, leads to sufficient insights and assumptions which reveal the performance of this main process of the company. Based on the literature study and the case study held in this master thesis conclusions are made concerning the relation of tasks and resources, dependencies between employees, throughput times of different process tasks and cases and the overall performance of the process. The added value of the different tools and their plug-ins used in the different phases of the process diagnostics are presented in Table 6.1. It projects the significance of the various plug-ins for every stage of the diagnostics.

Table 6.1 Added value of Nitro and different process mining plug-ins in each pocess diagnostics phase

Phase/Plug-in	***Log preparation***	***Log inspection***	***Control-flow analysis***	***Performance analysis***	***Role analysis***
Nitro	**++**	**++**	**++**	**+-**	**+-**
Fuzzy miner	**--**	**++**	**+-**	**--**	**--**
Performance sequence analysis	**--**	**--**	**++**	**+-**	**--**
Conformance analysis	**--**	**--**	**+-**	**--**	**--**
Basic performance analysis	**--**	**--**	**--**	**++**	**+-**
Dotted chart analysis	**--**	**--**	**+-**	**++**	**+-**
Basic log statistics	**--**	**+-**	**--**	**+-**	**--**
LTL checker	**--**	**+-**	**--**	**+-**	**+-**
Social network miner	**--**	**--**	**--**	**+-**	**++**
Role hierarchy miner	**--**	**--**	**--**	**+-**	**++**

Through the analysis of different graphs, charts and tables, basic or more specific interferences are drawn. The case study connected to the work order process in Cox serves to support the main research question of this master thesis, i.e. ‘How can process mining be used to analyze the performance of business processes in organizations?'. The examination of the event log connected to the activities executed for a few months proves the theory that process mining can be very useful for the process performance analysis of business processes. It can help organizations in their persuasion of right business operating, gaining of competitive advantage and keep the business in pace. More and more people from the business tend to enter the field. Process mining seems to be positive assessed by the business and respectively gains more and more participation in it to support the business processes performance in the enterprise.

Bibliography

Adriansyah, A., Van Dogen, B. F., & Van der Aalst, W. M. (n.d.). *Conformance checking using cost-based fitness analysis.* Technical University of Eindhoven, Department of Mathematics and Computer Science.

Bozkaya, M., Gabriels, J., & Van der Werf, J. M. (n.d.). *Process diagnostics: a method based on process mining.* Eindhoven University of Technology; Radboud University Nijmegen, Eindhoven.

Brandl, H., & Guschakowski, D. (2007). *Complex event processing in the context of business activity monitoring.* University of Applied Sciences , Regensburg.

Buijs, J. C. (2010). *Mapping data sources to XES in a generic way.* Eindhoven.

Calderon-Ruiz, G., & Sepulveda, M. (n.d.). *Improving business processes: Failure analysis with process mining.* Pontificia Universidad Catolica de Chile, Department of Computer Science, Chile.

Celino, I., Alves de Madeiros, A. K., Zeissler, G., Oppitz, M., Faaca, F., & Zoeller, S. (n.d.). *Semantic business process analysis.* CEFRIEL -Politecnico of Milano; Eindhoven University of Technology; IBIS Prof. Thome AG.

Coulson-Thomas, C. J. (1992). The learning organization.

Flapper, S. D., Fortuin, L., & Stoop, P. P. (1996). Towards consistent performance management systems. *International Journal of Operations and Production Management*.

Fluxicon academic initiative for process mining. (н.д.). Eindhoven. Свалено от http://win.tue.nl

Gamble, J., Strickland, A., & Thompson, A. (2007). Crafting & Executing strategy. (15).

Gunther, C. W., & Van der Aalst, W. M. (2008). *Trace clustering in process mining.* Eindhoven University of Technology, Eindhoven.

Gunther, C. W., Rinderle-Ma, S., Reichert, M., Recker, J., & Van der Aalst, W. M. (2007). Using process mining to learn from process changes in evolutionary system. *Business Process Integration and Management, 1*, pp. 111-111.

Hammer, M., & Champy, J. (1993). *Reengineering the corporation: A manifesto for business revolution.* New York: Harper Business.

Havey, M. (2005). *Essential business process modeling.* (A. Odewahn, & M. O'Brein, Eds.) Sebastopol: O'Relly Media.

Hornix, P. T. (2007). *Performance analysis of business processes through process mining.* Tecnical University of Eindhoven, Department of Mathematics and Computer Science, Eindhoven.

Neely, A. (2002). *Business performance measurement: Theory and practice.* Cambridge University Press.

Petri, C. A., & Reisig, W. (2008). Petri net. *3*. Свалено от http://scholarpedia.org

Rolland, C., & Pernici, C. T. (1998). A comprehensive view of process engineering. *10th International Conference CAiSE'98.* Pisa, Italy: Springer.

Rosset, A., & Sheldon, K. (2001). *Beyond the podium: Delivering training and performance to a digital world.* San Francisco: Jossey-Bass/Pfeiffer.

Van der Aalst, W. M. (2011). *Process mining: Discovery, conformance and enhancement of business processes.* Berlin: Springer.

Van der Aalst, W. M., & Van Hee, K. (2002). Workflow management: Models, methods and systems.

Van der Aalst, W. M., & Weijters, A. J. (n.d.). Process mining: a research agenda. *Computers in industry*.

Van der Aalst, W. M., De Beer, H. T., & Van Dongen, B. F. (n.d.). *Process mining and verification of properties: An approach based on temporal logic.* Eindhoven University of Technology, Department of Technology Management.

Van der Aalst, W. M., Nakatumba, A., Rozinat, & Russell, N. (n.d.). *Business process simulation: How to get it right?* Eindhoven University of Technology, Eindhoven.

Van Dongen, B. F., & Adriansyah, A. (n.d.). *Process mining: Fuzzy clustering and performance visualization.* Eindhoven University of Tecnology, Eindhoven.

Van Dongen, B. F., De Medeiros, A. K., Verbeek, H. M., Weijters, A. J., & Van der Aalst, W. M. (n.d.). The ProM framework: A new era in process mining tool support.

Vom Brocke, J., & Rosemann, M. (2010). *Handbook on business process management: Strategic alignment, governance, people and culture* (Vol. 1). Berlin: Springer.

Weiters, A. J., & Van der Aalst, W. (2001). Process mining discovering workflow models from event-based data.

http://www.businessdictionary.com

http://www.fluxicon.com

http://www.investopedia.com

http://www.mvpprograms.com

http://www.netmba.com

http://www.processmining.org

http://www.dataminingapps.com

http://www.thefreedictionary.com

Appendix 1

Table 6.2 Some process mining plug-ins in ProM 6

Plug-in	**Description**
Alpha miner	Discovers a Petri net using α-algorithm
Heuristic miner	Discovers a C-net using heuristic mining
Generic miner	Discovers a C-net using genetic mining
Fuzzy miner	Discovers a fuzzy model using fuzzy mining
Transition system miner	Discovers a transition system based on a state representation function and a log
Transition system to Petri net	Creates a Petri net based n a transition system
Declare miner	Discovers a declare model
ILP miner	Discovers a Petri net using language-based regions
Simple log filter	Filters a log by answering simple questions
Dotted chart analysis	Creates dotted chart showing all events at a glance
Trace alignment	Aligns events based on their context rather than time
Guide tree miner	Clusters cases in a tree based on similarities
Social network miner	Creates a social network based on a selected criterion
LTL checker	Checks a property
Fitness	Computes fitness on of Petri net based on event log
ETConformance	Checks conformance by counting "escaping edges" from the state space of the log to the state space of the model
Replay log on flexible model	Conformance checker based on A^* algorithm; can also be applied to Petri nets, C-nets and YAWL models
PomPom	Abstracts infrequently visited parts of a Petri net
Transition system analyzer	Creates a model to predict the remaining flow time

Appendix 2

Basic performance analysis

(9 different result charts)

Figure 6.2 a) presents the first possible result – the *bar chart*. It projects the first dimension chosen on the axis and the second – on the bars. When more than one dimension is chosen and the option *multiple* is chosen in the *display* field, a few graphs are drawn for each item in the second dimension. Time unit, measure, performance sort and different view of the bars can also be specified according to the options available in all these boxes.

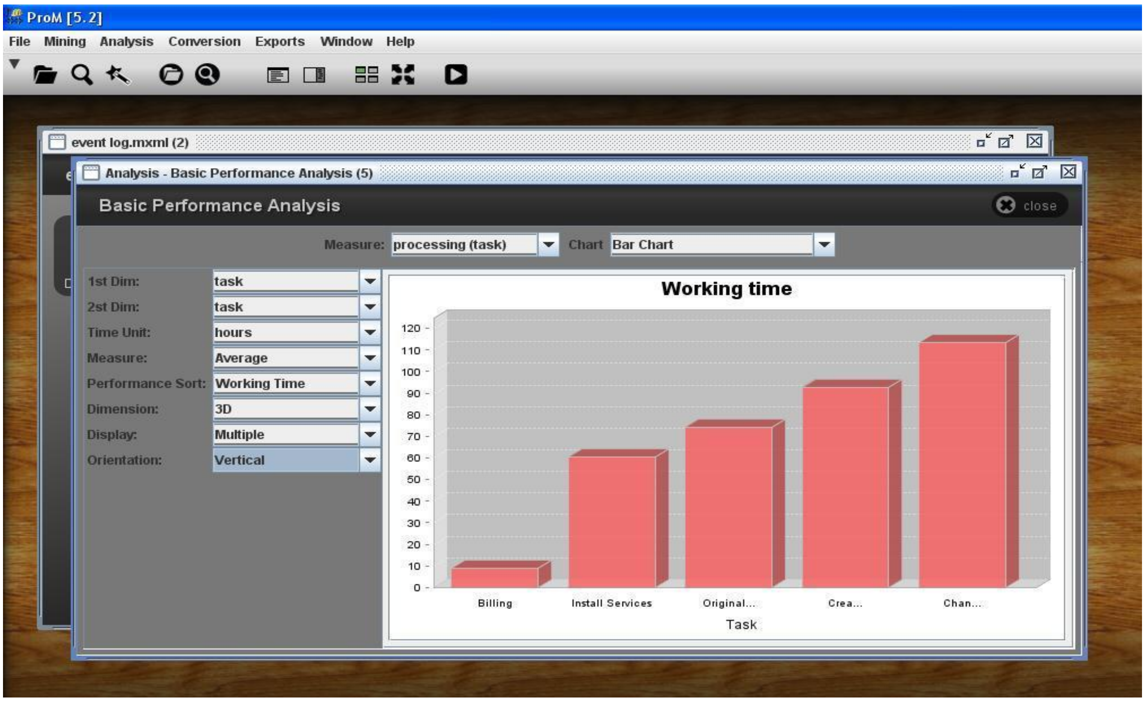


Figure 6.2 a) Bar chart

The second possible result graph is the *pie chart* (fig. 6.2 b)). The two dimensions selected are projected respectively to the pie and the chart (several charts are drawn for each item in the second dimension). Here similar options to these in the bar chart are offered.

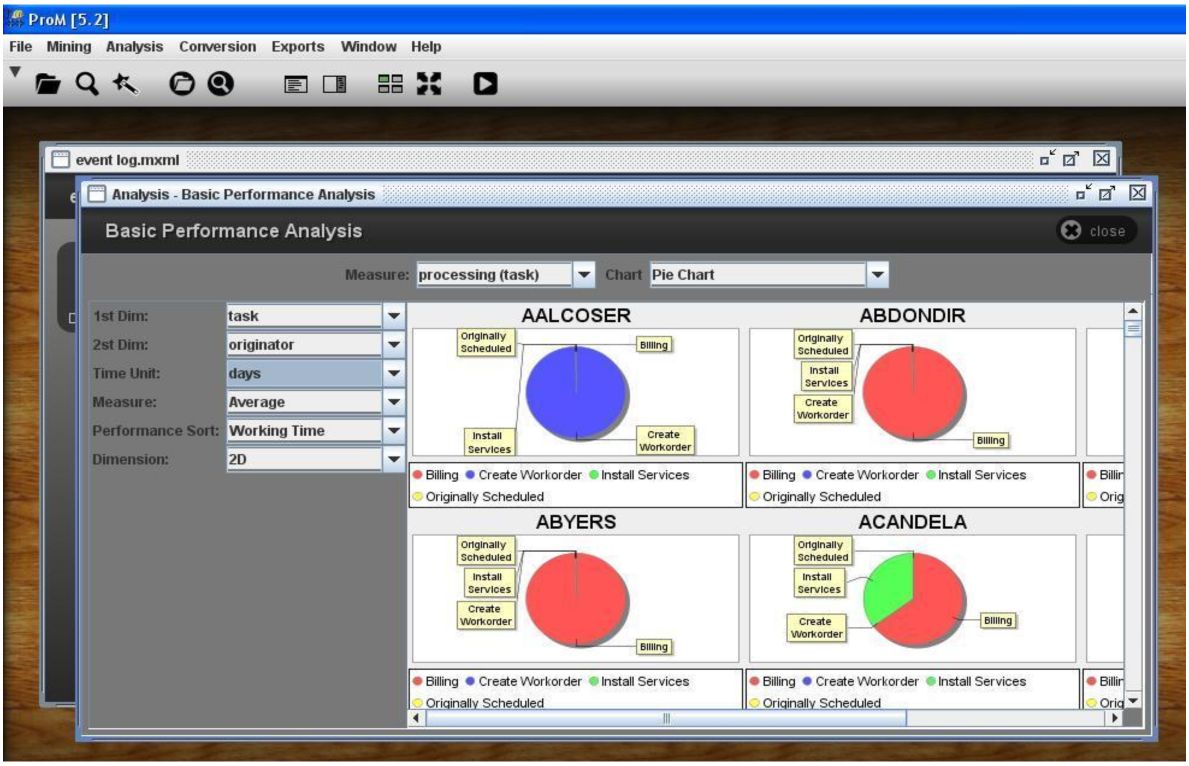


Figure 6.2 b) Pie chart

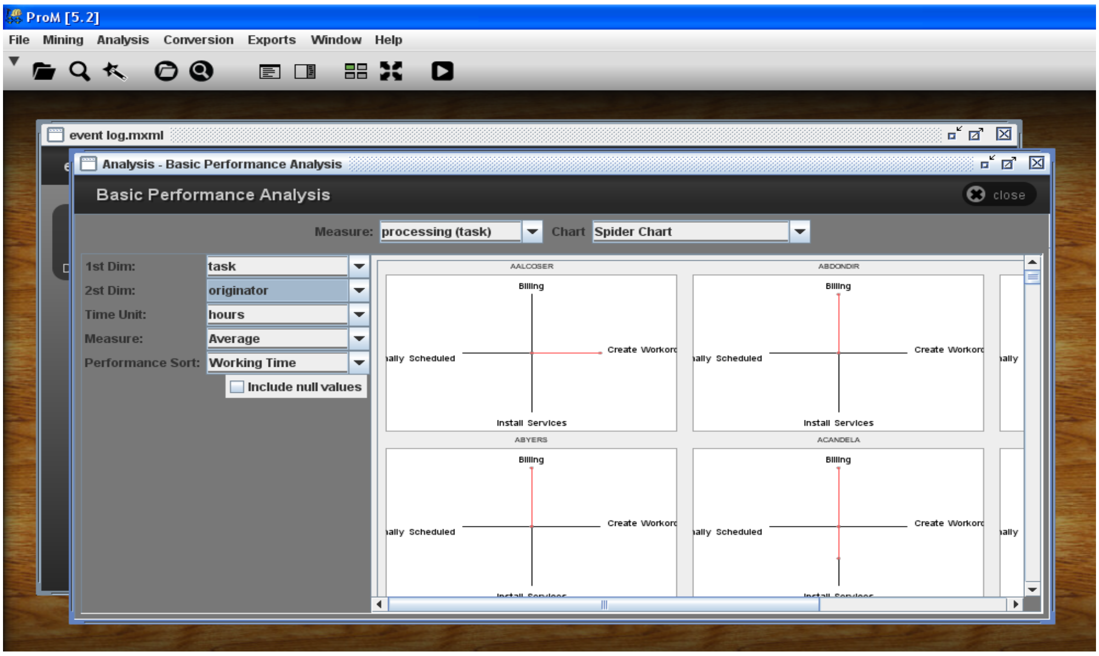


Figure 6.2 c) Spider chart

The *spider chart* shown on figure 6.2 c) is the third available result presentation included in the basic performance analysis plug-in. The first dimension is the one for the spider and for the second, several charts are drawn for each item in the second dimension selected.

Here null values can be included as well, i.e. items which value is 0. The other options are again similar to these to the bar chart.

The *Box-and-Whisker chart* (figure 6.2 d)) shows a minimum, a maximum, a median, an average, a first quartile and a third quartile. The first dimension is for the box-and-whisker and the second one is for the chart.

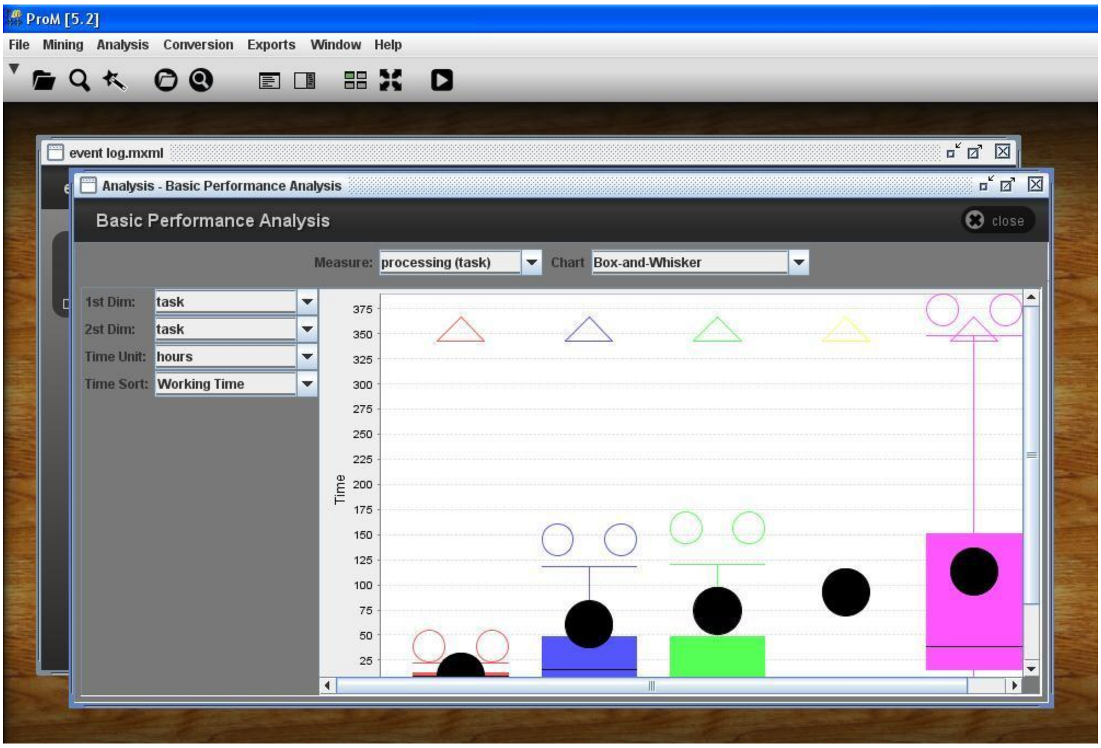


Figure 6.2 d) Box-and-Whisker chart

Figure 6.2 e) illustrates the *Gantt chart* and its options. It is obligatory two dimensions to be selected when projecting this chart. The previous result charts can also project charts on base of just one dimension but in this case – two of them are obligatory.

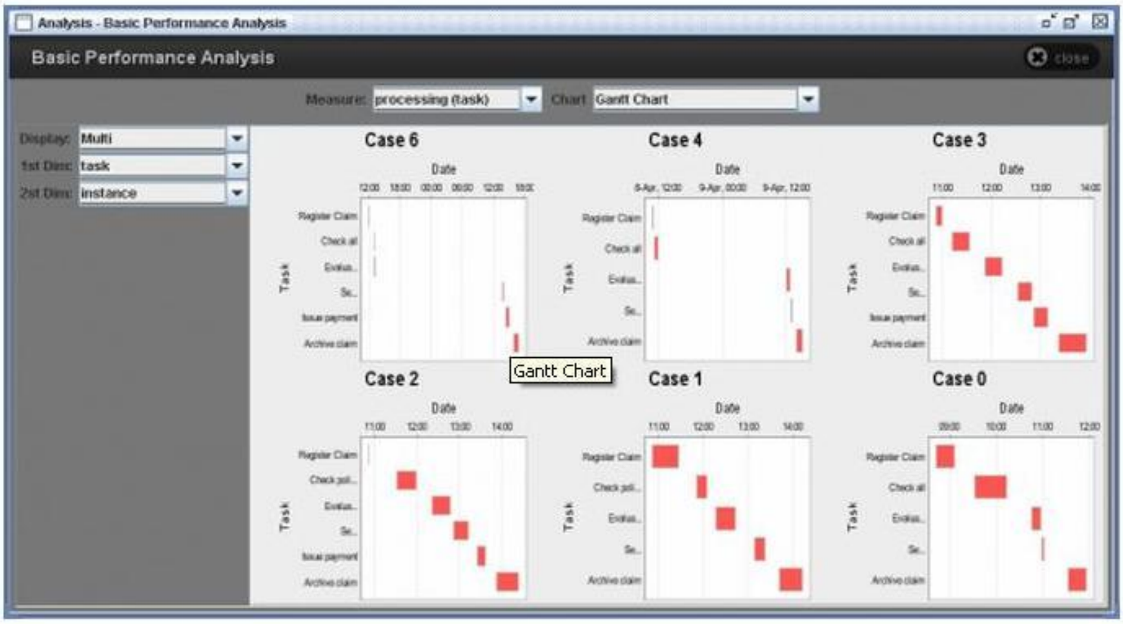


Figure 6.2 e) Gantt chart

The *time chart* shows a time graph at figure 6.2 f). The x-axis shows the time from the start to the end of the log. The y-axis reflects the number of working events or aggregated working hour along the x-axis. If the *overall* option is selected in the *display* box, one chart that reflects all events in a log is drawn. The *task*, *originator* and *instance* draw several charts respective to the item chosen. In the *measure* box, *time* and *frequency* of events can be selected. The line type box supports step and curve line types and finally start and end time can be defined.

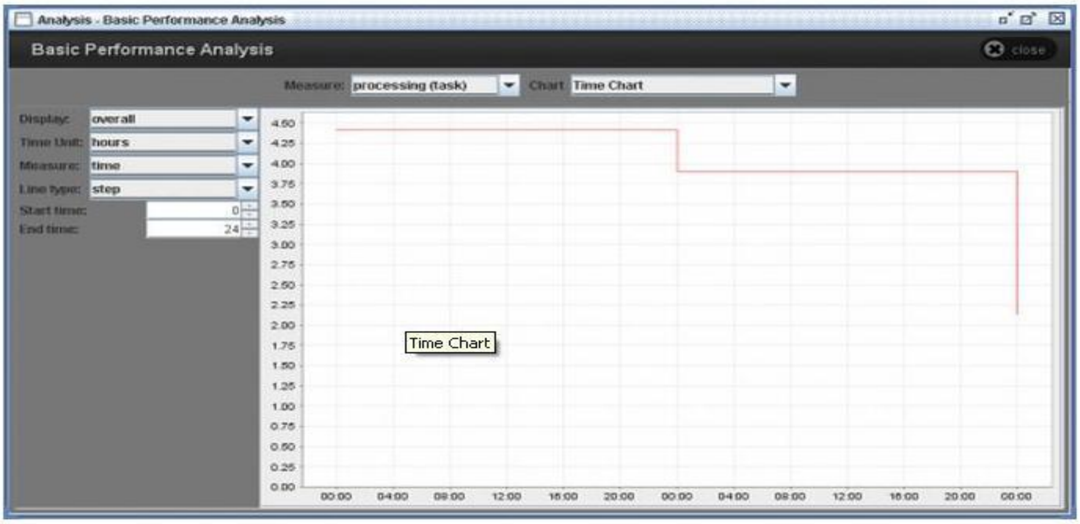


Figure 6.2 f) Time chart

The *day-hour chart* on figure 6.2 g) illustrates the density of work. The x-axis of the chart refers to a day (24 hours) and the y-axis shows the dates from the start till the end of the event log. Boxes on the chart refer to time units and their color represents the density of work.

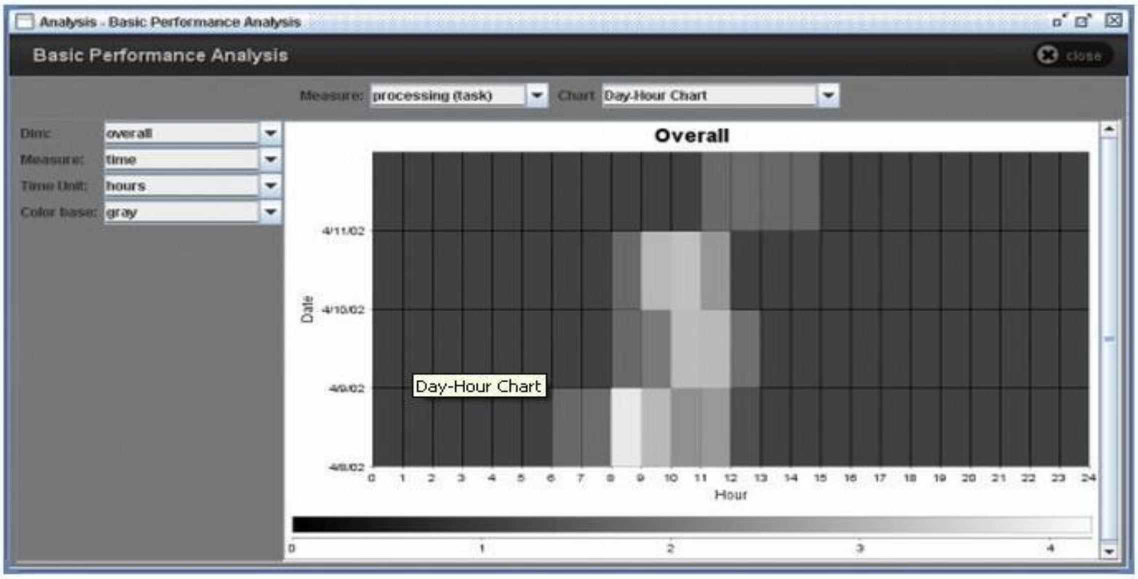


Figure 6.2 g) Day-hour chart

The *meter chart* is the eight way to represent results of the basic performance analysis plug-in. The graph has four ranges. The white one stands for area with no relevant values. The green shows normal, yellow – warning and red – critical zone. Ranges can be changed.

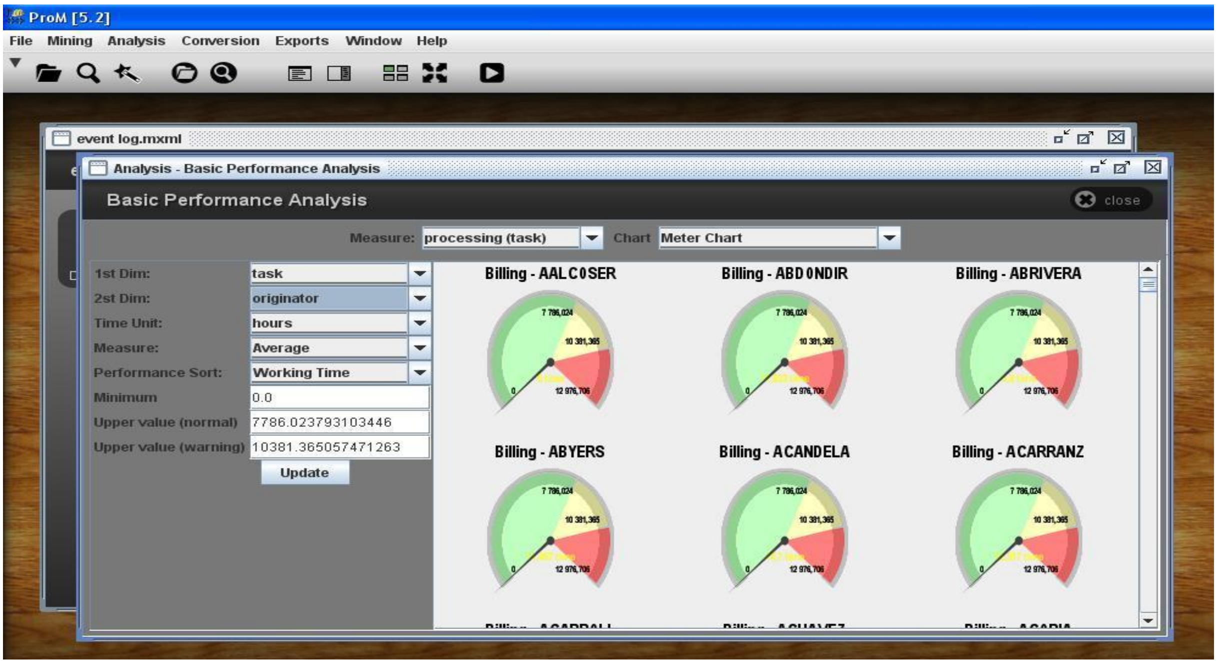


Figure 6.2 h) Meter chart

The last results available – the text view – appear in the form of table with values (see fig. 6.2 i)).

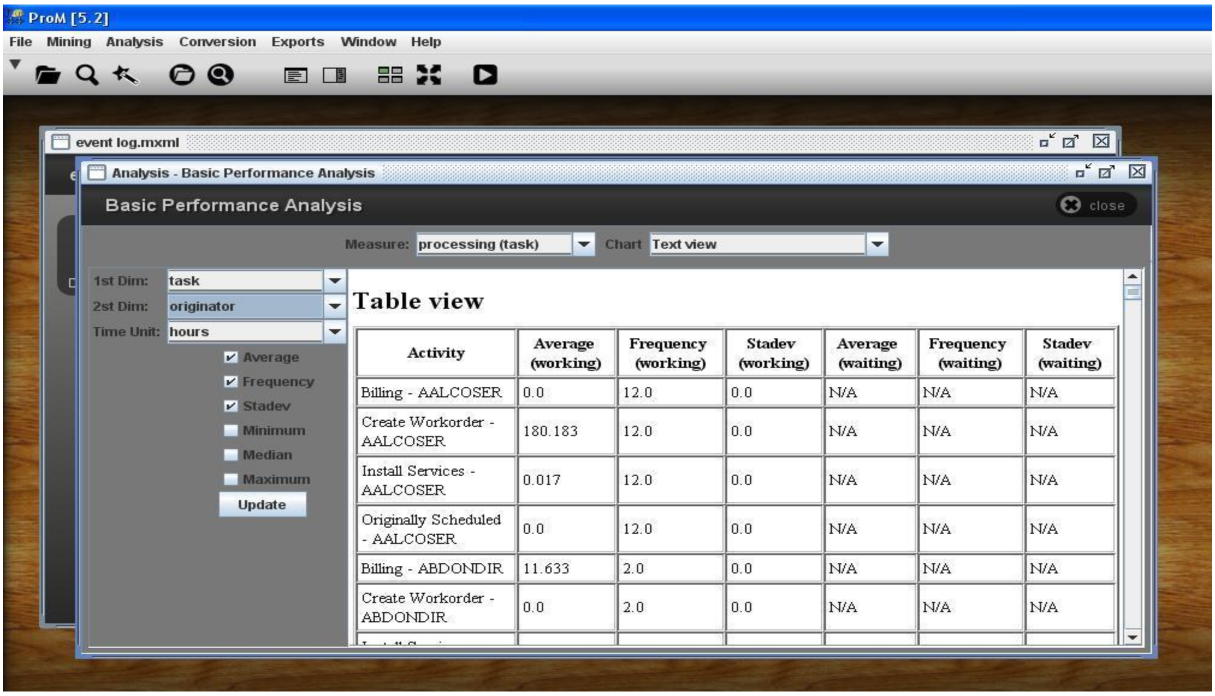


Figure 6.2 i) Text view

Auteursrechtelijke overeenkomst

Ik/wij verlenen het wereldwijde auteursrecht voor de ingediende eindverhandeling:
Performance analysis of business processed through the use of process mining

Richting: **Master of Management-Management Information Systems**
Jaar: **2012**

in alle mogelijke mediaformaten, - bestaande en in de toekomst te ontwikkelen - , aan de Universiteit Hasselt.

Niet tegenstaand deze toekenning van het auteursrecht aan de Universiteit Hasselt
behoud ik als auteur het recht om de eindverhandeling, - in zijn geheel of gedeeltelijk -,
vrij te reproduceren, (her)publiceren of distribueren zonder de toelating te moeten
verkrijgen van de Universiteit Hasselt.

Ik bevestig dat de eindverhandeling mijn origineel werk is, en dat ik het recht heb om de rechten te verlenen die in deze overeenkomst worden beschreven. Ik verklaar tevens dat de eindverhandeling, naar mijn weten, het auteursrecht van anderen niet overtreedt.

Ik verklaar tevens dat ik voor het materiaal in de eindverhandeling dat beschermd wordt door het auteursrecht, de nodige toelatingen heb verkregen zodat ik deze ook aan de Universiteit Hasselt kan overdragen en dat dit duidelijk in de tekst en inhoud van de eindverhandeling werd genotificeerd.

Universiteit Hasselt zal mij als auteur(s) van de eindverhandeling identificeren en zal geen wijzigingen aanbrengen aan de eindverhandeling, uitgezonderd deze toegelaten door deze overeenkomst.

Voor akkoord,

Boyanova, Veronika

Datum: **11/06/2012**