Eindhoven University of Technology

MASTER

Process mining tools
a comparative analysis

Ailenei, I.M.

Award date:
2011

Disclaimer
This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.
Process Mining Tools: A Comparative Analysis

Irina-Maria Ailenei

in partial fulfillment of the requirements for the degree of

Master of Science in Business Information Systems

Supervisor: prof. dr. ir. W.M.P. (Wil) van der Aalst (TU/e - W&I - AIS)
Tutor: dr. A. (Anne) Rozinat (Fluxicon)
Examination Committee:
prof. dr. ir. W.M.P. (Wil) van der Aalst
dr. A. (Anne) Rozinat
dr. ir. H.A. (Hajo) Reijers

Eindhoven, August 2011
Abstract

Process mining is an emerging topic that has attracted the attention of both researchers and vendors. As a result, several process mining solutions have become available on the market. Even though the development of the process mining field and the number of systems available is a positive thing, problems also arise. The most significant one is, perhaps, the lack of standardization and the fact that these tools use their own standards and naming. This can become rather confusing for a process mining user when deciding what is the most suitable system for a task at hand.

In this thesis, we propose an evaluation framework that is used to assess the strengths and the weaknesses of process mining tools. We apply the framework in practice for evaluating four process mining systems: ARIS PPM, Flow, Futura Reflect, and ProcessAnalyzer. The framework is based on a collection of use cases. A use case consists of a typical application of process mining functionality in a practical situation. The set of use cases was collected based on the functionality available in ProM and was validated by conducting a series of ten semi-structured interviews with process mining users and by handing out a survey. The validated collection of use cases formed the base of our tool evaluation.

For the actual evaluation, we created for each use case four dimensions: a brief description, a practical example of a situation in which the use case can be employed, a set of assumptions made with respect to the event log used to test the use case’s support, a set of acceptance criteria for deciding whether the use case is available or not in a system. Another part of the evaluation framework is the set of artificially generated event logs that were used to assess the support of the use cases.

Next to the use cases-based evaluation, we also test the performance limitations of the systems and we look at several extra functionalities, which are not covered by the collection of use cases (e.g. import and export capabilities, integration with external systems, etc.). Additionally, we discuss the behavior of the tools when analyzing event logs stemming from two real-life processes.
Acknowledgements

I would like to thank my supervisor Wil van der Aalst for all his feedback and guidance throughout the entire process.

I am deeply grateful to my tutor Anne Rozinat. Her motivation and dedication were an inspiration to me, while her good advice and thorough feedback were extremely helpful for my work.

I would also like to thank Hajo Reijers for being a member in my assessment committee. My gratitude also goes to Albert Eckert for his constant suggestions of improvement and for attending my presentation.

This project wouldn’t have been possible if it wasn’t for the process mining companies participating in the evaluation. I am very grateful for consenting to be part of our study and for the support I received every time I needed it. I would like to thank Tobias Blickle from Software AG, Jon Espen Ingvaldsen from Fourspark, Peter van den Brand from Futura Process Intelligence, as well as Antti Miettinen and Teemu Lehto from QPR.

Finally, I would like to express my gratitude for the other persons that played a role during my research. I thank Claus Nagler, Andreas Patzer, Frank van Geffen, Wim Leeuwenkamp, Piet Goeyenbier, Maria Haasnoot, Geert-Jan Rens, and prof. dr. Piet Bakker for their time and for sharing their ideas with me. Many thanks also go to the ones that took the time to fill in the survey.

Irina Ailenei

Eindhoven, August 2011
# Contents

## 1 Introduction

1.1 Thesis Motivation .................................................. 1
1.2 Project Goal ..................................................... 2
1.3 Research Questions ............................................... 2
1.4 Approach and Research Method ................................. 2
1.5 Related Work ................................................... 4
1.6 Thesis Outline .................................................. 4

## 2 Preliminaries

2.1 Process Mining and the ProM Framework ....................... 6
2.2 Commercial Process Mining Systems ......................... 7
   2.2.1 Software AG - ARIS Process Performance Manager (PPM) .... 8
   2.2.2 Fourspark - Flow ........................................ 9
   2.2.3 Futura Process Intelligence - Futura Reflect .......... 10
   2.2.4 QPR - ProcessAnalyzer ................................ 11
   2.2.5 Other systems ........................................... 12
2.3 Other Tools Used ............................................... 12
   2.3.1 CPN Tools ............................................... 13
   2.3.2 ProMimport Framework ............................... 13

## 3 Process Mining Use Cases

3.1 Approach .......................................................... 14
3.2 Definition of the Use Cases .................................... 15
   3.2.1 Discovery ............................................... 16
   3.2.2 Conformance Checking ................................. 16
   3.2.3 Enhancement - Extension - Organizational Perspective .... 16
   3.2.4 Enhancement - Extension - Time Perspective .......... 17
   3.2.5 Enhancement - Extension - Case Perspective .......... 17
3.3 Validation of the Use Cases .................................... 17
   3.3.1 Interviews .............................................. 17
   3.3.2 Survey ................................................ 19
   3.3.3 Conclusions of the Validation Phase .................. 22
3.4 Use Case Framework .......................................... 22
Chapter 1

Introduction

This thesis is the result of a graduation project carried out within Fluxicon\(^1\) and the Architecture of Information System (AIS) group\(^2\) of the Mathematics and Computer Science Department of TU/e. The goal of the project was the evaluation of commercial process mining systems based on a defined set of criteria. The outcome of this study is an evaluation framework based on process mining use cases that can be used to assess the functionalities provided by any process mining tool. Additionally, we present the results obtained after applying the framework in practice. We compared four commercial systems available on the market (ARIS Process Performance Manager, Futura Reflect, Flow, and ProcessAnalyzer) by using our evaluation framework.

This chapter introduces the context of this master thesis in Section 1.1 and presents the goal of the project in Section 1.2. The research questions that emerged from the project’s goal are described in Section 1.3 while the approach we followed as well as the research methodology we used are explained in Section 1.4. Related work is presented in Section 1.5. Section 1.6 concludes the chapter by describing the remaining structure of this thesis.

1.1 Thesis Motivation

The area of Process Mining has attracted the attention of both researchers and practitioners. As a consequence, a significant number of algorithms and tools have been developed within this domain. For instance, the academic process mining tool ProM version 5.2\(^3\) contains more than 280 pluggable algorithms, developed to provide a wide range of functionalities and techniques. Additionally, commercial process mining tools have also emerged on the market and these tools often use their own standards and naming.

The situation described above might be quite confusing for a potential user and makes it difficult to choose the most suitable process mining tool or algorithm for the task at hand. Moreover, in most of the cases, the needs of a process mining user do not cover the entire spectrum of available process mining functionalities, but are focused on a subset of analysis techniques depending on the context in which process mining is used. Currently, there is, however, no possibility of consulting an overview of

\(^1\)http://www.fluxicon.com/
\(^2\)http://www.win.tue.nl/ais
\(^3\)http://promtools.org/prom5/
the capabilities offered by the available process mining tools and of comparing their strengths and the weaknesses before choosing to use one tool or the other.

1.2 Project Goal

This project aims at overcoming the problems stated in the previous section. Since the problems are caused by the lack of an overview on the functionalities provided by the commercial process mining tools, our purpose is to perform an objective analysis of these tools aimed at providing insights on the support offered for different process mining functionalities. This analysis will be conducted from the user’s perspective based on a set of practical examples describing different process mining techniques. These practical examples capture situations in which process mining can be used for various types of process analysis and are referred to as process mining use cases.

The goal is to perform, thus, an objective and repeatable evaluation of the available process mining systems, having the set of use cases as basis. For this, we develop an evaluation framework consisting of use cases and event logs. The applicability of the framework is tested in practice by evaluating four systems. The criterion used when selecting a tool for the evaluation was its ability of performing process discovery. In order to make the results reproducible and to enable the evaluation of other process mining tools using the same framework, the event logs used for the evaluation will be made available.

The goal of the project can be therefore summarized as follows:

*Develop an evaluation framework based on a set of typical process mining use cases that can be used to assess the strengths and the weaknesses of different process mining tools. This framework will then be applied in practice with the purpose of evaluating several commercial process mining tools.*

Note that the framework can be used to analyze other contemporary tools and tools that will become available in the next couple of years.

1.3 Research Questions

The main objective of the project can be split into several research questions. Each question is the central focus of a certain step in the project and its outcome is used as input for the subsequent questions. The research questions formulated and addressed in the context of this work are:

1. What are the current functionalities provided by process mining techniques?
2. What are typical process mining use cases?
3. Is the identified list of use cases relevant and comprehensive?
4. Which process mining tools are suitable for which use cases?
5. What additional functionalities are provided by the evaluated tools?
6. What is the behavior of the tools when analyzing real-life processes?

1.4 Approach and Research Method

The approach we followed for addressing the research questions consisted of three major phases, which are also depicted in Figure 1.1.
1.4. APPROACH AND RESEARCH METHOD

We began by defining the set of process mining use cases that will be used as the basis for the tool evaluations. In order to do this, we first conducted a literature study focused on academic articles, the functionality available in ProM, and the use cases described in the marketing material of commercial process mining systems. In this step we aimed to address the first research question (What are the current functionalities provided by process mining techniques?). Based on the findings of this step we tackle the second research question (What are typical process mining use cases?) by defining the list of process mining use cases. A use case was focused on describing a particular situation in which a certain process mining techniques would be employed. The third research question (Is the identified list of use cases relevant and comprehensive?) is addressed through the validation step, which was mainly aimed at checking the completeness of the list and at verifying whether all the use cases in the list are relevant in practice. During the validation step we conducted a series of ten semi-structured interviews and performed a survey among process mining users. Based on the feedback received, we developed a use case framework that incorporated descriptive aspects of a use case as well as more technical aspects related to the practical setup of the evaluation.

After defining the list of use cases, we moved to the next phase of the project. The focus of this phase was on preparing the setup for the evaluation. In this context, we generated artificial event logs to be used for checking the support of the tools for the use cases, we adapted a set of real-life event logs based on the input requirements of each system, and we prepared the process mining tools for the evaluation.

The evaluation of the tools constituted the last phase of the project. The evaluation phase consisted of three steps. The first one was the evaluation of the process mining systems based on the defined use case framework. The results of this represent our answer for the fourth research question (Which process mining tools are suitable for which use cases?) we identified. Besides the use case-based evaluation, we also analyze the tools from the point of view of some additional functionalities not covered in the list of
use cases. This sort of analysis emerged from the interviews and from the survey responses, in which the process mining users stated their interest into functionalities like import and export capabilities, filtering, and animation features. We tackle the last question (What is the behavior of the tools when analyzing real-life processes?), by using real life logs from three different settings with different characteristics (municipality, hospital, and test processes).

The above can be summarized as follows. The research methods we used throughout this project are: conducting a literature study, performing interviews with process mining users, evaluating the results of a survey, defining the set of use cases, generating the event logs needed for the evaluation and performing the actual tools evaluation.

1.5 Related Work

This project was done in the area of process mining. Process mining consists of a set of techniques that aim at analyzing processes based on the event logs generated by the systems supporting them [1]. The process mining spectrum is divided in three classes: discovery [2, 3, 4, 5, 6], conformance [7, 8, 9, 10], and enhancement [11, 12, 13, 14]. A more detailed overview on process mining is given in Section 2.1.

Since process mining is an emerging research domain, the focus has been mostly on developing new techniques and algorithms that have broaden its scope. However, little work has been done on defining a standard set of use cases or on evaluating the existing process mining techniques and systems.

The importance of assessing the quality of the models discovered using different process discovery techniques is motivated in [15]. The authors propose four orthogonal dimensions (fitness, precision, generalization, and structure) that can be used to determine the quality of a discovered model by comparing it with the input event log. In addition, the need of having a common and standard event log repository used to test different process discovery algorithms is identified.

Such a benchmark set of event logs is developed in [16]. Its purpose is to enable the comparison of different process discovery algorithms by investigating the influence of varying the distributions of choice points, execution times as well as the ability of discovering and representing advanced constructs. The quality of a model is assessed using the four metrics proposed in [15]. The benchmark set is applied by comparing four process discovery algorithms implemented in ProM 5.2.

The work described in [17], done as part of a bachelor project, analyzes several process mining tools based on a set of criteria, including aspects like the supported modeling languages, the import format required, the integration with external software systems, the algorithms used for discovering process models, the support for the three process perspectives (control-flow, organizational, and case), the existence of tutorials and documentations, as well as the export capabilities. The analysis includes both commercial and open-source process mining systems. The open-source systems are the ProM Framework, Process-Mining-Workbench, and RapidMiner, while from the class of commercial systems ARIS Process Performance Manager, BPMone, Fujitsu Automated Business Process Discovery & Visualization, and Futura Reflect were analyzed.

1.6 Thesis Outline

The remainder of this thesis is depicted in Figure 1.2. Chapter 2 provides some preliminary information about the concepts and tools used throughout the project. In Chapter 3 we introduce the list of process
mining use cases that were defined based on the current functionalities provided by process mining techniques and we discuss the results of the phase in which we validated the defined set of use cases. Chapter 4 gives details about the steps undertaken to prepare the practical evaluation of the tools by presenting the set of event logs used and the input formats required by each system. We present the results of the use case-based evaluation as well as a couple of additional functionalities not covered in the use cases, in Chapter 5. In Chapter 6 we test the behavior of the evaluated systems when dealing with event logs stemming from real-life processes and we discuss the findings. The thesis concludes with Chapter 7. In this chapter, we summarize the work we have done, present the conclusions, and provide recommendations for future work.

Figure 1.2: Thesis outline
Chapter 2

Preliminaries

This chapter provides a short overview of the preliminary concepts used throughout the thesis. In section 2.1 the concept of process mining and the ProM framework are briefly discussed. Section 2.2 states the criteria used to include a process mining tool in this study and introduces the systems participating in the evaluation. Section 2.3 describes the other tools used in this thesis.

2.1 Process Mining and the ProM Framework

Process mining consists of a set of techniques that combine aspects from process modeling and analysis with data mining and machine learning. The goal of process mining is to exploit the data recorded by information systems in the form of event logs by extracting different kinds of information related to the analyzed processes [1]. The general idea of process mining is depicted in Figure 2.1. From the figure it can also be seen that process mining techniques are typically divided into three classes: discovery, conformance and enhancement.

Discovery focuses on retrieving the control flow of a process based on the traces generated by its past executions without using any a-priori model. The challenge is to discover a process model that closely represents the behavior observed in the event log, but at the same time does not allow for too much extra behavior. Examples of process discovery algorithms are described in [2, 3, 4, 5].

Conformance checking [7, 8, 9, 10] compares an existing process model with an event log belonging to the same process to determine whether the behavior observed in practice conforms to the documented process. This class of process mining techniques enables the detection and location of deviations that may occur in reality.

Enhancement techniques aim at extending or redesigning a given process model with the help of the additional information recorded in the event log. Based on this, a process could be repaired in such a way to better reflect the reality observed in the log, or it can be extended with performance [14], resources [11, 12] or decision rules [13].

Traditionally, process mining has been focusing on analyzing data in an off-line setting. This means that only the completed cases from the event log have been taken into consideration when applying the mining techniques. However, recent research in the domain has also enabled the possibility of performing online process mining, based on the data corresponding to the cases that are not completed at the moment of the analysis. This kind of analysis is known as operational support [18] and it incorporates activities like detecting possible deviations at runtime, predicting values for the parameters of the process and
The process mining functionalities described above are included in the ProM tool. ProM is an extensible open-source framework developed to support a wide variety of process mining techniques. Different algorithm implementations serving different process mining functionalities are available in ProM in the form of plugins. ProM has been developed by the process mining group at TU/e in collaboration with research groups all over the globe.

ProM is able to import event logs compliant with the MXML [19] or XES [20] formats and can load process model definitions in different standards. Some of the main functionalities provided by ProM are: discovering the control-flow perspective of a process, mining less structured processes, analyzing the resource and the data perspective of a process, verifying the conformance of an event log with a given process model and inspecting the performance metrics specific to a process. Additionally, the tool supports a wide variety of filtering techniques which can be used on an event log in the preprocessing step. ProM provides several export formats, ranging from visual representations of the results (e.g. PNG) to different log formats (e.g. CSV).

As an academic platform ProM is at the forefront of process mining research. It is suitable for process mining researchers and makes it easy to develop and test new algorithms. However, it requires a certain expert level to use it and has no professional support. Therefore, commercial process mining tools may be a better choice for professionals who use process mining techniques as a support for their core activities.

2.2 Commercial Process Mining Systems

This section aims at introducing and briefly describing the process mining systems that are evaluated in this study. The main selection criterion for considering a system for the evaluation is the ability of the tool to perform process discovery. More specifically, a system should be able to discover an unknown process structure based on logged traces of past executions. Therefore, the focus of this study is not on...
systems used to manually model business processes, but on the systems that analyze already existing processes based on captured event log data.

We identified eight process mining tools that met our selection criteria. Subsections 2.2.1 - 2.2.4 are dedicated to the process mining tools that agreed to participate in this evaluation, while subsection 2.2.5 enumerates the rest of the software that were considered, but decided not to take part.

2.2.1 Software AG - ARIS Process Performance Manager (PPM)

ARIS Process Performance Manager (PPM)\(^1\) belongs to Software AG and is a component of the ARIS Controlling Platform, aimed at analysing business processes based on historical data. ARIS PPM supports a wide range of input formats, such as CSV files, event logs extracted from databases and external systems (e.g. ERP, CRM, etc), PPM graph format, and PPM event system format. The discovered models can be exported as image files, XML, and AML files, while the data generated as a result of different types of analysis can be saved in Microsoft Office Excel files. Figure 2.2 depicts the main window of the system’s user interface.

![Figure 2.2: Screenshot of ARIS PPM](image)

The system supports process discovery using two different approaches. The first approach consists of two phases. Initially, the sequences of events corresponding to each process instance are constructed. Next, this information is aggregated at the process level, generating the process model, displayed in the EPC\(^2\) format. The second approach displays the activities sequence by counting the relations of succession for the entire event log.

ARIS PPM allows the definition and computation of various Key Performance Indicators (KPIs), such as throughput times, processing times, costs, processing frequencies, etc. Based on the values of the defined KPIs, different kinds of analysis are displayed as tables or graphs, which put together constitute the

---

\(^1\)http://www.softwareag.com/nl/products/aris_platform/aris_controlling/aris_process_performance/

\(^2\)http://en.wikipedia.org/wiki/Event-driven_process_chain
2.2. COMMERCIAL PROCESS MINING SYSTEMS

performance dashboard of the process. Using the process benchmarking capabilities, processes can be compared based on aspects including differences between their structures, organizational units involved, and KPIs.

The tool supports organizational analysis by identifying the communications relations between the involved organizational units and by connecting the activities of the process to the resources performing them.

2.2.2 Fourspark - Flow

Flow\(^1\) is a process mining tool developed by the Norwegian company Fourspark. The supported import format is Microsoft Office Excel 2003 (.xls) files. The user interface is built as a dashboard and provides flexibility by allowing the user to create various widgets that display different types of information. Figure 2.3 represents a screenshot of the user interface of Flow.

![Figure 2.3: Screenshot of Flow](http://fourspark.no/)

Once the dataset is imported into the system, filtering the event log is possible through the Search functionality. The selection of process instances that should be considered for further investigation is done by inserting a free text query. A query contains a set of terms that should occur in the traces corresponding to the relevant process instances.

Along with the functionality for discovering the structure of a process, called Process Variants Flow also provides a couple of additional features. For instance, it is possible to visualize a model constructed based on process instances occurring between two selected timestamps. Infrequent behavior can be left out from the model by modifying the value of a threshold and the visual representation of the model suggests which paths in the process are more common by using different variations of colors and thickness of the arrows connecting activities.

The Motion chart function allows the possibility to analyze how different performance measures evolve over time. If the execution of the analyzed process is spread over multiple geographical locations and information about these locations is available in the event log, it is possible to visualize the links between them on a map by making use of the Geo Map functionality.

\(^1\)http://fourspark.no/
Flow provides an overview on the data stored in the log through the Business Ontology feature, which builds the 3-level tree, with a root node, then all attribute names from the input file on the second level and all possible values for each attribute on the third level.

### 2.2.3 Futura Process Intelligence - Futura Reflect

Futura Reflect\(^1\) is a Dutch process mining system that can be either used as a stand-alone tool or as a component integrated in the BPM\(^{\text{one}}\) suite of Pallas Athena\(^2\). Futura Reflect is a web-based tool which supports the CSV format as input format for the event logs. After the analysis, the results can be exported can be saved as image files on the local machine or as FLOWer models that can be further on uploaded into the BPM\(^{\text{one}}\) platform. Another type of export that Futura Reflect supports is the Excel format that can be used to save data resulting from performance analysis.

The functionality available in Futura Reflect can be summarized using the following grouping: **Overview, Mine, Explore, Animate and Charting.** This grouping can also be observed in Figure 2.4, which depicts a screenshot of the tool.

The **Overview** function provides a summarized view of the main characteristics of the imported data set. There are three types of information given by this functionality spread over different tabs. The user can read a summary of the event log containing information such as the number of activities, cases and events in the log, as well as the values of the first and respectively the last timestamp present in the log. Furthermore, the user can visualize the attributes in the dataset by looking at their name and type, the number of unique values encountered, the minimum, and the maximum values. The last tab offers the possibility of displaying the sequence of activities, the activities timestamps and the originators for a select process instance.

The **Mine** functionality allows the discovery of process models based on the logged execution traces. The algorithm used is based on the genetic miner approach described in \([2]\) and only takes completed cases into consideration. The user can provide as input for the algorithm a target percentage that indicates how many of the cases in the event log can be replayed using the discovered model. Then, the system starts building various process models until the desired fitting percentage is achieved.

![Figure 2.4: Screenshot of Futura Reflect](http://www.futuratech.nl/site/)

\(^1\)http://www.futuratech.nl/site/
\(^2\)http://www.pallas-athena.com/
Another approach to discover the model corresponding to the analyzed process is to use the \textit{Explore} functionality. The differences between this functionality and the \textit{Mine} functionality are due to the different underlying algorithms. Depending on the algorithm, extra options are provided. The algorithm used by \textit{Explore} assumes a sequential behavior, which makes it impossible to detect parallelism, but has the advantage of filtering infrequent behavior. Moreover, the most frequent paths in the discovered model are represented by darker colors, a feature that makes the visualization of the result easier for the user. Another advantage brought by this functionality is the possibility of filtering the cases from the event log to be considered when building the process model. In addition to the process discovery function, \textit{Explore} is also able to provide an insight in the organizational perspective of a process, by constructing the social network of resources based on the handovers of work.

For dynamic insights into the process, the \textit{Animate} functionality is available. This capability enables the animation of process models by taking into account the timestamp information present in the event log. The user can thus better understand the dynamics of the process by looking at the way cases flow through the process and furthermore visually detect and localize possible bottlenecks.

The last functionality offered by Futura Reflect, \textit{Charting}, allows the creation of charts based on the data imported from the data set. The user can choose both the data to be included in the analysis and the chart type (e.g. Line Chart, Bar Chart, Pie Chart, etc).

\subsection*{2.2.4 QPR - ProcessAnalyzer}

ProcessAnalyzer\footnote{\url{http://www.qpr.com/products/qpr-processanalyzer.htm}} is an Excel-based process mining tool produced by the Finnish company QPR. The tool is installed as a Microsoft Office Excel Add-on. After the installation a new tab, called QPR ProcessAnalyzer, is created in the Microsoft Office Excel environment. The system is capable of importing any suitable event log that is currently open in Excel, provided that the log is contained in a sheet that has the title “Data”.

After loading the dataset, a new sheet called “Cases” is automatically created. This sheet contains information for each case present in the event log, such as the start and end date, the number of events, the frequency with which each event took place and the number of events executed by each resource involved in the process. Besides the “Case” sheet, two additional sheets are created in the Excel file. One sheet provides information about all possible process paths identified in the event log, together with the number of occurrences and the percentage relative to the entire log. The second sheet keeps a overview of the different filters that were used when loading the current dataset. The filtering of process instances can be done using the filtering functionality provided by Microsoft Office Excel. Based on the information extracted from the imported event log, QPR ProcessAnalyzer provides five different basic analysis types (see Figure 2.5).

The \textit{Process} capability shows the control flow of the process being analyzed. The control flow can be represented in four variants of flowcharts and there are also other visibility settings that control the level of detail of the information displayed in the result. For instance, infrequent sequences of events can be left out by changing the threshold of the number indicating the percentage an activity was executed. Additional functionality enables the user to visualize the structure of the process for a selected process instance.

The \textit{Variation} function provides insights about the variation in the process. This information is the same as the one in one of sheets previously mentioned. The difference consists of the visual representa-
CHAPTER 2. PRELIMINARIES

The organization of all possible paths in the process and the possibility of looking at the cases that have undertaken each path.

The Timeline analysis shows the duration of each case in days on the same timeline. Again, there is the possibility of selecting a particular case for viewing its control flow.

With the Path analysis the activities preceding or succeeding a particular selected event are displayed in a graph format. This function also displays the time elapsed between the selected event and any other successor or predecessor. The level of depth of the resulting graph can be varied.

The organizational perspective of a process can be investigated using the Resource analysis. More specifically, it is possible to visualize the work load corresponding to each resource or organizational unit involved in the process.

In addition to the possibility of using the filtering capabilities provided by Microsoft Office Excel, QPR ProcessAnalyzer makes it also possible to define filter by selecting to include or to exclude specific cases from the analysis. QPR ProcessAnalyzer also incorporates an animation functionality that allows the user to see the flow of cases through the process in between a selected start and end date.

Figure 2.5: Screenshot of QPR ProcessAnalyzer tab in Microsoft Office Excel

2.2.5 Other systems

As previously mentioned, the main criterion for considering a process mining tool for this study was the presence of a process discovery capability. Besides the tools that are evaluated in this thesis, the initial list of considered systems also included: Iontas - Process Discovery Focus\(^1\), Open Connect - Comprehend\(^2\), Fujitsu - Interstage Automated Process Discovery\(^3\) and StereoLOGIC - Discovery Analyst\(^4\). However, the vendors of these systems did not agree to participate in this evaluation study and therefore we could not evaluate them. Hence, we could not determine whether they indeed have process mining capabilities.

2.3 Other Tools Used

This section gives a short overview of two tools, CPN Tools\(^5\) and ProMimport\(^6\) that were used during the preparation of the experimental setup. More specifically, these two tools were involved in generating and converting respectively the artificial event logs used to test the functionality provided by the analyzed process mining systems.

---

\(^1\)http://www.iontas.com/pages/products/pdf.php
\(^2\)http://www.oc.com/technology/
\(^3\)http://www.fujitsu.com/global/services/software/interstage/solutions/bpm/apd.html
\(^4\)http://www.stereologic.com/stereologic_software.htm
\(^5\)http://cpntools.org/
\(^6\)http://www.promtools.org/promimport/
2.3. OTHER TOOLS USED

2.3.1 CPN Tools

CPN Tools is a tool that supports the modeling of concurrent processes using Coloured Petri nets. Tokens have types assigned and the nets defined can be extended with time and hierarchies and manipulated using the underlying functional programming language, SML. Another important functionality provided by CPN Tools is the ability to simulate nets both in a step-by-step manner controlled by the user and in an automatic manner, which is faster and can generate statistical information. Figure 2.6 shows a screenshot of a Coloured Petri net being simulated manually.

In the context of this thesis, CPN Tools was used to generate MXML event log fragments that were afterwards aggregated into event logs with the help of the ProMimport framework. The generation of these fragments is possible using the logging functions available in the CP-net extension described in [21]. The first step of this approach was to define processes using the CPN notation. Next, the processes were simulated for different cases and for each of these cases a separate MXML event log fragment was created. The last step, i.e. the aggregation of all MXML fragments belonging to the same process, was performed using the ProMimport framework, which is described in the next subsection.

![Figure 2.6: Screenshot of CPN Tools](image)

2.3.2 ProMimport Framework

The ProMimport framework was designed and implemented with the purpose of converting process data having different formats into MXML event log files. The tool consists of several filters that extract event logs stored in systems, such as SAP R/3, FLOWer or Staffware and convert them into MXML event logs. The main benefit brought by ProMimport framework is the possibility to apply process mining techniques on logs originating from real information systems.

As previously mentioned, the ProMimport framework was used to aggregate the MXML event log fragments generated after simulating a process using CPN Tools. For this purpose, the CPN Tools filter was used [21]. The filter merges the MXML fragments into a MXML event log containing all the execution traces belonging to the same process. Such an event log was the start point in performing the systems evaluation. The rest of the steps followed are described in detail in Chapter 4.
Chapter 3

Process Mining Use Cases

Our goal is to develop an evaluation framework based on a set of practical use cases. This chapter presents the steps of the process we followed for gathering and validating the set of process mining use cases. We begin by giving an overview on the approach used in Section 3.1. Next, the collected used cases are introduced in Section 3.2 by presenting a brief description of each use case. In Sections 3.3.1 and 3.3.2 we discuss the main results of the validation phase through interviews and survey, while Section 3.4 presents the use case framework used for the evaluation of tools.

3.1 Approach

An important element of our study was to decide which approach we are going to follow in defining and validating the list of use cases to be used for the tools evaluation. Since there was no standard reference for process mining use cases, we followed an inductive approach, similar to the one described in [18], which aimed at defining a list of process mining functionalities needed in practice that is as complete and relevant as possible. Figure 3.1 illustrates the sequence of steps that constitute the approach we followed to collect a representative collection of use cases.

Figure 3.1: The four phases of the approach followed to create a validated collection of use cases

Literature Study

Definition of Use Cases

Validation through Interviews

Validation through Survey

**Literature study** The purpose of the literature study was to get an overview about the existing functionality available in the context of process mining. In order to do this, we looked at the functionality provided by the process mining tool ProM [22] and focused our attention on academic articles about process mining techniques. We studied papers on process discovery [2, 3, 23], conformance checking [24], organizational analysis [11], and process mining case studies [25, 26]. In addition, we consulted the marketing brochures and descriptions of a couple of commercial process mining tools [27, 28, 29, 30, 31].

**Definition of Use Cases** The next step was the definition of an initial list of process mining use cases. We consider a use case to represent the use of a concrete process mining functionality with the goal to obtain an independent and final result. Therefore, actions performed before the actual analysis,
3.2. DEFINITION OF THE USE CASES

like the import of the event log or filtering, are not included in our list. When defining the list of use cases, we used the classification of process mining techniques described in [1]. Figure 3.2 shows a simplified representation of this classification and also shows our scope in relation with the entire classification. The definition of use cases is thus restricted to the offline analysis and does not include any techniques that deal with prediction, detection or recommendation. This limitation was introduced due to the inability of evaluating the systems participating in the study in an online analysis environment. The description and examples of each use case are introduced in Section 3.2.

![Figure 3.2: The project’s scope in the context of process mining](image)

**Validation through Interviews** The great number of existing process mining techniques and the lack of a standard list of use cases led to the need of validating the defined list. We started our validation phase by conducting a series of ten semi-structured interviews with practitioners having process mining expertise. First, we wanted to verify the understandability of the descriptions of the use cases by asking them to provide examples with situations in which each use case would be useful. Second, the goal of the interviews was to validate the list of use cases by removing the use cases that the participants considered irrelevant, and by determining whether there are use cases missing from the initial set. Furthermore, we wanted to find out whether there are differences between the importance of each use case for different categories of end users. One lesson learnt from the interviews was that participants have the tendency of saying that all use cases are equally important. As a result of this observation, we deviated from the approach described in [18], where use cases were just classified as important or not important, and instead used the sorting method for ranking the use cases based on their importance. The findings of the interviews are presented in detail in Section 3.3.1.

**Validation through Survey** Distributing a survey among people familiar with the field of process mining was the most suitable method to collect a larger number of responses for the validation phase. In total, we obtained 47 responses. The main goals of the survey were to capture the context of the respondents by asking for their role and domain, get the use cases rankings, and find out what additional functionality not covered by the list of use cases is considered important and should be included in our tool evaluation. The results of the survey are discussed in Section 3.3.2.

3.2 Definition of the Use Cases

This section introduces the list of process mining use cases by providing a short description of each use case. A more complete presentation, containing in addition a practical example for every use case,
is given in Appendix A. The use cases are grouped into the categories described in §3.2.1. Section 3.2.1 contains use cases belonging to the process discovery part, subsection 3.2.2 focuses on the conformance checking use cases, while Sections 3.2.3, 3.2.4, 3.2.5 present the use cases related to the organizational, the time, and the case perspective.

### 3.2.1 Discovery

The use cases belonging to this category are focused on the control flow perspective of the process. The user gets a clear understanding of the analyzed process by looking at its structure, frequent behavior and at the percentages of cases following every discovered path.

**Use case 1: Structure of the process.** Determine the structure of an unknown process or discover how a process looks like in practice.

**Use case 2: Routing probabilities.** Get a deeper understanding of the process by looking at the probabilities of following one path or another after a choice point.

**Use case 3: Most frequent path in the process.** Discover the path in the process that is followed by the highest percentage of cases. A path is considered to be a sequence of activities.

**Use case 4: Distribution of cases over paths.** Discover common and uncommon behavior in the process by looking at the distribution of cases over the possible paths in the process. A path is considered to be a sequence of activities.

### 3.2.2 Conformance Checking

This category consists of use cases which have the purpose of checking whether the process has the intended behavior in practice. The use cases pertaining to this category have in common that in order to execute them one needs an additional input besides the event log of the process to be analyzed. This input may be a reference model of the process or a rule which the discovered process has to be checked against.

**Use case 5: Exceptions from the normal path.** Discover the outliers of the process by looking at the least frequent behavior observed in practice. A path is considered to be a sequence of activities.

**Use case 6: The degree in which the rules are obeyed.** Check whether the rules and regulations related to the process are obeyed.

**Use case 7: Compliance to the explicit process.** Compare the documented process model with the real process, either given by the event log either given by a explicit model.

### 3.2.3 Enhancement - Extension - Organizational Perspective

The focus of the use cases included in this category is on the organizational analysis. The outcome of executing these use cases provides the user with an insight in the issues related to the resource perspective of the process.

**Use case 8: Resources per task.** Discover the relation between resources and tasks.

**Use case 9: Resources involved in a case.** Discover the group of resources involved in executing a particular case.

**Use case 10: Work handovers.** Manage resource location or determine possible causes for quality and time issues by looking at how work is transferred between resources.

**Use case 11: Central employees.** Determine who the central resources for a process are by analyzing
3.3. VALIDATION OF THE USE CASES

the social network based on handovers of work.

3.2.4 Enhancement - Extension - Time Perspective

As performance-related insights are most valuable, most of the use cases related to enhancement correspond to the time perspective.

**Use case 12: Throughput time of cases.** Determine the time that passed since the start of a case in process until its completion.

**Use case 13: Slowest activities.** Discover potential time problems by looking at the activities in the process that take a long time.

**Use case 14: Longest waiting times.** Determine delays between activities by analyzing the waiting times before each activity.

**Use case 15: Cycles.** Learn whether additional delays occur in the process due to cycles.

**Use case 16: Arrival rate of cases.** Determine the frequency with which new cases arrive in the process.

**Use case 17: Resource utilization rate.** Determine what are the utilization rates of the resource i.e, measure the fraction of time that a resource is busy.

**Use case 18: Time sequence of events.** Get a deeper understanding on the organization of a process by looking at the time sequence of activities for a specific case. (e.g. Gantt-graph for activities).

3.2.5 Enhancement - Extension - Case Perspective

The case perspective of the process is represented by a single use case.

**Use case 19: Business rules.** Discover what are the process attributes that influence the choice points and what are the conditions for following one branch or another.

3.3 Validation of the Use Cases

The use cases were validated by conducting ten interviews (Section 3.3.1) and by distributing a survey (Section 3.3.2) among process mining users and experts.

3.3.1 Interviews

We conducted in total ten interviews with process mining users and domain experts. The interviews can be divided into two categories: (1) interviews aimed at gaining some qualitative feedback on the understandability of the use cases and (2) interviews which were focused on obtaining a ranking of the use cases based on their importance for the interviewees and on identifying missing use cases.

(1) Based on the feedback received from the first type of interviews (in total: four) two non-relevant use cases were removed from the list, the descriptions of a couple of use case were refined and a short motivation was added for each remaining use case. The two irrelevant use cases referred to the possibility of identifying the paths in the process taking most time and to the possibility of visualizing the list of process attributes stored in the event log. The aim of refining the use case descriptions and of adding the motivation dimension was to increase the understandability and clarity of what each use case is about and what its practical purpose is.
(2) In the second type of interviews (in total: six) we asked the interviewees to sort the list of cases in the order of their importance in practice and on discovering any missing use cases. Moreover, we were interested in gaining additional insights on what are the functionalitites that a process mining tool should provide to its users. These interviews were structured in three parts. The first part aimed at getting information about the experience of the interviewee in the context of process mining and about the added value that process mining brings to their work. Secondly, the interviewees were shown the list of use cases and were asked to assign to each use case a score from 1 to 19 based on its importance (1 being the most important). The last part of the interview was meant to summarize the discussion, to learn about possible use cases missing from the initial list and about additional functionality that interviewees consider useful in a process mining tool. The complete summary of the outcomes of these six interviews can be found in Appendix B.

The six interviews we conducted were balanced from the point of view of the interviewee’s role in the context of using process mining techniques. Three of the persons interviewed were process analysts and the other three were auditors. The second dimension we took into account when selecting the interviewees was the domain they belong to. In this context we aimed at having a broader range of domains and therefore we talked with people working in the banking industry, healthcare, public sector, and business process consulting.

![Figure 3.3: Use cases ranking results from the interviews with process analysts and auditors](image)

Figure 3.3 depicts the profiles of process analysts and auditors based on the use case rankings collected from our interviews. On the x-axis we refer to use case numbers, while the y-axis represents the averages of the scores the use cases were assigned during the interviews. The graphic shows there are some differences in ranking the use cases based on the profile of the respondents. For instance, use case 12 (Throughput time of cases) is one of the most important use cases according to the process analysts group, while the auditors consider this quite irrelevant in practice. The opposite holds for use case 5 (Exceptions from the normal path), which is ranked as highly important by the auditors and less important by the process analysts.

Furthermore, the top five and bottom five use cases were extracted for each category of respondents (cf. Table 3.1 and Table 3.2). Our expectations regarding the difference in needs of people having
different roles are confirmed by comparing the top five use cases for each category. The contents of the
top rankings are quite different, except for two use cases that are considered important by all: discovering
the structure of a process and looking at the distribution of cases over the paths in the process.

When comparing the rankings of the least interesting use cases, one can also identify some similari-
ties. Four use cases are common for both rankings. Respondents, independent of their role, consider that
determining the group of resources performing a task and the group of resources involved in a case, as
well as looking at the central employees of a process and at the arrival rate of cases in the process are
less relevant use cases.

Table 3.1: Top 5 and Bottom 5 Use Cases for Process Analysts

<table>
<thead>
<tr>
<th>Top 5 Use cases</th>
<th>Bottom 5 Use Cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>U3. Most frequent path in the process</td>
<td>U8. Resources per task</td>
</tr>
<tr>
<td>U1. Structure of the process</td>
<td>U11. Central employees</td>
</tr>
<tr>
<td>U15. Cycles</td>
<td>U9. Resources involved in a case</td>
</tr>
<tr>
<td>U12. Throughput time of cases</td>
<td>U5. Exceptions from the normal path</td>
</tr>
<tr>
<td>U4. Distribution of cases over paths</td>
<td>U16. Arrival rate of cases</td>
</tr>
</tbody>
</table>

Table 3.2: Top 5 and Bottom 5 Use Cases for Auditors

<table>
<thead>
<tr>
<th>Top 5 Use cases</th>
<th>Bottom 5 Use Cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>U1. Structure of the process</td>
<td>U10. Work handovers</td>
</tr>
<tr>
<td>U7. Compliance to the explicit process</td>
<td>U11. Central employees</td>
</tr>
<tr>
<td>U4. Distribution of cases over paths</td>
<td>U8. Resources per task</td>
</tr>
<tr>
<td>U2. Routing probabilities</td>
<td>U17. Resource utilization rate</td>
</tr>
<tr>
<td>U5. Exceptions from the normal path</td>
<td>U9. Resources involved in a case</td>
</tr>
<tr>
<td></td>
<td>U16. Arrival rate of cases</td>
</tr>
</tbody>
</table>

3.3.2 Survey

As a next step, we designed and distributed a survey to collect a larger number of responses. The survey
contained all the questions addressed during the interviews, but also additional ones, which serve the
purpose of capturing more detailed information about the end user’s need in terms of process mining
functionality. The complete contents of the survey is given in Appendix C, while Appendix D presents
the results for the complete survey.

This section presents the results obtained for a selection of the questions asked. We focus on the role
and activity domain of the respondents, the ranking of the use cases, the identification of missing use
cases and the possible functionality important for a process mining tool but not covered in the list of use
cases.

From this survey, we received 47 responses. Although this number of responses is not enough to ob-
tain statistically significant results, the survey results can provide useful qualitative feedback to validate
our use cases. The highest percentages of responses we received are from people working in domains
like academia (43%, 20 responses), information technology (21%, 10 responses), business process man-
agement consulting (19%, 9 responses), and banking (6%, 3 responses). The distribution over the roles
shows a high percentage of researchers (51%, 24 responses), followed by process analysts (28%, 13 responses), process managers (9%, 4 responses), and consultants (6%, 3 responses).

The scores obtained by each use case-based on the rankings were computed both over all responses and based on the role of the respondent. The score of a use case is the average of all scores registered from all rankings of the respondents belonging to the same role (the lower the score the more important is the use case). Based on these scores, we generated the graph depicted in Figure 3.4, which presents the profiles of the four most representative roles among the respondents.

Figure 3.4: Use cases ranking results based on respondents roles

Again, the results confirmed our expectation that the way users rank the use cases differs based on the role they have. It is interesting to see that use case 6 (The degree in which rules are obeyed) is considered medium important by researchers, process analysts and process managers while consultants view it as an essential use case. The same observation holds for use case 17 (Resource utilization rates); process managers view it as a highly relevant use case, while the respondents belonging to the other categories have a different opinion.

However, similarities in the ranking are also quite frequent. For instance, use case 1 (Structure of the process) is graded as one of the most important use cases by all the roles. Similarly, use case 3 (Most frequent path in the process) and 7 (Compliance to the explicit process) are present in the tops of all rankings. The lower parts of the four rankings also share common use cases. Examples are use case 11 (Central employees) and use case 16 (Arrival rate of cases).

The rankings obtained for the use cases were also grouped based on the domains of activity of the respondents. The results show few differences between the three domains considered (academia, information technology and business process management consulting). The profiles of the domains are depicted in Figure D.4 in Appendix D.

Table 3.3 presents the results of rankings of the use cases based on the survey responses. We make the distinction between use cases relevant for all the roles, use cases less relevant for all the roles and use cases relevant only for some specific roles. This distinction was made by considering relevant the top
nine use cases from the aggregated rankings of each role and less relevant the remaining ten use cases.

Four use cases (U1, U3, U4, and U7) are considered important by all the groups of respondents, while six use cases (U8, U9, U10, U14, U16, and U18) are rated as less important by all the groups. It is interesting to note that there are two use cases (U13 and U17) that are relevant for only one of the categories of respondents. The opposite holds for use cases U5, U6, U12, and U15, which resulted to be important for three out of the four categories of respondents.

For the question asking whether there are any missing use cases, 58% of the respondents answered no, while 42% suggest new use cases. Among these suggestions, the measurement of different KPIs (cost, quality, flexibility, etc), the creation of a simulation model, and the online analysis of an event log with the purpose of making predictions were mentioned. Since our scope is limited to the process mining techniques that perform an offline analysis of processes and the last two suggestions are related to the online type of analysis, they are not considered for new use cases. The suggestion related to the KPIs measurement does however fit in our scope, but at the moment is too vague and general to be transformed in a testable use case.

The answers regarding the additional functionalities that a process mining system should offer to its users can be grouped into the following categories: input and output capabilities, the ability to filter and cluster data, the integration with external systems like databases, BPM tools, ERP, CRM, etc, animation capabilities, and the support for large input event logs. This information will be used as basis for the extended evaluation of the process mining tools in the following phases of the project.
3.3.3 Conclusions of the Validation Phase

The use cases ranking results derived from the survey are in line with the ones resulted from the interviews, in the sense that respondents having different roles have different needs in terms of process mining functionality. This is reflected in the scores assigned to the use cases. Another similarity between the results of the two validation steps is the fact that use case 1 (Structure of the process) was considered overall the most important one, while use cases 11 (Central employees) and 16 (Arrival rate of cases) are considered to be the least significant ones.

Based on the feedback received during the validation phase of our approach, we removed two irrelevant use cases, we rephrased all the use cases descriptions that were unclear, and we obtained a classification of use cases based on their importance for different roles.

The outcome of the interviews and survey was the validated list of process mining use cases. By validated, we mean use cases properly formulated, understandable, and corresponding to the needs of process mining users. For the practical tool evaluation additional dimensions of the use cases, besides their descriptions, are needed. These dimensions are described in the next section.

3.4 Use Case Framework

Next to the (1) description, given in Section 3.2, a use case consists of a (2) practical example where the use case could be used, the (3) assumptions made, and (4) a set of acceptance criteria. These four elements were defined for each use case and constitute the use case framework. The definition of such a framework is related to the work described in [32] in the context of workflow patterns. In this section, we discuss our use case framework by describing its four elements and by illustrating it with the help of a couple of examples of use cases. A complete detailed description of all use cases is given in Appendix A.

Earlier a use case was defined by one sentence, which aims at capturing the essence of the use case by referring to the action the use case is about. More detailed information about what the use case consists of, as well as about the reasons why it would be executed, are illustrated in the practical example. The assumptions capture any aspects that were assumed with respect to the event logs used to test the use case, while the acceptance criteria represent a set of guidelines and requirements for deciding whether the use case is supported or not by an evaluated system.

One can view this framework as a combination of aspects related to the definition and understandability of each use case, consisting of the description and the practical example, and aspects which correspond to the more technical side of this evaluation, namely the event log assumptions and the acceptance criteria.

In most of the cases, each use case consists of the four elements we mentioned. However, the more complex use cases are split into sub-use cases when we define the list of acceptance criteria. We decided to do so with the purpose of avoiding any confusion or unclarity that might occur if the identified sub-use cases would be treated together as a whole. An example of such a situation is use case 1 (Structure of the process), which was divided into nine independent sub-use cases, with separate acceptance criteria, each of them being focused on exactly one control flow pattern.

In order to illustrate the framework, we present as examples one use case from each of the process mining techniques classes, used for the classification of the use cases in Section 3.2. We choose the use cases that scored the highest from each category, based on the results of the survey. The complete use
3.4. USE CASE FRAMEWORK

The use case framework is given in Appendix A.

Two assumptions hold for every use case. In some cases, additional assumptions are considered for a use case. The two general assumptions are: (1) the process instances logged are representative for the process and (2) a sufficiently large set of possible behavior is observed in order to be able to draw valid conclusions.

Use case 1: Structure of the process

**Description:** Determine the structure of an unknown process or discover how a process looks like in practice.

**Example:** This use case could be used in order to mine the structure of an unknown process. By looking at a discovered process model it is possible to get an overview on how things are really happening in practice. Moreover, the discovered model can be used for communication and redesign purposes.

**Assumptions:** An event log is required for testing the support of the case. The traces in the event log need to contain only information about the name of the activities executed.

**Acceptance criteria:** The set of acceptance criteria for this use case are based on the most common control flow pattern that might occur in a process. The systems will be evaluated whether they incorporate mechanisms able to discover:

1. *Sequential activities* - The tools supporting this sub use case provide a control flow representation that suggests that a set of activities is executed in sequential order, meaning that one activity cannot start before the previous one is completed.
2. *XOR splits and joins* - The tools supporting this sub use case provide a control flow representation that suggests that after the completion of an activity that is considered to be a XOR split, the thread of control is passed to exactly one of the outgoing branches. The convergence of the multiple branches into one subsequent branch is, then, done by executing an activity, considered to be a XOR join.
3. *OR splits and joins* - This sub use case is supported if the system provides a control flow representation that suggests the fact that after the completion of an activity, considered to be the OR split, the thread of control can be passed to one or more of the outgoing branches and is able to discover such a behavior. The convergence of the multiple branches into one subsequent branch is, then, done by executing an activity, considered to be a OR join.
4. *Parallel activities* - The tools supporting this sub use case provide a control flow representation suggesting the divergence of a branch into multiple branches that execute concurrently.
5. *Structured loops - the “repeat” variant* - The tools supporting this sub use case provide a control flow representation suggesting the possibility of executing an activity or a set of activities repeatedly. In particular for this sub use, from the discovered model it should be clear that the activities involved in the loop are executed at least once.
6. *Structured loops - the “do while” variant* - The tools supporting this sub use case provide a control flow representation suggesting the possibility of executing an activity or a set of activities repeatedly. In particular for this sub use, from the discovered model it should be clear that a subset of the activities involved in the loop can be skipped.
7. *Arbitrary loops* - The tools supporting this sub use case provide the ability to discover processes that contain cycles which have more than one entry of exit point.
8. *Skipped activities* - The tools supporting this sub use case have the ability to detect the possibility of skipping a certain activities or set of activity in the discovered process.
9. **Duplicate activities** - The tools supporting this sub use case have the ability to differentiate between two activities having identical names, but being executed in different contexts (e.g., only in the beginning and in the end of the process). The result of this distinction is the representation of the activity in the discovered model as two or more separate (duplicate) activities.

**Use case 7: Compliance to the explicit process**

**Description**: Compare the documented process model with the real process, either given by the event log either given by a explicit model.

**Example**: By executing this use case scenario, the user can detect possible inconsistencies between the real process and the theoretical one, which may then either lead to the enforcement of the documented process or to an update of the documentation.

**Assumptions**: For testing the support of the use case an event log and a model of the explicit process are required. The traces in the event log need to contain only information about the name of the activities executed.

**Acceptance criteria**: The criteria used to decide whether this use case is supported are divided into two categories. The two categories are:

1. **Qualitative criterion**: The tools supporting this pattern provide the possibility of comparing two process models (the discovered one and the explicit one) in a visual fashion and are able to highlight the parts of the process where differences occur.
2. **Quantitative criterion**: The tools supporting this pattern should provide the possibility of measuring the percentage of fit between the discovered process model and the event log, based on predefined metrics.

**Use case 10: Work handovers**

**Description**: Manage resource location or determine possible causes for quality and time issues by looking at how work is transferred between resources.

**Example**: This information can be used to improve the distribution of work or for analyzing quality, time and cost problems.

**Assumptions**: For testing the support of the use case an event log and a model of the explicit process are required. The traces in the event log need to contain information about the name of the activities executed and the identity of the resources performing executing them.

**Acceptance criteria**: A tool that supports this pattern is capable of constructing a social network of interactions between the resources involved in the execution of a process, based on the handovers of work that occur.

**Use case 12: Throughput time of cases**

**Description**: Determine the time that passed since the start of a case in progress until its completion.

**Example**: Statistics related to throughput time serve as a starting point for more in depth performance analysis. Moreover, there are often target throughput times in a company in which a process should be finished. Throughput time analysis allows to verify whether these targets have been met.

**Assumptions**: An event log is required for testing the support of the case. The traces in the event log need to contain information about the name of the activities executed and their start and end timestamps.
Acceptance criteria: One requirement for supporting this use case is to provide the throughput time corresponding to each case in the process. Alternatives include the ability of providing the average throughput time of cases and the minimum and maximum values calculated. Both options are accepted.

Use case 19: Business rules

Description: Discover what are the process attributes that influence the choice points and what are the conditions for following one branch or another.

Motivation: The user can discover which data fields stored in the log determine the choice between different branches in the process. The choices made can be explained by identifying rules which take into account the characteristics of a case.

Assumptions: An event log is required for testing the support of the case. The traces in the event log need to contain information about the name of the activities executed and all attributes that influence the routing decision at choice points.

Acceptance criteria: The tools supporting this pattern should highlight the decisions points and the attributes values for following each of the possible paths.

This chapter focused on presenting the process mining use case framework used to perform the tools evaluation and the approach we followed in developing and validating it. In the next chapter we move to more technical aspects related to the evaluation by presenting the steps undertaken in preparing the experimental setup.
Chapter 4

Experimental Setup

As previously stated, the evaluation of the process mining tools will be based on the use cases defined in the previous chapter. We test the support provided by a tool for a given use case by using as input a particular event log. Additionally, performance aspects are also analyzed using artificially generated logs that can easily be varied in size. The set of artificial event logs and the steps we followed when generating them are presented in Section 4.1. Next to this use case-based evaluation, we also evaluate the behavior exhibited by the tools when dealing with real-life processes running in practice. For this, we use a couple of real-life logs, which are introduced in Section 4.2. We conclude this chapter by presenting the inputs used for each of the systems in Section 4.3.

4.1 Artificial Event Logs

The main reason for creating artificial event logs was to be able to test particular features in a controlled manner. We created example process models with the purpose to test a specific use case and then generated event logs according to these examples. In addition to the artificial event logs for testing the use cases, we also created a couple of artificial event logs that were used to test some performance aspects related to the systems. In this section, we first give the steps we followed from designing a process until getting the final event log, then we present the artificial events log used for the use case-based evaluation, and the ones used for testing the performance issues.

4.1.1 Event Logs Generation

The process of generating an event log to be used for the tool evaluation is depicted in Figure 4.1.

Figure 4.1: The process of generating event logs

The first step was the design of a process model using the coloured Petri nets notation in CPN Tools. Figure 4.2 represents the model built to generate the event log used to test the sequence sub use case of
the first use case. The model consists of three sequential activities: A, B, and C. Each activity consists of events of two types: start and complete. The name, the type, the timestamp, the originator, and any other additional related case data can be logged for each event by using the logging functions available in the CP-net extension described in [21].

Simulating this model for 1000 cases resulted into 1000 separate log fragments stored in files having the .cpnxml extension. Listing 4.1 illustrates two entries in such a .cpnxml file. In the next steps, these process fragments can be aggregated into a single event log by using the CPN tools filter available in the ProM Import Framework. The resulting event log is compliant to the MXML format, whose schema definition is given in [19].

Listing 4.1: .cpnxml file fragment

```xml
1
  <AuditTrailEntry>
    <WorkflowModelElement>A</WorkflowModelElement>
    <EventType>start</EventType>
    <Timestamp>2006-01-01T00:00:00.000+01:00</Timestamp>
    <Originator></Originator>
  </AuditTrailEntry>
...
  <AuditTrailEntry>
    <WorkflowModelElement>C</WorkflowModelElement>
    <EventType>complete</EventType>
    <Timestamp>2006-01-01T00:40:00.000+01:00</Timestamp>
    <Originator></Originator>
  </AuditTrailEntry>
```
Next, the MXML event log was imported into ProM 5.2 with the purpose of converting it into a Comma Separated Values (CSV) file. This is step was required given the fact that the CSV input format is needed for tools not supporting MXML or XES. The conversion was possible by using the CSV for log Exporter functionality. A fragment of the resulting CSV file (Opened in Excel) is depicted in Figure 4.3(a).

![Initial CSV event log](image1.png) ![Final CSV event log](image2.png)

Figure 4.3: CSV event logs

In order to have the event logs in a general format accepted by all the evaluated tools, some transformations were needed. The transformation were executed by defining a set of Microsoft Office Excel macros. Unlike the event type attribute used by ProM to differentiate between event types (start, complete, scheduled, etc), the evaluated tools expect this information in a separate attribute. As a consequence, one of the events corresponding to an activity and also the column specifying the event type were removed. Therefore, each row corresponds to a performed activity with start and end time, rather than an event. However, the information about the start and the end timestamps should not have been lost. As a result of this, we split the initial timestamp column into two columns: a start timestamp corresponding to the start event and an end timestamp corresponding to the complete event. Figure 4.3(b) shows a fragment of what we obtained in the end. An alternative to using Microsoft Office Excel macros in order to make these conversions is the tool Nitro\(^1\).

The same set of steps were performed for all the artificial event logs we generated. The following section gives more information about the benchmark set of artificial event logs that were generated for the use case-based evaluation.

### 4.1.2 Use Case-based Evaluation Event Logs

This section presents the set of events log we generated for the use case-based evaluation. The detailed summaries of each event in the set can be found in Appendix E. The event logs can be found at [location].

Our use case-based evaluation set consists of 15 artificial event logs, generated by following the steps described in the previous section. The number suggests that some of the event logs are used for testing multiple use cases, since the process mining use case framework incorporates 19 use cases. The set contains, however, also event logs designed with the purpose of evaluating one specific use case or sub-use case. This observation holds for all 9 sub use cases belonging to use case 1 (Structure of the process),

\(^1\)http://fluxicon.com/nitro/
4.2. REAL-LIFE EVENT LOGS

which were tested using 9 separate event logs. Each of these 9 event logs corresponds to a simple process model, that only contains the workflow pattern to be tested. The initial idea of testing the support for use case 1, was however different. We first built a more complex process model which incorporated all the control flow patterns that we want to test within the use case. Nevertheless, when applying this approach we realized the difficulty of making sound and accurate conclusions about the ability of discovering each of the nine control flow patterns due to the complexity of the process. As a result, we decided to split the process into separate smaller processes focused only on one pattern.

Some of the event logs in the benchmark set store only data about the name of the activities executed in the process and their timestamp, while other event logs also contain the identity of the resources performing an activity, as well as any additional case attributes that are related to the execution of the activity. The amount of information logged in an event log depends on the purpose it was intended for, namely the process perspectives that the corresponding set of use cases covers. All event logs capture execution traces corresponding to 1000 cases, with the number of events per log ranging between 3000 and 7000.

The purpose of Table 4.1 is to show the relations between the process mining use cases and the artificial event logs.

The relations between the artificial event logs generated for the use case-based evaluation and the use cases are also given in Chapter 5 of the thesis, in which the a selection of the results of the evaluation are presented and in Appendix F, in which the complete results are described.

4.1.3 Other Artificial Event Logs

Next to the event logs used for the evaluation of the tools from the use cases perspective, we generated additional artificial event logs aimed at testing performance related issues. For example, these logs are used to see whether a tool is able to discover parallelism when dealing with highly parallel processes. We also tested the limitations of the systems in terms of the size of the input event logs.

The event log corresponding to the highly parallel process consists of 500,000 events grouped in 50,000 cases. The process itself contains 10 activities which are distributed on two levels of parallelism. The reason for including such an event log in the benchmark set is to test the discovery of parallelism using also a more complicated example, but also show the issues that emerge when the discovery of parallel construct is not supported.

The other performance issue we investigate on is the event log size that a system supports. For this purpose, we used a rather simple and structured process consisting only of sequential and simple choice patterns. We generated three different event logs having 10,000, 100,000 or 1 million cases. The corresponding number of events for each of the three logs are 63,000, 630,000 and 6.3 million. The idea was to test each of the tools with all three event logs, starting with the smallest one, to see which is the supported limit.

4.2 Real-life Event Logs

The main part of the evaluation of the process mining systems is one based on the set of use cases. However, we consider that testing the behavior of the tools when dealing with real-life processes is also an important aspect that should be part of the evaluation, since in practice users need to analyze processes based on real logs rather than synthetic ones. For this purpose, we decided to perform an
informal evaluation of the systems participating in the study by using a set of event logs corresponding to processes running in actual organizations.

The real-life event logs used belong to processes from two different domains, with different structural properties. The reason for selecting these processes and their corresponding event logs consists of two aspects. First, we selected processes from distinct domains, given the fact that the domain of a process has a great influence on its characteristics. The two domains in which the chosen processes run are healthcare and administrative. Second, we also looked at the properties (e.g. size, diversity, length of
cases, etc.) of the event logs and aimed at having various combinations. A characterization of each event log based on a subset of the event log metrics, described in [34], is given in Chapter 6. In this section, we focus on providing a short description of each event log.

The first event log in the set belongs to a process running in a Dutch hospital. The log consists of approximately 150,000 events in more than 1,100 cases, each case representing exactly one patient. This suggests the diversity of the behavior observed. The attributes stored in the log contain information about when the activities took place, what group of resources executed them as well as additional case related aspects.

The third component of the real-life event logs collection originates from a rather simple complaint handling process running in a Dutch municipality [35, 8]. The event log consists of about 9,100 events grouped in 374 cases and stores information about the originators of the activities, the timestamps, and other additional attributes relevant for the cases.

### 4.3 Tool Setup

The CSV version of the event logs, presented in Figure 4.3(b), we obtained at the end of our event log generation process had to be slightly changed and adapted to the input requirements specific to each of the analyzed process mining systems. In this section, we present these requirements.

**ARIS PPM** From the many import options supported by the system (see Chapter 5), we choose to use the CSV files importer. It is required that an input event log contains a case identifier, an activity name and the activity’s start timestamp. The order does not necessarily has to be respected. The first row can contain the column headers, but this is not mandatory. The tool supports as columns separators the tab, the space, the colon, the semicolon or any other symbol specified by the user. However, the same separator needs to be used throughout the entire file.

**Flow** The input format required by Flow is Microsoft Office Excel 2003 (.xls). The compulsory fields that a file needs to contain are, in this order, the event timestamp, the case identifier and the name of the activity. Any other attributes that an event log might contain need to be placed after these three mandatory fields.

**Futura Reflect** For the evaluation of Futura Reflect we used event logs having the CSV format. The minimal fields required for an input file are the case identifier, an activity name and an activity timestamp. The order is not important, as the user can choose the mappings between these attributes and the corresponding columns in the input file while loading it. The first row of the file can optionally contain the column headers. The fields in a row need to be separated by one of the five supported separators: tab, space, comma, colon, and semicolon.

**Process Analyzer** We evaluated the tool using event logs in the Microsoft Office Excel 2007 (.xlsx) format as input. The event data needs to be contained in a sheet called Data in order to have a successful import operation. The mandatory fields that a trace should contain are: an unique case identifier, the name of activity being executed, the organizational unit inside which the activity is performed, the start and the end timestamps of the activity. This is also the strict order in which the attributes should be recorded for each trace. Additional attributes can follow and can be included in the analysis.
4.4 Conclusion

This chapter was aimed at presenting the steps prior to the evaluation of the systems. In this context, we describe the approach we followed for generating the artificial event logs used for the use case-based evaluation and for the testing the performance aspects. Furthermore, we briefly introduced the event logs stemming from real-life processes, that constitute the basis for the discussion in Chapter 6. In the end, we explained what were the input requirements specific to each system. In the next chapter, we present the findings of our tool evaluations.
Chapter 5

Evaluation Results

This chapter presents the main findings of the tool evaluation. We begin by discussing the results for the support the tools provide for several use cases and by giving an overview on the complete results in Section 5.1. The performance aspects are discussed in Section 5.2 while the additional functionalities available in the systems are described in Section 5.3.

5.1 Use Cases-based Evaluations

In this section we discuss the results of the tool evaluations based on the use case framework. However, due to the large number of use cases, we focus our discussion only on a selection of use cases. The complete results are available in Appendix F. For consistency reasons, we selected the same use cases that were used in Chapter 3 to illustrate the evaluation framework. Again, we begin with use case 1 (Structure of the process), more specifically with the XOR splits and Parallel activities sub use cases. Next, we present the results for use case 7 (Compliance to the explicit process), use case 10 (Work handovers), use case 12 (Throughput time of cases), and use case 19 (Business rules). The problems we encountered while evaluating the tools are described in Section 5.1.1. Sections 5.1.2-5.1.6 present the detailed results for the selected subset of use cases. An overview of the results for every use case is given in Section 5.1.9. The detailed explanations of the findings can be found in Appendix F.

5.1.1 Problems Encountered During the Use Case Evaluation

Before presenting the evaluation results some comments need to be made regarding ambiguities in the acceptance criteria of several use cases. Even though we aimed at having clear criteria for deciding the support of a use case, there are some situations in which one could give a broader interpretation to a criterion, which could change the result. This is the case for use case 3 (Most frequent path), use case 5 (Exception from the normal path), use case 10 (Work handovers), and use case 11 (Central employee).

Let us take, for instance, the situation of use case 3. One could easily say that discovering the most frequent path in the process is a small part of use case 4 (Distribution of cases over paths). If a system supports use case 4, then one could simply find the answer for use case 3, by choosing the path followed by the highest percentage of cases. This implies that use case 3 is supported as well. However, the result we expect for use case 3 is that the tool actually highlights the most frequent path in the process model or indicates the sequence of activities that constitutes that path.
The issue about use case 5 is that one could consider that displaying the process model with the percentage of cases displayed on each arc is a sufficient criterion for determining what is the exceptional behavior. In our view, a system supporting this use case, should be able to identify exceptions on its own by selecting the paths with frequencies lower than a threshold.

The comments that might be made about the support of use case 10 is that as long as a system supports process discovery this use case is also supported. The explanation for this is that one could simply swap the activity name and the originator name in the event log. The process discovery algorithm, would then compute a model, with nodes representing originators connected based on the handovers of work between them. We decided, however, that this is not enough for supporting the use case, as it requires the user to manually change the event log.

Finally, for use case 11 there maybe an interpretation problem when the system supports use case 10 and provides, thus, the social network of the originators, based on the work handovers. In this situation, it can be argued that the central employee can be visually identified by the user by looking at the number of incoming and outgoing cases specific to each resource. This is indeed possible when analyzing processes with rather few originators and work handovers. If, however, the number of originators increases and so does the communication between them, the social network becomes unreadable, making the visual identification of the central employee impossible.

The examples mentioned above illustrate that more specific interpretations of the use cases were chosen during analysis. Now, move on to presenting the results for the previously mentioned selection of use cases.

5.1.2 Evaluation of Use Case 1 (Structure of the process)

**Description:** Determine the structure of an unknown process or discover how a process looks like in practice.

**Sub use case 2: XOR splits and joins**

**Acceptance criteria:** The tools supporting this sub use case provide a control flow representation that suggests that after the completion of an activity that is considered to be a XOR split, the thread of control is passed to exactly one of the outgoing branches. The convergence of the multiple branches into one subsequent branch is, then, done by executing an activity, considered to be a XOR join.

**Event log used:** Event log E2 (Table E.2). The event log consists of activity A, followed by the possibility of executing one of the activities B or C, and activity D at the end.

**ARIS PPM** The XOR split pattern can be discovered by both mining approaches implemented in ARIS PPM (EPC and activity sequence). Figure 5.1 shows the EPC process model. Therefore, the system supports the **XOR splits and joins** sub use case.

**Futura Reflect** The XOR splits and joins sub use case is supported by both Mine and Explore process discovery functionalities. Both discovered models (Figures 5.2(a) and 5.2(b)) depict an exclusive choice construct after activity A.

**ProcessAnalyzer** The process model represented in Figure 5.3 contains an exclusive choice pattern after activity A. The **XOR splits and joins** sub use case is thus supported by ProcessAnalyzer.
5.1. USE CASES-BASED EVALUATIONS

Figure 5.1: ARIS PPM supports the discovery of XOR splits and joins

(a) XOR split - Futura Reflect - Mine (b) XOR split - Futura Reflect - Explore

Figure 5.2: Futura Reflect supports the discovery of XOR splits and joins in both miners

Figure 5.3: ProcessAnalyzer supports the discovery of XOR splits and joins
Sub use case 4: Parallel activities

Acceptance criteria: The tools supporting this sub use case provide a control flow representation suggesting the divergence of a branch into multiple branches that execute concurrently.

Event log used: Event log E4 (Table E.4). The event log consists of traces starting with activity A, followed by the sequences B-C or C-B (activities B and C are executed in parallel, therefore), ending with activity D.

**ARIS PPM** The discovery of **Parallel activities** has a **limited support** in ARIS PPM in the EPC representation (Figure 5.4). The limited support is due to the condition that the activities involved in the concurrent construct need to have the same start and completion timestamps. The representation using activity sequences does not support this sub use case.

**Futura Reflect** The **Parallel activities** sub use case is **supported** by **Mine**, but not by **Explore**. The model built by the **Explore** miner (Figure 5.5(b)) shows no parallelism, but a combination of the two traces in the input event log. The algorithm used by **Mine** is, however, able to detect that activities B and C can be performed in parallel (Figure 5.5(a)). The use of the special “+” symbol in the boxes representing activities A and D denote AND split, respectively AND join, semantics.

**ProcessAnalyzer** The model discovered (Figure 5.5) contains no representation that might suggest that the execution of activities B and C is concurrent in practice. Instead, the model covers the two possible paths from the event log (A-B-C-D and A-C-B-D) by displaying the exact two sequences. Moreover, the model allows for additional behavior (e.g. A-C-D, A-B-D, etc), not observed in the log. Therefore, the **Parallel activities** sub use case is **not supported**.

Discovering parallel construct is not an easy task. One of the challenges associated to it is the explosion of possible tasks combination, mentioned in **[8, 4, 36]**. For the use-case based evaluation, we
5.1. USE CASES-BASED EVALUATIONS

(a) Parallelism - Futura Reflect - Mine
(b) Parallelism - Futura Reflect - Explore

Figure 5.5: Futura Reflect supports the discovery of parallel activities in the Mine miner, but not in the Explore miner

![Diagram](image)

Figure 5.6: ProcessAnalyzer does not support the discovery of parallel activities

tested the ability of the tools of discovering parallelism using a quite simple event log with only one level of concurrency involving only two activities. In this situation, even the models that do not contain the parallel construct, are readable. However, in most of the real cases, processes are far from being that simple and the inability of representing concurrency can become an issue with respect to the readability and clarity of a process model.

In the following, we illustrate this problem by showing the models discovered based on the highly parallel process, described in Section 4.1.3. The CPN tool representation of the initial process is depicted in Figure 5.7. The activities in the process have random durations and the executions of the activities involved in a parallel constructs overlap in time.

The models depicted in Figures 5.8 and 5.9 are the results of algorithms unable to detect parallelism. The first model was obtained by using ProcessAnalyzer and the second one is the result of the Explore miner from Futura Reflect. The two models are identical. Opposite to this, we present the model discovered using the Mine functionality in Futura Reflect in Figure 5.10. The model correctly captures the situations in which activities are executed concurrently. The difference between the structure, the complexity, and the understandability of the the two sequential models and the parallel one proves the fact that the inability of discovering and representing parallelism has a negative impact on the readability of a process model.

Due to the random activities durations, the condition required by ARIS PPM for activities to have the same start and end timestamps does not hold. As a result, the system could not reproduce the parallel
Figure 5.7: Representation in CPN tools of a highly parallel process model with random activity durations

constructs present in the initial process. However, we modified the process by introducing constant activity durations values to enforce identical start and completion timestamps for activities. The resulting groups of activities that meet this condition are B-C-D and E-F-I-H. The corresponding event log was mined using ARIS PPM and the result is depicted in Figure 5.11. The discovered model is correct given the groups of activities for which executions completely overlap in time.

Figure 5.8: ProcessAnalyzer is unable to detect the parallel constructs present in the highly parallel process
5.1. USE CASES-BASED EVALUATIONS

Figure 5.9: The Explore miner in Futura Reflect is unable to detect the parallel constructs present in the highly parallel process.

Figure 5.10: The Mine miner in Futura Reflect is able to correctly capture the parallel constructs present in the highly parallel process.
5.1.3 Evaluation of Use Case 7 (Compliance to the explicit process)

**Description:** Compare the documented process model with the real process as observed in the event log.

**Event log used:** Event log E12 (Table E.12). The event log stores for each trace, besides the name of the activities, also the identity of the originators of the activities. The most frequent behavior observed in the event log (97.7%) covers the traces A-B-C-D-H-I-J, A-B-E-F-G-H-I-J, A-B-C-D-H-J, and A-B-E-F-G-H-J. The rest of observed behavior (2.3%) consists of the traces A-B-E-G-F-H-I-J, A-B-D-C-H-I-J, A-B-D-C-H-J, and A-B-E-G-F-H-J.

**Sub use case 1: Qualitative criterion**

**Acceptance criteria:** The tools supporting this pattern provide the possibility of comparing two process models (the discovered one and the explicit one) in a visual fashion and are able to highlight the parts of the process where differences occur.

**ARIS PPM** The Qualitative criterion sub use case is supported by using the visual comparison of two selected processes, which in this case can be the discovered process and the to-be one. Additional features related to the qualitative criterion are available in the ARIS Modeling Tool.

**Futura Reflect** Figure F.40(a) depicts the explicit process model, while figure F.40(b) represents the result of comparing the explicit model with the model discovered from the event log. The results
are presented in a visual fashion, by building a process model that contains the merged elements of the compared models. The arcs of the resulting process model are symbolized using different colors, as follows: light grey arcs are present in the reference model, but not in the discovered model, orange arcs are present in the discovered model, but not in the reference one, and light blue are belong to both models. The result of the comparison performed for the evaluation of this use case contains orange and light blue arcs, but the focus is to check whether indeed the orange arcs only belong to the discovered process model. By looking at the event log 12 and the reference model, depicted in figure F.40(a) one can observe, the possibility of executing the tasks C and D, respectively F and G in any order. This kind of behavior is not conform to the reference model, since there is only one possible order of performing the mentioned activities. The inconsistencies indicated in the comparison of the models are thus valid. One can conclude that the Qualitative criterion is supported.

Figure 5.12: Futura Reflect compares two process models by merging them into one model and highlighting the arcs constituting differences

**ProcessAnalyzer** The tool provides no mechanism of visually comparing the discovered process model with the reference one. Therefore, the Qualitative criterion is not supported.
Sub use case 2: Quantitative criterion

**Acceptance criteria:** The tools supporting this pattern should provide the possibility of measuring the percentage of fit between the discovered process model and the event log, based on predefined metrics.

**Results:** None of the analyzed tools provides a mechanism of comparing a reference model to the discovered one by measuring the percentage of fit between them. The **Quantitative criterion** is thus not supported.

5.1.4 Evaluation of Use case 10 (Work handovers)

**Description:** Manage resource location or determine possible causes for quality and time issues by looking at how work is transferred between resources.

**Event log used:** Event log E13 (Table E.13). The event log stores for each trace, besides the name of the activities, also the identity of the originators of the activities and the activities timestamps. The possible traces are: A-B-C-D-H-I-J, A-B-E-F-G-H-I-J, A-B-C-D-H-J, and A-B-E-F-G-H-J.

**Acceptance criteria:** A tool that supports this pattern is capable of constructing a social network of interactions between the resources involved in the execution of a process, based on the handovers of work that occur.

**ARIS PPM** The system is capable of constructing the social network of resources based on the handovers of work that take place using the activity sequence representation. On the arrows connecting the nodes (which represent the resources), the number of transfers of work is displayed (Figure 5.13(a)). Moreover, the user also has the possibility to aggregate several nodes and display the resulting network (Figure 5.13(b)). One can thus conclude that the **Work handovers use case is supported.**

![Social Network](image1.png)

(a) Social Network  
![Aggregated Social Network](image2.png)  
(b) Aggregated Social Network

Figure 5.13: ARIS PPM displays the handovers of work between resources using the social network representation
Futura Reflect In the Explore functionality, in the “Options” pane, the “Attribute” should be set on “Originator” and the layout as “Social Network”. The result is a resource social network based on the handovers of work, depicted in Figure 5.14. The thickness of the arcs is given by the number of transfers between two resources. Therefore, one can conclude that the Work handovers use case is supported in the Explore functionality.

Figure 5.14: Futura Reflect displays the handovers of work between resources using the social network representation.

ProcessAnalyzer The handovers of work between organizational units are visible by looking at the arrows connecting activities in the swimlane representation (Figure F.49). However, this representation brings the limitation that any activity, even the ones executed by multiple resources, can only be present in one swimlane, causing some of the swimlanes to be displayed as empty. Therefore, the information about the complete set of resources that can undertake it is lost. The conclusion is that the Work handovers use case is not supported by ProcessAnalyzer.

5.1.5 Evaluation of Use case 12 (Throughput time of cases)

Description: Determine the time that passed since the start of a case in process until its completion.

Event log used: Event log E13 (Table E.13). The event log stores for each trace, besides the name of the activities, also the identity of the originators of the activities and the activities timestamps. The possible traces are: A-B-C-D-H-I-J, A-B-E-F-G-H-I-J, A-B-C-D-H-J, and A-B-E-F-G-H-J.

Acceptance criteria: The minimum requirement for supporting this use case is to provide the throughput time corresponding to each case in the process. Alternatives include the ability of providing the average throughput time of cases and the minimum and maximum values calculated. Both options are accepted.

ARIS PPM The throughput time of cases is represented by the KPI called “Process cycle time”. One can choose to display in the process instances table, next to the identifier of each process instance, also its throughput time (Figure 5.16). The use case Throughput time of cases is therefore supported.

Futura Reflect Using the Overview functionality, in the “Attributes/Metrics” tab, one can add attributes or metrics to the process instances in the event log. The user can select from a range of predefined...
CHAPTER 5. EVALUATION RESULTS

Figure 5.15: ProcessAnalyzer does not display the complete information about the handovers of work due to the limitations of the flowchart representation (e.g. empty swimlanes)

Figure 5.16: ARIS PPM displays the throughput time specific to each case in a table

attributes. By choosing the metric “Simple throughput time” from the “Throughput time” menu in the prompted window, information about the time passed since the beginning of a case until its completion is attached to it in the dataset. After this operation, the newly added metric and its minimum and maximum values are visible in the table containing all the attributes and metrics of the process (Figure 5.17). This suggests that the Throughput time of cases use case is supported.

Figure 5.17: Futura Reflect displays the throughput time specific to each case in a table

**ProcessAnalyzer** In the “Cases” sheet it is possible to see for each case in the event log the duration in hours, days or month (Figure 5.18). By sorting one of the duration columns, one can also see the minimum and the maximum value of the throughput time and the corresponding cases. Therefore, the Throughput time of cases use case is supported.
5.1. USE CASES-BASED EVALUATIONS

5.1.6 Evaluation of Use case 19 (Business rules)

**Description:** Discover what are the process attributes that influence the choice points and what are the conditions for following one branch or another.

**Event log used:** Event log E15 (Table E.15). The event log stores for each trace, besides the name of the activities, also the identity of the originators of the activities, the activities timestamps and two additional attributes (“Value damage” and “History”). The possible traces are A-B-C-F-G-H, A-B-C-F-G-I, A-B-C-D-E-G-I, and A-B-C-D-E-G-H. The “Value damage” attribute influences the decision taken after activity D (values lower than 5000 follow D and values higher than 5000 follow F), while the “History” attribute influences the decision taken after activity G (values lower than 15000 follow H and values higher than 15000 follow I).

**Acceptance criteria:** The tools supporting this pattern should highlight for each choice point in the process what are the event log attributes that are involved in taking the decision and what values correspond to following the possible paths.

**Note:** The acceptance criterion was initially stating the need of having as a result of this use case a decision tree, built using data mining functionalities. However, none of the evaluated tools provides such a feature. Nevertheless, two of the tools (ARIS PPM and Futura Reflect) tackle this type of analysis using a simpler and less powerful approach. We adapted the acceptance criteria accordingly and we consider the use case supported.

**ARIS PPM** Using the activity sequence of a process model, one can display different types of information on the arcs. Figure F.68(a) shows the average values of the “Value damage” attribute, while Figure F.68(b) shows the average value of the “History” attribute. If we look at the choice points C, respectively G, and at the large difference between the values of the corresponding attribute (“Value damage” in case of C and “History” in case of G), we can tell that the attribute has an influence on following one path or the other, as well as the range of values specific to each path. Therefore, the Business rules use case is supported.

**Futura Reflect** In the Explore functionality, there is the possibility of displaying the average value of an attribute under each activity in the model discovered. Figure F.69(a) depicts the model with the average values of the “Value damage” attribute displayed under the activity boxes, while in Figure F.69(b) one can see the average values of the “History” attribute. By looking at the average values of the two attributes at the choice points activities (C and G) and then at the values at the following activities (F and D, respectively H and I), one can get an idea related to the attribute influencing the choice, as well as the average of the values specific to subsequent each path. The conclusion is that the Business rules use case is supported.

**ProcessAnalyzer** There is no possibility of visualizing what are the attributes and the attribute values that influence a choice point. It can, therefore, be concluded that the use case Business rule is not supported.
CHAPTER 5. EVALUATION RESULTS

(a) Average value of “Value damage” attribute - ARIS PPM
(b) Average value of “History” attribute - ARIS PPM

Figure 5.19: ARIS PPM displays the average value of an attribute for all cases following an arc. A high difference between the values of the attribute on two arcs following a choice point suggest that the routing decision is based on the attribute.

(a) Average value of “Value damage” attribute - Futura Reflect
(b) Average value of “History” attribute - Futura Reflect

Figure 5.20: Futura Reflect displays the average value of an attribute for all cases following an arc in the Explore miner. A high difference between the values of the attribute on two arcs following a choice point suggest that the routing decision is based on the attribute.
5.1. USE CASES-BASED EVALUATIONS

5.1.7 Problems Caused by the Inability of Discovering Parallelism

The inability of distinguishing between sequential and concurrent behavior might have an effect on the correctness of the results given for different types of analysis. From the complete set of use cases, we identified several use cases, which can be influenced by the support a system provides for the detection of parallelism. These use cases are: use case 2 (Routing probabilities), use case 3 (Most frequent path), use case 4 (Distribution of cases over paths), use case 5 (Exception from the normal path), use case 7 (Compliance to the explicit process), use case 10 (Handovers of work), use case 14 (Longest waiting times), use case 15 (Cycles), and use case 19 (Business rules). Next, we discuss the reasoning for selecting them.

We illustrate the influence of the inability of discovering parallelism on use case 2 (Routing probabilities) by considering the process depicted in Figure 5.21. The process consists of a AND split pattern involving activities B and C and a XOR split pattern following activity B. After the XOR split 70% of the cases are routed to activity D and 30% to activity E. The result of applying a process discovery algorithm unable to detect parallel construct on an event log corresponding to the described process would be equivalent with the process represented in Figure 5.22. This model introduces a fake XOR split construct after activity A. As a consequence of this additional choice point, the values of the routing probabilities after activity B are smaller than the ones in the original situation.

Figure 5.21: Process model depicting a parallel construct (B and C) and a XOR split after activity B

Figure 5.22: Model of the process in Figure 5.21 discovered by an algorithm unable to detect parallelism
CHAPTER 5. EVALUATION RESULTS

The impact that the inability of discovering concurrent activities has on the correctness of the results of the analysis described in use cases 3, 4 and 5 are similar and are based on the situation in which a parallel construct would be contained by one of the paths included in the analysis. Let us assume the process model depicted in Figure 5.23. After activity A, there is the choice of continuing by executing activity B (55% chance) or by executing activity F (45% chance). Given this situation, the correct answer for the most frequent path, would be the path highlighted in the picture. If, however, the concurrent execution of activities C and D is not detected, the 55% cases following that branch, would be split after executing activity B. A system unable to discover parallelism, would see this as two different paths (A-B-C-E-H and A-B-D-E-H) with lower percentages of cases that the path A-F-G-H. The result of the most frequent path computation would be, thus, the path A-F-G-H, which does not constitute the correct answer. The same type of reasoning holds for including use case 4 (Distribution of cases over paths) and use case 5 (Exceptions from the normal path) in the set of use cases affected by parallelism.

![Figure 5.23: The computation of the most frequent path in the process is affected when the process contains parallel constructs which are not discovered](image)

In order to illustrate the problems caused by the impossibility of discovering the parallel execution of activities with respect to use case 7 (Compliance to the explicit process), let us imagine the situation depicted in Figure 5.24. The left process is the original process, with activities B and C involved in a parallel construct. The process on the right, is the model discovered by an algorithm unable to detect parallelism. By visually comparing the two models, we can observe two major differences. The first difference is the transformation of the AND split construct represented by activity A in the original model into an XOR split pattern in the right model, while the second difference concerns the transformation of the AND join pattern denoted by activity D into an XOR join pattern. The problems caused by the inability of detecting parallelism would affect, thus, the comparison between the explicit process model and the discovered one.

For the discussion regarding use case 10 (Handovers of work), we consider the same initial process and its corresponding model discovered (Figure 5.24). We enhance the initial model with information about the identity of resources performing the tasks (left side of Figure 5.25). Based on this, the correct social network is the one depicted in the upper right side of Figure 5.25. However, an algorithm unable to detect the parallel construct involving activities B and C, would most likely construct the social network represented in the lower right side of the same figure. The second social network suggests that work is being transferred between Sam (originator of activity B) and Mary (originator of activity C), but in the real situation this is never the case, as the two activities are being executed independently.

We use the same simple parallel process and its corresponding discovered model (Figure 5.26), as before, to explain the problems that occur then computing the waiting time before an activity. Let us consider the end time of activity B (endB), the end time of activity C (endC), and the start time of activity D (startD). For both models, we are interested in finding the expression for the waiting time
before activity D, based on these three time values. In case of the process containing the parallel pattern, the waiting time would be the maximum of the difference between startD and endB, on one hand, and of the difference between startD and endC, on the other hand. In the other process, only one of the branches can be executed in a case. The waiting time would be, therefore, the average between the waiting times corresponding to each branch (startD-endB and startD-endC). Since the two expressions are not equivalent, we can argue that the inability of detecting parallelism causes incorrect results for the waiting time values.
Figure 5.26: The value of the waiting time before an activity representing an AND join construct is affected by the inability of detecting parallelism

The analysis corresponding to use case 15 (Cycles) is aimed at checking whether the large time delays specific to some process instances are caused by cycles. In order to do this, one would first need to build the process model corresponding to the cases having the highest throughput times. In the situation in which, a parallel construct would be part of such a cycle, the result provided by a system unable to detect concurrency for this use case would be inaccurate.

For illustrating the influence that the inability of detecting parallelism has on the result of use case 19 (Business rules), let us consider again the model depicted in Figure 5.23. Additionally, let us assume that the decision of following activity D or activity C after executing B, in the original process, is based on the value of an attribute. In this situation, the result of use case 19 would consist of the name of the attribute and the average values specific to each of the two branches. If, however, the parallel branching involving activities C and D would not be detected and instead a false XOR split construct would be introduced, then the algorithm performing the decision mining might derive incorrect rules, which are never applied in practice.

5.1.8 Evaluation of Flow

The reason for not including the results of the use case-based evaluation of Flow in our comparison was the inability of performing the evaluation. Due to some malfunctioning problems that occurred when testing the support of the use cases, we were not able to conclude this step.

The major problem we encountered was an error in the process discovery functionality, which affected the correctness of the analysis results corresponding to the other use cases. For illustration purposes, we present the result obtained when mining the sequential event log (Event log E1 - Table E.1). Although,
5.1. USE CASES-BASED EVALUATIONS

the event log consists only of A-B-C traces, the model depicted in Figure 5.27 displays an arc between activity A and activity C. This constitutes an incorrect result and the problem causing it is propagating and affecting the rest of the use cases, which are implemented in Flow.

Figure 5.27: The model discovered by Flow for the sequential process displays an incorrect arc between activities A and C

5.1.9 Overview of Use Case-based Evaluation Results

The overview of the evaluation results for the complete set of use cases is given in Table 5.1. Two symbols are used to represent the support of the system for a use case: “+” means that the use case is supported, and “-” symbolizes the opposite. For the use cases that consist of several sub use cases (e.g. use case 1) we present the results at the sub use case level. Another thing that needs to be mentioned is that there are two process discovery features in ARIS PPM (Activity sequence and EPC) and Futura Reflect (Mine and Explore). In order to provide a clear overview on the capabilities of the tools, we make the distinction between the two features when presenting the support of the use cases. However, for the situation in which the support of a use case is independent of the two features (e.g. some use cases belonging to the time perspective), the result is given at the system level. Next to the “+” and “-” symbols, we also use the “(+)” symbol to represent the weak support that the EPC representation in ARIS PPM provides for the detecting parallelism, due to the assumption that the concurrent activities need to have identical start and completion timestamps.

The process discovery class of use cases (use cases 1-4) has a rather good support in all the three systems. The exception is use case 3 (Most frequent path), which is only supported by one of the functionalities available in Futura Reflect. The most challenging control-flow patterns for the discovery algorithms turned out to be the parallel activities (supported only by two mining functionalities), OR splits and joins and duplicate activities (both of them not supported at all).
Table 5.1: Result Overview of the Use Case-based Evaluation

<table>
<thead>
<tr>
<th>Use case</th>
<th>Acceptance Criteria</th>
<th>ARIS PPM</th>
<th>Futura Reflect</th>
<th>Process Analyzer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>A.s.</td>
<td>EPC</td>
<td>Mine</td>
</tr>
<tr>
<td>U1. Structure of the process</td>
<td>Discover sequence</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td>Discover XOR splits and joins</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td>Discover OR splits and joins</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Discover parallel activities</td>
<td>-</td>
<td>(+)</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td>Discover structured loop - Repeat</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td>Discover structured loop - Do while</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td>Discover arbitrary loop</td>
<td>-</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td>Discover skipped activities</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td>Discover duplicate activities</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>U2. Routing probabilities</td>
<td>Assign probabilities to arcs</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>U3. Most frequent path in the process</td>
<td>Show most frequent path</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>U4. Distribution of cases over paths</td>
<td>Show distribution of cases over paths</td>
<td>-</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>U5. Exceptions from the normal path</td>
<td>Show exceptional cases or paths</td>
<td>+</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>U6. The degree in which the rules are obeyed</td>
<td>First level - hard coded rules</td>
<td>+</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Second level - Generic rules</td>
<td>+</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>U7. Compliance to the explicit process</td>
<td>Qualitative criterion</td>
<td>+</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td>Quantitative criterion</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>U8. Resources per task</td>
<td>Show associations resource-task</td>
<td>+</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>U9. Resources involved in a case</td>
<td>Show set of resources involved in a case</td>
<td>+</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>U10. Work handovers</td>
<td>Show social network</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>U11. Central employees</td>
<td>Show central employees</td>
<td>+</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>U12. Throughput time of cases</td>
<td>Show throughput time per case</td>
<td>+</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>U13. Slowest activities</td>
<td>Show duration of activities</td>
<td>+</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>U14. Longest waiting times</td>
<td>Show the waiting times before activities</td>
<td>+</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>U15. Cycles</td>
<td>Show cycles causing time delays</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>U16. Arrival rate of cases</td>
<td>Show arrival rate pattern</td>
<td>+</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>U17. Resource utilization rate</td>
<td>Show resource utilization</td>
<td>+</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>U18. Time sequence of events</td>
<td>Show events in a case in a time sequence</td>
<td>+</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>U19. Business rules</td>
<td>Show attribute values at choice points</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Most of the use cases from the conformance checking category (use cases 5-7) are available in two of the evaluated systems (ARIS PPM and Futura Reflect). The only exception consists of the Quantitative criterion of use case 7 (Compliance to the explicit process), which is not present in any of the tools.

The use cases analyzing the resource (uses cases 8-11) and the time perspective (use cases 12-18) of a process are supported in a great extent. Use case 11 (Central employee) and use case 16 (Arrival rate of cases) constitute the exceptions here, as they are only available in ARIS PPM. The analysis involving the case perspective of the process, represented by use case 19 (Business rules), is supported by two systems (ARIS PPM and Futura Reflect).

Traditionally, ARIS PPM has been focusing on analyzing the performance aspects of the process. This fact is also proved by the good support that the tool provides for the use cases oriented on the resource and the time perspective analysis. At the system level, ARIS PPM offers full support for the organizational (use cases 8-11) and the time (use cases 12-18) analysis. The use cases belonging to the conformance checking are also available in the tool in a large extent (4 out 5 sub use cases). The results obtained for the sub use cases corresponding to the process discovery class differ in one aspect between the two mining functionalities available in the system. The mining approach using the activity sequence representation is only able of detecting basic control-flow patterns (therefore, no OR splits and joins, parallel constructs and duplicate activities), while the approach using the EPC representation enables the discovery of parallelism, provided that the activities executed concurrently have identical start and completion timestamps. The tool provides no information with respect to the frequency of the different paths in the process (use cases 3 and 4). To sum up, ARIS PPM is a very suitable option for analyzing any performance related aspects of a process, given the large number of use cases supported from this category. It has, however, the disadvantage of requiring a complex and rather long set up step.

Futura Reflect supports use cases from the complete process mining spectrum. The algorithm behind the Mine functionality is capable of discovering parallel constructs, besides the basic control-flow patterns. The Explore functionality is, however, less powerful in the process discovery part (it supports only the discovery of the basic control-flow patterns), but provides a good insight into the time perspective of the process, by enhancing the discovered model with information about activity durations and waiting times. An important aspect to be noted with respect to the control-flow perspective is that Futura Reflect is the only evaluated system able to highlight the most frequent path at the process level (use case 3). The conformance checking and the organizational analysis classes have both a good support in Futura Reflect (4 out of 5 sub use cases, respectively 3 out of 4 use cases are available). The time performance analysis use cases are either present in the models discovered by the Explore miner, either independent of the two mining functionalities, such as use case 12 (Throughput time of cases) and use case 17 (Resource utilization rate). In conclusion, Futura Reflect provides support for a large number of use cases and sub use cases at the system level. The scores at the functionality level, however, suggest that the Mine miner is intended for discovering complex process models, but provides little information about the additional perspectives of the process. The opposite holds for the Explore functionality, which scores better than Mine, when it comes to analyzing the organizational, the time and the case perspective. Futura Reflect is a tool that covers, thus, the complete spectrum of process mining techniques, and has the advantage of not needing any setting up, due to its web-based nature.

ProcessAnalyzer provides support for discovering the basic control-flow patterns, but implements no conformance checking techniques. The support for the analysis of the organizational perspective, features the limitation brought by the process representation using swimlanes flowcharts, which makes the analysis at resource level impossible. This is also reflected in the weak support for the use cases from this category, only use case 9 (Resources involved in a case) being present. Nevertheless, the spectrum
of time perspective analysis techniques has a good support in ProcessAnalyzer (6 out of 7 use cases are available). As a general remark, ProcessAnalyzer is a less powerful system compared with the other two judging by the number of supported use cases. However, it has the advantage of being easy to set up, due to its integration with Microsoft Office Excel, and easy to use. ProcessAnalyzer is a suitable option for analyzing processes consisting of basic constructs and for getting insights into the performance aspects related to the time perspective.

As a last remark with respect to the results of the use case-based evaluation, we would to mention the fact that process mining users should weight the use cases based on their needs and opinions on what is relevant in practice. When deciding what process mining system to use for a particular task, one should first analyze the support for the use cases that will be applied.

5.2 Performance Aspects

Next to the evaluation based on the use cases, we also investigated the performance aspects of the tools. For this purpose, we used three events log, originating from the same process, with different size ranges (10,000, 100,000 and 1,000,000 cases). We measured the time it takes to load each event log into a system, as well as time needed to discover the process model from it.

Unfortunately, no sound comparison can be made between the results of the tools, due to the fact that we could not run them on the same platform. The evaluation of ARIS PPM was done remotely through live sessions, with the system running on a dedicated server at the vendor’s site and Futura Reflect is a web-based tool. Tables 5.2-5.4 present the time values specific to each system for the load and mining times of the three event logs, as well as the hardware specifications of the machines used. In the case of ProcessAnalyzer, due to the way data is imported (using Microsoft Office Excel sheets), the testing using the largest event log (6.3 million events) was not possible.

Table 5.2: Performance Aspects - ARIS PPM

<table>
<thead>
<tr>
<th>Hardware specifications</th>
<th>Number of events</th>
<th>Load time</th>
<th>Aggregation time (EPC)</th>
<th>Aggregation time (activity sequence)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>63,000</td>
<td>00:01:28</td>
<td>00:00:09</td>
<td>00:00:01</td>
</tr>
<tr>
<td></td>
<td>630,000</td>
<td>00:08:05</td>
<td>00:02:20</td>
<td>00:00:01</td>
</tr>
<tr>
<td>Memory</td>
<td>8GB RAM</td>
<td>01:24:52</td>
<td>00:00:17</td>
<td>00:00:04</td>
</tr>
</tbody>
</table>

Table 5.3: Performance Aspects - Futura Reflect

<table>
<thead>
<tr>
<th>Hardware specifications</th>
<th>Number of events</th>
<th>Load time</th>
<th>Mine mining time</th>
<th>Explore mining time</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>63,000</td>
<td>00:00:45</td>
<td>00:00:02</td>
<td>00:00:01</td>
</tr>
<tr>
<td></td>
<td>630,000</td>
<td>00:07:30</td>
<td>00:00:05</td>
<td>00:00:03</td>
</tr>
<tr>
<td>Memory</td>
<td>1GB RAM</td>
<td>01:10:12</td>
<td>-</td>
<td>00:00:10</td>
</tr>
</tbody>
</table>
Table 5.4: Performance Aspects - ProcessAnalyzer

<table>
<thead>
<tr>
<th>Hardware specifications</th>
<th>Number of events</th>
<th>Load time</th>
<th>Mining time</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>63,000</td>
<td>00:06:09</td>
<td>00:01:45</td>
</tr>
<tr>
<td>Intel Core2 Duo T9600</td>
<td>2.8GHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Memory</td>
<td>630,000</td>
<td>01:10:53</td>
<td>00:22:34</td>
</tr>
<tr>
<td>1GB RAM</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5.3 Additional Functionalities

In this section we present several additional functionalities that the evaluated systems provide. These functionalities were not covered by the use cases, but appeared to be important information for process mining users, according to the results of the interviews and of the survey. We decided, thus, to describe the capabilities of the tools in terms of import and export, preprocessing, integration with external systems, and animation.

5.3.1 Import Capabilities

**ARIS PPM** The input format supported by ARIS PPM is a specific type of XML, called event/graph format. The tool provides conversions mechanisms to this format for data extracted from CSV files, JDBC databases, and SAP systems.

**Futura Reflect** The tool is able to import data from CSV files, as well as Oracle and SQL databases systems.

**ProcessAnalyzer** Two import options are available for ProcessAnalyzer. The first is to load event logs from Microsoft Office Excel files (.xlsx). The second one is to connect to a database and extract the event log.

5.3.2 Export Capabilities

**ARIS PPM** The discovered EPC models can be exported as XML and AML files, image files (e.g. JPG, GIF, BMP, PNG, SVG) and PDFs. Charts can be exported as image files, while tables can be saved in XML, CSV, Excel (.xlsx), Minitab (statistics tool), RTF, HTML, and PDF files. Models can be, furthermore, edited.

**Futura Reflect** Mined models can be exported as: Adobe Flash file, BPM\one files and Visio Drawings. Charts can be exported as Microsoft Office Excel files and PNG image files. The BPM\one models can be, furthermore, edited.

**ProcessAnalyzer** The discovered process models can be exported in the XPDL format. However, unlike ARIS PPM and Futura Reflect, no editor is provided.

5.3.3 Preprocessing Capabilities

**ARIS PPM** Filters can be defined based any defined KPI and any dimension. All data is stored on instance level and linked to the specific process instances, enabling the possibility of setting any filter combination (e.g. “Select all process instances where the overall process takes between 25 and 45 minutes, more than 3 organizational units are involved and that have been started in January
2011”). In the user’s interface, “interactive filtering” can be done when, for instance, multiple charts/EPCs/views are displayed and clicking on one of them the corresponding filter would be set in the other diagrams.

**Futura Reflect** The tool provides the possibility of defining, saving, removing, and modifying filters. There are four types of filters: attribute, unique paths, incomplete cases, and sequence. The attribute filter filters the event log based on any of its attributes. By using the unique path filter it is possible to filter out exceptional cases that represent deviations from the normal behavior. The incomplete cases filter enables the user to define the possible start and end activities of a process and filter only those cases that start and end with the specified activities. With the sequence filter it is possible to filter the process instances that contain a sequence of two subsequent activities specified by the user.

Additionally, one can also create filters by clicking activity boxes or arcs in the discovered model by the Explore miner. Clicking an activity is equivalent with defining a attribute filter based on the activity name, while clicking on a arc connecting two activities is equivalent with defining a sequence filter.

**ProcessAnalyzer** Data filtering can be done in two main ways in Process Analyzer. First of all, it is possible to use the filtering features available in Microsoft Office Excel. Secondly, there is the possibility of applying filters on the event log by clicking on the activity boxes and the arcs in the discovered model. For a selected activity, for instance, one has the options of including in or excluding from the analysis all process instances containing that activity. The same possibilities are available when an arc is selected.

### 5.3.4 Integration with Other Systems

**ARIS PPM** The tool can be integrated with the SAP ERP system or with databases systems by means of adapters, with the purpose of extracting process related data.

**Futura Reflect** The tool can run both as a stand-alone system (setup used throughout the evaluation) and as an integrated part of the BPM|one platform of Pallas Athena. Integration with databases systems is also possible for extracting event logs.

**ProcessAnalyzer** The system is by default integrated in Microsoft Office Excel. Additionally, ProcessAnalyzer can be connected to databases systems for log extraction purposes.

### 5.3.5 Animation Capabilities

**ARIS PPM** No animation capabilities are available in ARIS Process Performance Manager 5.

**Futura Reflect** The Animate functionality enables the animation of process models by taking into account the timestamp information available in the event log. Animation is possible only after a model has been discovered, using the Mine or Explore functionality, and saved. One can set parameters like the duration of the animation and the process attribute the animation is focused on. The animation can be saved as Flash movies (.swf).

**ProcessAnalyzer** Through the Animate capability users can visualize the cases moving through the process. One can choose the start and the end date and time for the animation.
5.4 Conclusion

The main purpose of this chapter was to present the results of the use case-based evaluation. We explained the findings in details for a couple of use cases and we summarized the results for the complete set of use cases. In this context, we also pointed out a series of problems that emerge due to the inability of detecting parallel constructs and we indicated several use cases for which the acceptance criteria are still subject to interpretation. Next, we presented the import and mining time values specific for each tool, based on a set of three event logs with different ranges of size. In the end, several additional functionalities implemented in the systems (e.g. import, export formats, preprocessing, etc) are described. The next chapter is focused on discussing the functionalities of the tools that can be used to analyze more complicated event logs, originating from real-life processes.
Chapter 6

Discussion

In Chapter 5, we evaluated the process mining tools in a controlled and structured way using artificial event logs. In this chapter we present the outcome of analyzing two real-life event logs using the process mining systems included in this study. The first event log originates from a healthcare process, while the second one corresponds to a complaint handling process running in a Dutch municipality. Both event logs were introduced and briefly described in Section 4.2.

The focus of this chapter is, thus, to discuss the ability of the tools to deal with event logs stemming from real processes. Before discussing our findings, we position the event logs used in this context, based on a couple of log metrics (Section 6.1). We present the results of the analysis of the healthcare process in Section 6.2 and the analysis of the administrative one in Section 6.3. Unfortunately, due to some limitations we faced, we could not perform the analysis of a process using all four tools. The healthcare process was investigated using ARIS PPM and Futura Reflect only, as ProcessAnalyzer produced a “System out of memory” exception when mining the event log. For the administrative process, we could not test the capabilities of ARIS PPM, due to the evaluation setting. None of the real-life event logs could be analyzed using Flow, due to problem described in Section 5.1.8 and to the limitation in the number of lines in the Microsoft Office Excel 2003 file format required by the tool (e.g. the hospital event log has approximately 150,000 events).

6.1 Overview on Structural Log Metrics

As part of the work done in [34], a set of structural log metrics are introduced. The idea of defining such a set came from the fact that event logs, especially the ones originating from real-life processes, are structured very differently. The metrics can be computed based only on the characteristics of the event log and are independent of the domain knowledge specific to the process. The metrics can be used to characterize event logs, but can also represent a means of selecting the most suitable process mining algorithms to be used given an event log.

Out of the eight defined structural log metrics, we focus on three of them: magnitude, support, and level of detail. We briefly describe these three metrics and then based on them we position the two event logs used in this chapter.

Magnitude represents the size of an event log in terms of the number of events it contains. Event logs featuring a large magnitude constitute a challenge for both the analysis environment and mining algorithms, which most of the times scale linearly with the log magnitude.
6.2. HOSPITAL EVENT LOG

Support is the metric that describes the number of independent observations that are stored in the event log, i.e., the support of a log refers to the number of cases logged. From the point of view of the mining algorithms, a high level of supported is desired. However, in real situations, this is not always the case. The influence that this has on the quality and the correctness of the results of a mining algorithm should, thus, be taken into consideration.

The level of detail of an event log captures the average number of distinct events per trace. Event logs characterized by a high level of detail increase the complexity of the process model discovered by the mining algorithms, which subsequently has a negative impact on the readability of the diagram.

Table 6.1 presents the values of the three selected structural log metrics for the hospital and the municipality event logs. Compared to the municipality log, the hospital event log is characterized by large values for the magnitude and level of detail. The municipality log contains 374 cases, while the hospital one 1,143, information which is given by the support values.

Table 6.1: Structural log metrics

<table>
<thead>
<tr>
<th>Event Log</th>
<th>Magnitude</th>
<th>Support</th>
<th>Level of Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hospital</td>
<td>150,291</td>
<td>1,143</td>
<td>113.12</td>
</tr>
<tr>
<td>Municipality</td>
<td>9,174</td>
<td>374</td>
<td>3.59</td>
</tr>
</tbody>
</table>

6.2 Hospital Event Log

The hospital event log originates from a Dutch Academic Hospital. As other healthcare processes, the process that the event log originates from, is highly unstructured, flexible, and depicts a large amount of different behavior suggested by the 981 possible paths. Given this situation, discovering a process model that covers all the observed behavior and then being able to analyze it, is not feasible. However, the process can be analyzed from different perspectives and one can gain insights into it, by applying various types of process mining techniques. Our purpose is, therefore, to discover what information about the process can be found out using ARIS PPM and Futura Reflect.

As previously stated, analyzing the complete behavior observed in the event log is not a realistic expectation, since the resulting process would be a “spaghetti process”. However, one can get a grasp on the control flow perspective of the process, by analyzing the process model corresponding to the most frequent behavior.

ARIS PPM provides the possibility of filtering the less frequent arcs connecting the activities in the activity sequence representation. This is done based on a threshold set by the user. Figure 6.1 depicts the activity sequence obtained by filtering out all the arcs with a frequency less than 1,000. In the case in which all arcs connected to an activity are removed, the activity is removed as well from the diagram, which reduces the models complexity. If, however, a loose activity is kept in the model (e.g. “vervolgconsult poliklinisch”), it is because the activity is involved in a self loop. The approach used for reducing the complexity of the process is, thus, based on removing individual arcs based on frequency. This implies the fact that the complete flow of cases is not preserved, as only the most dominant following relations become visible.

In Futura Reflect filtering out the infrequent behavior can be done by using the “Unique paths” filter, with the activity name as the attribute used to compute the paths. Figure 6.2 represents the model discovered by the Explore miner based on the most 20% frequent cases. The approach used by Futura
Reflect is, therefore, a case-based one. However, for situations in which all cases are unique, the cases included in the analysis would be randomly selected. The quality of the simplified model in such complex situations cannot be, thus, trusted.

The different approaches employed by ARIS PPM and Futura Reflect for reducing the complexity of a process model are also reflected in the resulting models. The two simplified processes (Figures 6.1 and 6.2) consist of completely different set of activities.

Both systems are capable of discovering the social network of resources based on the handovers of work. In this healthcare process there are 43 different organizational units involved. Its complete social network diagram, with all the units and all the connections between them, would be unreadable and impossible to use for any analysis. This is why there is the need for mechanisms that provide less complex versions, based on a filtered event log or only on a subset of resources.

Figure 6.3 represents the social network of organizational units displayed in ARIS PPM by using the same principle of filtering out the connections with a frequency less than 1,000. Unlike the activity sequence representation of the process model, where an activity was also deleted from the diagram if all arcs connected to it were removed, all organizational units are kept in the network. The default layout of the social network is with the nodes displayed on a circle, but the user can rearrange them or aggregate multiple nodes based on the communications structure. Additional insights into the resource perspective of the process can be gained by looking at the matrix made of activities and organizational units. In this case, as well, the less frequent mappings activity-organizational unit can be left out by setting a threshold value. The social network depicted in Figure 6.3 suggests the fact that the clinical chemistry department is frequently interacting with the other departments involved in the process.

Futura Reflect supports two ways of simplifying the social network. The first one is based on the
6.2. HOSPITAL EVENT LOG

Figure 6.2: Futura Reflect - Simplified model of the hospital process built based on the most 20% frequent cases

Figure 6.3: ARIS PPM - Simplified social network of the hospital process containing only the relations with frequencies greater than 1,000

approach used for the simplification of the process model, namely by displaying the social network
built only on the 20% most frequent cases. The result is depicted in Figure 6.4. The second option is to generate the social network based on the handovers of work between the resources that perform activities most frequently. This is also done by using the “Unique paths” filter, but the name of the originator needs to be set as the attribute used for computing the paths. Figure 6.5 shows the resulting network. The two social networks provide different types of insights into the process (the first one by having the frequent control flow behavior as starting point and the second one by considering the most frequent behavior based on the resource perspective), which is also suggested by the fact that their diagrams have few similarities.

The hospital process could not be analyzed using the other two evaluated tools. ProcessAnalyzer produced a “System out of memory” exception when mining the event log and we were not able to import the event log in Flow due to the 65536 lines limit specific to Microsoft Office Excel 2003 files.

6.3 Municipality Event Log

Compared to the hospital process, the municipality complaint handling process is rather structured and consists of only 14 different possible paths.

Given the structured nature of the process and the relative little number of activities (14), the first step in analyzing it could be discovering the model that covers all the traces registered in the log. We
performed this first step using two of the evaluated process mining tools, namely Futura Reflect and ProcessAnalyzer.

The process discovered by ProcessAnalyzer is depicted in Figure 6.6. The layout of the model makes it readable and the extra information incorporated in it (e.g. the frequencies of activities and of arcs), gives the user additional insights. Coincidentally, for this process the most frequent path is composed of the sequence of the most frequent arcs. Therefore, in this specific case, the path depicted on the upper part of the model (Domain:heus1 - AG02 Voorbereiden - AG08 GBA afnemer) represents actually the most frequent path in the process. The user can get additional information about the other paths in the process and their frequencies through the Variation Analysis feature.

![Figure 6.6: ProcessAnalyzer - Complete model of the municipality process](image)

Opposite to the situation of the hospital event log, where only the Explore miner could be used for process discovery purposes (the Mine miner does not provide mechanisms for simplifying the model by filtering out infrequent behavior), both miners were used for discovering the process corresponding to the municipality event log. The process discovered by Explore is depicted in Figure 6.7 and the one resulting from Mine is given in Figure 6.8. The two processes are identical. Although the Explore model is able to capture additional information about the process (e.g. arc frequency using different widths and activity frequency using different shades of color), its layout is rather unstructured, which makes it difficult to follow. The opposite holds for the Mine model: no additional information is incorporated in the model, but the activities are displayed on different levels, making the diagram more readable.
CHAPTER 6. DISCUSSION

Figure 6.7: Futura Reflect - Complete model of the municipality process in the Explore miner

Figure 6.8: Futura Reflect - Complete model of the municipality process in the Mine miner

ProcessAnalyzer provides mechanisms of reducing the complexity of process models, as well. This can be done by adjusting the minimum percentages of the activity and flow volumes, in the visibility settings. The tool displays only the activities and the arcs with frequencies higher than the set percentages.
Figure 6.9 represents the model of the municipality process in which only the activities with a volume higher than 10%, on the event log level, are shown. The same process model (Figure 6.10) is obtained in Futura Reflect by only considering the top 75% frequent cases in the algorithm used by the Explore miner.

Using the time perspective for analyzing a process is another option. For instance, in ProcessAnalyzer, the user has the option to display the duration of each case on the same timeline. Cases are represented through boxes of different color based on their duration: green boxes represent cases with a short duration; yellow boxes are the cases with an average duration, while the red boxes symbolize the cases taking the longest time to complete. A fragment of such a timeline is depicted in Figure 6.11. Using this coloring convention enables the user to easily identify the cases with the highest throughput times and focus the more detailed analysis on them.

Another option to identify the most time consuming cases, in ProcessAnalyzer, is to look at one of the duration columns (e.g. “DurationDays”) in the “Case” sheet. By applying the filters available in Microsoft Office Excel, on such a column, one could include for further analysis, for instance, only the cases with duration spanning between two values. We followed this approach to select the cases with durations between 100 and 121 days. Two cases were identified in this range. Their corresponding process model is shown in Figure 6.12. The long waiting times (displayed on the arcs) specific to some transitions, like AG08 GBA afnemer - AG09 Wacht afnemer or AG10 BS derden - AR01 Wacht archief, indicate what is the root cause for the high throughput times.
Figure 6.11: ProcessAnalyzer - Time Analysis functionality displaying the duration of each case on the same timeline

Figure 6.12: ProcessAnalyzer - Model of the municipality process corresponding to the cases with durations between 100 and 121 days

Futura Reflect provides a similar mechanism for analyzing the process corresponding to the most time consuming cases. Using the “Attribute” filtering, one can select only the cases with the throughput times between certain limits. We obtain the same process model as before, with the same waiting times associated to the transitions between activities (Figure 6.13).

6.4 Conclusion

In this chapter we, first, looked at a set of structural metrics that can be used to describe any event log. Next, we focused on discussing the capabilities of the evaluated systems when analyzing event logs stemming from real processes. The first process runs in the healthcare domain, while the second one corresponds to a complaint handling procedure implemented in a municipality. The healthcare event log
was analyzed using ARIS PPM and Futura Reflect. Due to the complexity of the process and its lack of structure, we focused on ways of analyzing the most frequent behavior observed, both from the control-flow and resource perspective. The municipality process, however, more structured and contains less variations. This allowed us to perform process discovery in Futura Reflect and ProcessAnalyzer using the entire event log. Furthermore, we discussed the capabilities provided by the two systems for focusing the analysis on the process instances having the highest durations.
Chapter 7

Conclusions

In this master thesis we identified the need to objectively compare existing process mining systems. Having this as starting point, we defined the goal of the project as follows:

*Develop an evaluation framework based on a set of typical process mining use cases that can be used to assess the strengths and the weaknesses of different process mining tools. This framework will then be applied in practice with the purpose of evaluating several commercial process mining tools.*

The main contribution of this thesis is the development of an objective and systematic approach for assessing the functionalities available in a process mining system. The evaluation we performed on the four tools is repeatable and can be applied on other softwares from the domain. Moreover, the proposed framework can be extended by adding future relevant use cases and corresponding event logs.

We conclude this thesis by summarizing the approach we followed for the development of the use cases evaluation framework for process mining tools (Section 7.1). Next, we present the way the evaluation was performed and we mention the most important findings of this step in Section 7.2. The limitations and some future work suggestions are given in Section 7.3. In Section 7.4 we reveal our opinion with respect to the impact of this study and the future developments of the process mining domain.

7.1 Approach

The first step of the approach we followed consisted of the definition and validation of a set of process mining use cases, that formed the basis of our evaluation framework. This constitutes an important contribution made by this study, due to the lack of a standard list of available process mining functionalities. We defined the set of use cases by conducting a literature study and by looking at the functionalities implemented in ProM. The relevance and the clarity of the resulting use cases were assessed by means of interviews with experienced process mining users and by analyzing the responses for a survey distributed among the process mining community. The use cases list was then transformed into the use case framework, by adding for every use case three additional dimensions, next to the existing description. The three dimensions were a practical situation in which the use case would be useful, any assumptions made relatively to the event log testing the use case, and a set of acceptance criteria used to decide whether the use case is supported or not.

The evaluation framework based on use cases was completed by generating a set of artificial event logs. Each event log in the set was used to test the support that the analyzed systems provide for one or more corresponding use cases. Besides the event logs dedicated to the use case-based evaluation, the
7.2. EVALUATION

The applicability of the developed evaluation framework was demonstrated by evaluating four commercial process mining tools: ARIS Process Performance Manager (Software AG), Flow (Fourspark), Futura Reflect (Futura Process Intelligence), and ProcessAnalyzer (QPR). In the following, we summarize the main findings for each of the systems.

ARIS PPM is a complex process mining system that provides functionalities covering the entire process mining spectrum. Its strongest point is the performance analysis based on the definition of KPIs. This fact is also reflected in the full support that ARIS PPM provides for the use cases related to the organizational and time perspective analysis. Nevertheless, the tool also incorporates features that can be used in the context of conformance checking. The process discovery class of functionalities is fairly good supported, as ARIS PPM uses discovery algorithms able to detect all basic control-flow patterns tested. Limited support is available for the detection of parallelism, due to the additional condition requiring that concurrent activities need to have identical start and completion timestamps.

Unfortunately, no conclusion can be drawn with respect to the support that Flow provides for the list of use cases. The evaluation of Flow could not be performed due to some malfunctioning problems that occurred in the beginning of this step.

Futura Reflect provides a good support for the set of process mining use cases, judging by the number of “+” symbols in the overview table (Table 5.1). The tool incorporates two different process discovery approaches serving different roles. The first one (Mine) is based on a powerful genetic mining algorithm which enables the discovery of parallel constructs, even in more complex processes. However, besides the discovery function, little analysis can be performed using this functionality. The second mining approach (Explore) employs an algorithm that assumes sequential behavior, making the detection of parallelism impossible. Its advantage is, though, the large spectrum of available analysis (both organizational and time perspective analysis). Use cases corresponding to conformance checking tasks can also be executed in Futura Reflect by using filtering and a model comparison feature.

ProcessAnalyzer is mainly focused on process discovery, variation analysis and time perspective analysis. The system uses a process discovery algorithm capable of discovering only the basic control-flow constructs. No conformance checking use cases are supported by system and the organizational analysis is limited due to the process representation using swimlanes flowcharts, which makes the analysis at resource level impossible. The use cases corresponding to the analysis of the time perspective of the process have, however, a good support in ProcessAnalyzer.

In order to ensure the correctness and the accuracy of our findings we presented beforehand the results of the use cases support to the vendors of the four tools. Next to the results for use case-based evaluation and for the performance aspects, the systems were also assessed by looking at additional functionalities, which have not been covered by the set of use cases. These additional functionalities include the import and export formats supported, the preprocessing capabilities and the possibility of integration with external systems. Furthermore, we tested the behavior of the analyzed tools when dealing with event logs stemming from real-life processes.

As a suggestion, we would to mention that the decision of using a process mining tool should not be made by counting the number of “+” and “-” symbols and selecting the system with the highest number
of “+” symbols. Users should weight the results based on the order of relevance of the use cases for the specific tasks and also consider all the aspects related to the functionalities that a process mining tool offers.

7.3 Limitations and Future Work

As previously stated, the definition of the use cases was restricted to the offline analysis and did not cover any of techniques included in the online analysis spectrum, such as prediction, detection or recommendation. The reason for doing this was the inability of running the evaluated systems in an online environment that would allow for these types of analysis. Nevertheless, we believe this is a limitation that could be addressed when considering future developments, since the online operational support is a key aspect in analyzing running processes.

Another limitation consists of the inability of drawing statistical representative conclusions based on the survey responses. This was caused by the fact that we only received 47 responses. Additionally, the distribution of the respondents over the roles is quite unbalanced, i.e., a large number of responses from researchers compared to few responses from process managers. As a consequence, the results obtained could easily change if, for instance, a couple of more responses would be added to a category with a small number of representatives.

7.4 Process Mining Expected Developments

Process mining is a relatively new field in the BPM marketplace. The great number of techniques that were developed in such a short time proves the huge potential that this domain has. However, based on the findings of this study, we believe that this potential is not yet completely exploited by the commercial process mining systems that are available. This conclusion is based on the relative little number of such systems (only eight) and on the results of the evaluation performed on four of them.

We hope that the work done in the context of this master thesis will be helpful in the maturing process of the process mining market. The vendors of the tools participating in this study, could use this experience to identify and improve the weaker points. Users can use this work to compare systems when deciding what process mining tool to use and vendors planning on entering this market in the future could have a reference point when designing the set of functionalities that their systems will incorporate.
Bibliography


# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Approach of the project</td>
<td>3</td>
</tr>
<tr>
<td>1.2</td>
<td>Thesis outline</td>
<td>5</td>
</tr>
<tr>
<td>2.1</td>
<td>Overview of Process Mining [1]</td>
<td>7</td>
</tr>
<tr>
<td>2.2</td>
<td>Screenshot of ARIS PPM</td>
<td>8</td>
</tr>
<tr>
<td>2.3</td>
<td>Screenshot of Flow</td>
<td>9</td>
</tr>
<tr>
<td>2.4</td>
<td>Screenshot of Futura Reflect</td>
<td>10</td>
</tr>
<tr>
<td>2.5</td>
<td>Screenshot of QPR ProcessAnalyzer tab in Microsoft Office Excel</td>
<td>12</td>
</tr>
<tr>
<td>2.6</td>
<td>Screenshot of CPN Tools</td>
<td>13</td>
</tr>
<tr>
<td>3.1</td>
<td>The four phases of the approach followed to create a validated collection of use cases</td>
<td>14</td>
</tr>
<tr>
<td>3.2</td>
<td>The project’s scope in the context of process mining</td>
<td>15</td>
</tr>
<tr>
<td>3.3</td>
<td>Use cases ranking results from the interviews with process analysts and auditors</td>
<td>18</td>
</tr>
<tr>
<td>3.4</td>
<td>Use cases ranking results based on respondents roles</td>
<td>20</td>
</tr>
<tr>
<td>4.1</td>
<td>The process of generating event logs</td>
<td>26</td>
</tr>
<tr>
<td>4.2</td>
<td>Process model example in CPN Tools</td>
<td>27</td>
</tr>
<tr>
<td>4.3</td>
<td>CSV event logs</td>
<td>28</td>
</tr>
<tr>
<td>5.1</td>
<td>ARIS PPM supports the discovery of XOR splits and joins</td>
<td>35</td>
</tr>
<tr>
<td>5.2</td>
<td>Futura Reflect supports the discovery of XOR splits and joins in both miners</td>
<td>35</td>
</tr>
<tr>
<td>5.3</td>
<td>ProcessAnalyzer supports the discovery of XOR splits and joins</td>
<td>35</td>
</tr>
<tr>
<td>5.4</td>
<td>ARIS PPM provides limited support for the discovery of parallel activities in the EPC representation due to the fact that activities B and C need to have identical start and completion timestamps</td>
<td>36</td>
</tr>
<tr>
<td>5.5</td>
<td>Futura Reflect supports the discovery of parallel activities in the <strong>Mine</strong> miner, but not in the <strong>Explore</strong> miner</td>
<td>37</td>
</tr>
<tr>
<td>5.6</td>
<td>ProcessAnalyzer does not support the discovery of parallel activities</td>
<td>37</td>
</tr>
<tr>
<td>5.7</td>
<td>Representation in CPN tools of a highly parallel process model with random activity durations</td>
<td>38</td>
</tr>
<tr>
<td>5.8</td>
<td>ProcessAnalyzer is unable to detect the parallel constructs present in the highly parallel process</td>
<td>38</td>
</tr>
<tr>
<td>5.9</td>
<td>The <strong>Explore</strong> miner in Futura Reflect is unable to detect the parallel constructs present in the highly parallel process</td>
<td>39</td>
</tr>
<tr>
<td>5.10</td>
<td>The <strong>Mine</strong> miner in Futura Reflect is able to correctly capture the parallel constructs present in the highly parallel process</td>
<td>39</td>
</tr>
<tr>
<td>5.11</td>
<td>ARIS PPM is able to discover parallelism if the concurrent activities have identical start and end timestamps</td>
<td>40</td>
</tr>
</tbody>
</table>
5.12 Futura Reflect compares two process models by merging them into one model and highlighting the arcs constituting differences .................................................. 41
5.13 ARIS PPM displays the handovers of work between resources using the social network representation ................................................................. 42
5.14 Futura Reflect displays the handovers of work between resources using the social network representation ................................................................. 43
5.15 ProcessAnalyzer does not display the complete information about the handovers of work due to the limitations of the flowchart representation (e.g. empty swimlanes) .................................................. 44
5.16 ARIS PPM displays the throughput time specific to each case in a table .................................................. 44
5.17 Futura Reflect displays the throughput time specific to each case in a table .................................................. 44
5.18 ProcessAnalyzer displays the throughput time specific to each case in a table .................................................. 45
5.19 ARIS PPM displays the average value of an attribute for all cases following an arc. A high difference between the values of the attribute on two arcs following a choice point suggest that the routing decision is based on the attribute. .................................................. 46
5.20 Futura Reflect displays the average value of an attribute for all cases following an arc in the Explore miner. A high difference between the values of the attribute on two arcs following a choice point suggest that the routing decision is based on the attribute. .................................................. 46
5.21 Process model depicting a parallel construct (B and C) and a XOR split after activity B .................................................. 47
5.22 Model of the process in Figure 5.21 discovered by an algorithm unable to detect parallelism .................................................. 47
5.23 The computation of the most frequent path in the process is affected when the process contains parallel constructs which are not discovered .................................................. 48
5.24 The comparison of a parallel process with the corresponding model discovered by an algorithm unable to detect parallelism .................................................. 49
5.25 The social networks based on the handovers of work for the a parallel model and the corresponding sequential discovered model .................................................. 49
5.26 The value of the waiting time before an activity representing an AND join construct is affected by the inability of detecting parallelism .................................................. 50
5.27 The model discovered by Flow for the sequential process displays an incorrect arc between activities A and C .................................................. 51
6.1 ARIS PPM - Simplified model of the hospital process containing only the arcs with frequencies greater than 1,000 .................................................. 60
6.2 Futura Reflect - Simplified model of the hospital process built based on the most 20% frequent cases .................................................. 61
6.3 ARIS PPM - Simplified social network of the hospital process containing only the relations with frequencies greater than 1,000 .................................................. 61
6.4 Futura Reflect - Simplified social network of the hospital process built based on the most 20% frequent cases .................................................. 62
6.5 Futura Reflect - Simplified social network of the hospital process built based on the resources that perform activities most frequently .................................................. 62
6.6 ProcessAnalyzer - Complete model of the municipality process .................................................. 63
6.7 Futura Reflect - Complete model of the municipality process in the Explore miner .................................................. 64
6.8 Futura Reflect - Complete model of the municipality process in the Mine miner .................................................. 64
6.9 ProcessAnalyzer - Simplified model of the municipality process containing only activities with frequencies greater than 10% .................................................. 65
LIST OF FIGURES

6.10 Futura Reflect - Simplified model of the municipality process built based on the most 75% frequent cases 65
6.11 ProcessAnalyzer - Time Analysis functionality displaying the duration of each case on the same timeline 66
6.12 ProcessAnalyzer - Model of the municipality process corresponding to the cases with durations between 100 and 121 days 66
6.13 Futura Reflect - Model of the municipality process corresponding to the cases with durations between 100 and 121 days 67

D.1 Question 1: In which industry do you work? 96
D.2 Question 2: Which role fits you best? 97
D.3 Question 3: Do you have any practical experience with process mining? 97
D.4 Use cases ranking results based on respondents activity domains 99
D.5 Question 7: Do you think there is any use case missing? 100
D.6 Question 8: How many of the processes in your area contain parallelism? 101
D.7 Question 9a: Typical process - Number of cases (process instances) per year 101
D.8 Question 9b: Typical process - Number of activities per case 102

E.1 Highly parallel process model with random activity durations 104
E.2 Highly parallel process model with constant activity durations 104

F.1 ARIS PPM supports the discovery of sequential activities 110
F.2 Futura Reflect supports the discovery of sequential activities in both miners 111
F.3 ProcessAnalyzer supports the discovery of sequential activities 111
F.4 ARIS PPM supports the discovery of XOR splits and joins 112
F.5 Futura Reflect supports the discovery of XOR splits and joins in both miners 112
F.6 ProcessAnalyzer supports the discovery of XOR splits and joins 113
F.7 ARIS PPM does not support the discovery of OR splits and joins 113
F.8 Futura Reflect does not support the discovery of OR splits and joins in any of the miners 114
F.9 ProcessAnalyzer does not support the discovery of OR splits and joins 114
F.10 Parallel activities - ARIS PPM 115
F.11 Futura Reflect supports the discovery of parallel activities in the Mine miner, but not in the Explore miner 115
F.12 ProcessAnalyzer does not support the discovery of parallel activities 116
F.13 ARIS PPM supports the discovery of the “repeat” variant of structured loops 117
F.14 Futura Reflect supports the discovery of the “repeat” variant of structured loops in both miners 117
F.15 ProcessAnalyzer supports the discovery of the “repeat” variant of structured loops 118
F.16 ARIS PPM supports the discovery of the “do-while” variant of structured loops 118
F.17 Futura Reflect supports the discovery of the “do-while” variant of structured loops in both miners 119
F.18 ProcessAnalyzer supports the discovery of the “do-while” variant of structured loops 119
F.19 ARIS PPM supports the discovery of arbitrary loops 120
F.20 Futura Reflect supports the discovery of arbitrary loops in both miners 120
F.21 ProcessAnalyzer supports the discovery of arbitrary loops 121
F.22 ARIS PPM supports the discovery of skipped activities 121
F.23 Futura Reflect supports the discovery of skipped activities in both miners 122
<table>
<thead>
<tr>
<th>FIGURE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>F.24</td>
<td>ProcessAnalyzer supports the discovery of skipped activities</td>
</tr>
<tr>
<td>F.25</td>
<td>ARIS PPM does not support the discovery of duplicate activities</td>
</tr>
<tr>
<td>F.26</td>
<td>Futura Reflect does not support the discovery of duplicate activities in any of the miners</td>
</tr>
<tr>
<td>F.27</td>
<td>ProcessAnalyzer does not support the discovery of duplicate activities</td>
</tr>
<tr>
<td>F.28</td>
<td>ARIS PPM supports the Routing probabilities by displaying the proportion of cases following each arc</td>
</tr>
<tr>
<td>F.29</td>
<td>Futura Reflect by displaying the percentage of cases following each arc</td>
</tr>
<tr>
<td>F.30</td>
<td>ProcessAnalyzer by displaying the percentage of cases following each arc</td>
</tr>
<tr>
<td>F.31</td>
<td>ARIS PPM does not highlight the most frequent path at the process level, but at the level of each XOR split</td>
</tr>
<tr>
<td>F.32</td>
<td>Futura Reflect highlights the most frequent path of the process (A-B-F-G-K) by displaying it as a straight line</td>
</tr>
<tr>
<td>F.33</td>
<td>ProcessAnalyzer does not highlight the most frequent path at the process level, but at the level of each XOR split</td>
</tr>
<tr>
<td>F.34</td>
<td>Futura Reflect computes statistics about the different paths in the process</td>
</tr>
<tr>
<td>F.35</td>
<td>ProcessAnalyzer computes the percentage of cases following each path in the process</td>
</tr>
<tr>
<td>F.36</td>
<td>ARIS PPM can display only the relations with a frequency lower than a user specified frequency threshold</td>
</tr>
<tr>
<td>F.37</td>
<td>Futura Reflect can display the process models corresponding to the least frequent behavior using both miners</td>
</tr>
<tr>
<td>F.38</td>
<td>Futura Reflect checks the four-eye principle using the &quot;Separation of Duties Violations&quot; filter</td>
</tr>
<tr>
<td>F.39</td>
<td>ARIS PPM provides three different ways of checking the compliance with any rule specified at import time</td>
</tr>
<tr>
<td>F.40</td>
<td>Futura Reflect compares two process models by merging them into one model and highlighting the arcs constituting differences</td>
</tr>
<tr>
<td>F.41</td>
<td>ARIS PPM displays associations between activities and their executors in a table</td>
</tr>
<tr>
<td>F.42</td>
<td>Futura Reflect displays associations between activities and their executors in a chart</td>
</tr>
<tr>
<td>F.43</td>
<td>ProcessAnalyzer does not output correct associations between activities and their executors due to the flowchart representation</td>
</tr>
<tr>
<td>F.44</td>
<td>ARIS PPM displays the name of the originator under each activity in a process instance</td>
</tr>
<tr>
<td>F.45</td>
<td>Futura Reflect displays the name of the resources involved in a process instance in a table</td>
</tr>
<tr>
<td>F.46</td>
<td>ProcessAnalyzer displays the number of activities executed by each originator in a process instance</td>
</tr>
<tr>
<td>F.47</td>
<td>ARIS PPM displays the handovers of work between resources using the social network representation</td>
</tr>
<tr>
<td>F.48</td>
<td>Futura Reflect displays the handovers of work between resources using the social network representation</td>
</tr>
<tr>
<td>F.49</td>
<td>ProcessAnalyzer does not display the complete information about the handovers of work due to the limitations of the flowchart representation (e.g. empty swimlanes)</td>
</tr>
<tr>
<td>F.50</td>
<td>ARIS PPM displays a chart with the number of received handovers of each employee</td>
</tr>
<tr>
<td>F.51</td>
<td>ARIS PPM displays the throughput time specific to each case in a table</td>
</tr>
<tr>
<td>F.52</td>
<td>Futura Reflect displays the throughput time specific to each case in a table</td>
</tr>
<tr>
<td>F.53</td>
<td>ProcessAnalyzer displays the throughput time specific to each case in a table</td>
</tr>
<tr>
<td>F.54</td>
<td>ARIS PPM displays a chart with the duration of each activity</td>
</tr>
<tr>
<td>F.55</td>
<td>Futura Reflect displays under each activity its duration in the Explore miner</td>
</tr>
<tr>
<td>F.56</td>
<td>ProcessAnalyzer displays in the bottom left corner of each activity its duration</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>F.57</td>
<td>ARIS PPM displays a chart with the waiting times before each activity.</td>
</tr>
<tr>
<td>F.58</td>
<td>Futura Reflect displays the waiting times on the arcs in the <em>Explore</em> miner.</td>
</tr>
<tr>
<td>F.59</td>
<td>ProcessAnalyzer displays the waiting times on the arcs.</td>
</tr>
<tr>
<td>F.60</td>
<td>ARIS PPM displays the EPC model corresponding to a set of process instances with throughput times in a certain interval.</td>
</tr>
<tr>
<td>F.61</td>
<td>Futura Reflect displays in the <em>Explore</em> miner the process corresponding to the cases with throughput times in a specified interval.</td>
</tr>
<tr>
<td>F.62</td>
<td>ProcessAnalyzer displays the process model corresponding only to the cases with throughput times higher than a specified value.</td>
</tr>
<tr>
<td>F.63</td>
<td>ARIS PPM displays a chart with the number of cases entering the process over time.</td>
</tr>
<tr>
<td>F.64</td>
<td>ARIS PPM displays a chart showing the fractions of time in which the resources are busy.</td>
</tr>
<tr>
<td>F.65</td>
<td>ProcessAnalyzer displays a chart with average resource utilization for a selected resource over time.</td>
</tr>
<tr>
<td>F.66</td>
<td>ARIS PPM displays the timely sequence of events for a specific case.</td>
</tr>
<tr>
<td>F.67</td>
<td>ProcessAnalyzer displays the timely sequence of events for a specific case.</td>
</tr>
<tr>
<td>F.68</td>
<td>ARIS PPM displays the average value of an attribute for all cases following an arc. A high difference between the values of the attribute on two arcs following a choice point suggest that the routing decision is based on the attribute.</td>
</tr>
<tr>
<td>F.69</td>
<td>Futura Reflect displays the average value of an attribute for all cases following an arc in the <em>Explore</em> miner. A high difference between the values of the attribute on two arcs following a choice point suggest that the routing decision is based on the attribute.</td>
</tr>
</tbody>
</table>
List of Tables

3.1 Top 5 and Bottom 5 Use Cases for Process Analysts ................................................. 19
3.2 Top 5 and Bottom 5 Use Cases for Auditors .............................................................. 19
3.3 Aggregated results survey .......................................................................................... 21
4.1 Relations Use case - Event log .................................................................................. 30
5.1 Result Overview of the Use Case-based Evaluation ...................................................... 52
5.2 Performance Aspects - ARIS PPM ............................................................................ 54
5.3 Performance Aspects - Futura Reflect ....................................................................... 54
5.4 Performance Aspects - ProcessAnalyzer ................................................................. 55
6.1 Structural log metrics ............................................................................................... 59
B.1 Results ranking - Interview 1 .................................................................................... 89
B.2 Results ranking - Interview 2 .................................................................................... 90
B.3 Results ranking - Interview 3 .................................................................................... 91
B.4 Results ranking - Interview 4 .................................................................................... 92
B.5 Results ranking - Interview 5 .................................................................................... 92
B.6 Results ranking - Interview 6 .................................................................................... 93
D.1 Question 4: For which purpose did you use process mining? .................................. 98
D.2 Question 5: What do you think is the biggest benefit of process mining? ................ 98
D.3 Question 6: Overall results use cases rankings ....................................................... 99
D.4 Missing use cases ..................................................................................................... 100
D.5 Question 10: Which additional functionalities do you consider important for a process mining tool? ............................................................................................................. 102
E.1 Event log E1 - Sequence ........................................................................................... 103
E.2 Event log E2 - XOR split ........................................................................................... 103
E.3 Event log E3 - OR split ............................................................................................. 103
E.4 Event log E4 - Parallel activities ................................................................................ 103
E.5 Event log E5 - Structured loop - Repeat ................................................................... 104
E.6 Event log E6 - Structured loop - Do while ............................................................... 105
E.7 Event log E7 - Arbitrary loop event ......................................................................... 105
E.8 Event log E8 - Skipped tasks .................................................................................... 105
E.9 Event log E9 - Duplicate tasks .................................................................................. 105
E.10 Event log E10 - Routing probabilities .................................................................... 105
E.11 Event log E11 - Frequent ......................................................... 106
E.12 Event log E12 - Conformance checking ........................................ 106
E.13 Event log E13 - Organizational and time perspective ......................... 107
E.14 Event log E14 - Cycles ........................................................... 108
E.15 Event log E15 - Data perspective ............................................... 109
Appendix A

Process Mining Use Case Framework

Additionally, a use case might have other specific assumptions that are mentioned in the body of the use case.

Use case 1: Structure of the process

Description: Determine the structure of an unknown process or discover how a process looks like in practice.

Example: This use case could be used in order to mine the structure of an unknown process. By looking at a discovered process model it is possible to get an overview on how things are really happening in practice. Moreover, the discovered model can be used for communication and redesign purposes.

Assumptions: An event log is required for testing the support of the case. The traces in the event log need to contain only information about the name of the activities executed.

Acceptance criteria: The set of acceptance criteria for this use case are based on the most common control flow pattern that might occur in a process. The systems will be evaluated whether they incorporate mechanisms able to discover:

1. Sequential activities - The tools supporting this sub use case provide a control flow representation that suggests that a set of activities is executed in sequential order, meaning that one activity cannot start before the previous one is completed.

2. XOR splits and joins - The tools supporting this sub use case provide a control flow representation that suggests that after the completion of an activity that is considered to be a XOR split, the thread of control is passed to exactly one of the outgoing branches. The convergence of the multiple branches into one subsequent branch is, then, done by executing an activity, considered to be a XOR join.

3. OR splits and joins - This sub use case is supported if the system provides a control flow representation that suggests the fact that after the completion of an activity, considered to be the OR split, the thread of control can be passed to one or more of the outgoing branches and is able to discover such a behavior. The convergence of the multiple branches into one subsequent branch is, then, done by executing an activity, considered to be a OR join.

4. Parallel activities - The tools supporting this sub use case provide a control flow representation suggesting the divergence of a branch into multiple branches that execute concurrently.
5. **Structured loops - the “repeat” variant** - The tools supporting this sub use case provide a control flow representation suggesting the possibility of executing an activity or a set of activities repeatedly. In particular for this sub use, from the discovered model it should be clear that the activities involved in the loop are executed at least once.

6. **Structured loops - the “do while” variant** - The tools supporting this sub use case provide a control flow representation suggesting the possibility of executing an activity or a set of activities repeatedly. In particular for this sub use, from the discovered model it should be clear that a subset of the activities involved in the loop can be skipped.

7. **Arbitrary loops** - The tools supporting this sub use case provide the ability to discover processes that contain cycles which have more than one entry or exit point.

8. **Skipped activities** - The tools supporting this sub use case have the ability to detect the possibility of skipping a certain activities or set of activity in the discovered process.

9. **Duplicate activities** - The tools supporting this sub use case have the ability to differentiate between two activities having identical names, but being executed in different contexts (e.g., only in the beginning and in the end of the process). The result of this distinction is the representation of the activity in the discovered model as two or more separate (duplicate) activities.

This use case constitutes the basic functionality that a process mining tool needs to support in order to be considered for this evaluation. Therefore, the acceptance criteria will be tested independently, using different event logs. The results of the evaluation will not be aggregated at the use case level, but presented separately for each acceptance criterion.

**Use case 2: Routing probabilities**

**Description:** Get a deeper understanding of the process by looking at the probabilities of following one path or another after a choice point.

**Example:** From the user’s point of view it might be useful to get a better grasp on the structure and behavior of the process by looking at the routing probabilities in choice points. These probabilities might be used as KPIs when monitoring the process.

**Assumptions:** An event log is required for testing the support of the case. The traces in the event log need to contain only information about the name of the activities executed.

**Acceptance criteria:** The tools supporting this use case should provide a mechanism of visualizing the routing probabilities specific to each choice point in the process. Each outgoing arc of a choice point has associated such a routing probability.

**Use case 3: Most frequent path in the process**

**Description:** Discover the path in the process that is followed by the highest percentage of cases. A path is considered to be a sequence of activities.

**Example:** The user is able to get a clearer understanding of their process by knowing the path of events followed by the majority of cases. This information can be used for redesign purposes, since optimizing the most frequent path in the process might bring benefits to the overall performance.
Assumptions: An event log is required for testing the support of the case. The traces in the event log need to contain only information about the name of the activities executed.

Acceptance criteria: The tools supporting this pattern should provide the possibility of highlighting or pointing out which is the most frequent path in the process.

Use case 4: Distribution of cases over paths

Description: Discover common and uncommon behavior in the process by looking at the distribution of cases over the possible paths in the process. A path is considered to be a sequence of activities.

Example: By visualizing the percentage of process instances following each possible path the user can decide to focus on the most frequent behavior. This also allows determining whether there are a lot of variations in the process and possibly representing a starting point for a redesign meant to reduce these variations.

Assumptions: An event log is required for testing the support of the case. The traces in the event log need to contain only information about the name of the activities executed.

Acceptance criteria: The tools supporting this pattern should have the ability to associate to each path in the process the percentage or the number of cases that follow it. Both possibilities are accepted.

Use case 5: Exceptions from the normal path

Description: Discover the outliers of the process by looking at the least frequent behavior observed in practice. A path is considered to be a sequence of activities.

Example: Deviations from the normal behavior can be an indication that the guidelines are not followed in practice or that things are not implemented in a correct way. Visualizing the outliers of a process may trigger a process redesign.

Assumptions: An event log is required for testing the support of the case. The traces in the event log need to contain only information about the name of the activities executed.

Acceptance criteria: The tools supporting this pattern should be able to identify any parts of the process that might constitute exceptions from the normal behavior and then provide a mechanism of highlighting these parts in the process model. The decision whether a pattern in the process model represents an exception or not is based on comparing its frequency of occurrence with other frequent behaviors observed in the rest of the event log. The tool can either establish its own threshold to be used in classifying the patterns as frequent or infrequent based on the specific characteristics of the input event log or it can use a threshold value given by the user. If at least one approach is provided by a tool, the use case is supported.

Use case 6: The degree in which the rules are obeyed

Description: Check whether the rules and regulations related to the process are obeyed.

Example: Checks whether the business rules and principles are respected in practice and assess whether they are necessary to ensure the integrity and the quality of the process.
**Assumptions:** An event log is required for testing the support of the case. The traces in the event log need to contain information about the name of the activities executed and the identity of the resources performing executing them.

**Acceptance criteria:** The acceptance criteria for this use case have two levels.

1. The *First level* corresponds to the possibility of evaluating whether the checking of a specific type of rule is supported. The first level criterion consists of establishing the existence mechanisms capable of checking the compliance to the four-eye principle, also known as the separation of duties rule, the possibility of checking whether an activity has been executed after a certain amount of time after another activity and so on. To sum up, any hard coded functionality that enables the possibility of verifying a certain kind of rule is accepted.

2. On the *Second level* a more generic functionality is evaluated. This functionality refers to the possibility that users define their own rules to be checked.

**Use case 7: Compliance to the explicit process**

**Description:** Compare the documented process model with the real process, either given by the event log either given by a explicit model.

**Example:** By executing this use case scenario, the user can detect possible inconsistencies between the real process and the theoretical one, which may then either lead to the enforcement of the documented process or to an update of the documentation.

**Assumptions:** For testing the support of the use case an event log and a model of the explicit process are required. The traces in the event log need to contain only information about the name of the activities executed.

**Acceptance criteria:** The criteria used to decide whether this use case is supported are divided into two categories. The two categories are:

1. *Qualitative criterion:* The tools supporting this pattern provide the possibility of comparing two process models (the discovered one and the explicit one) in a visual fashion and are able to highlight the parts of the process where differences occur.

2. *Quantitative criterion:* The tools supporting this pattern should provide the possibility of measuring the percentage of fit between the discovered process model and the event log, based on predefined metrics.

**Use case 8: Resources per task**

**Description:** Discover the relation between resources and tasks.

**Example:** This use case gives an overview on both the group of resources working on the process and the set of tasks specific to each of these resources. This is a starting point for understanding the organizational dimension of a process. Additionally, this use case tells who is responsible for each task in isolation and who is sharing the responsibility for a certain activity.
Assumptions: An event log is required for testing the support of the case. The traces in the event log need to contain information about the name of the activities executed and the identity of the resources performing executing them.

Acceptance criteria: In order to support this use case, a tool should be able to output associations between tasks and resources based on the information stored in the event log. These associations are not specific to a case or set of cases, but apply to the entire process.

Use case 9: Resources involved in a case

Description: Discover the group of resources involved in executing a particular case.

Example: This information could be used to track down problems related to a particular process instance and to find the source of error or the cause of the delay among the resources involved in the execution of the case.

Assumptions: An event log is required for testing the support of the case. The traces in the event log need to contain information about the name of the activities executed and the identity of the resources performing executing them.

Acceptance criteria: The tools supporting this pattern should present the group of resources involved in a specific selected process instance.

Use case 10: Work handovers

Description: Manage resource location or determine possible causes for quality and time issues by looking at how work is transferred between resources.

Example: This information can be used to improve the distribution of work or for analyzing quality, time and cost problems.

Assumptions: For testing the support of the use case an event log and a model of the explicit process are required. The traces in the event log need to contain information about the name of the activities executed and the identity of the resources performing executing them.

Acceptance criteria: A tool that supports this pattern is capable of constructing a social network of interactions between the resources involved in the execution of a process, based on the handovers of work that occur.

Use case 11: Central employees

Description: Determine who the central resources for a process are by analyzing the social network based on handovers of work.

Example: Knowing who the central employees of a process provides insight into the resource perspective and the way the process is organized. The central employee can be seen as the middle-man in the process, as a hub for work handover. Being aware of this information, the users can further determine for instance how sensitive the process is in case the central employee leaves the company.
**Assumptions:** An event log is required for testing the support of the case. The traces in the event log need to contain information about the name of the activities executed and the identity of the resources performing executing them.

**Acceptance criteria:** The tools supporting this pattern should point out who are the central employees of a process based on number of work items transferred to them.

**Use case 12: Throughput time of cases**

**Description:** Determine the time that passed since the start of a case in progress until its completion.

**Example:** Statistics related to throughput time serve as a starting point for more in depth performance analysis. Moreover, there are often target throughput times in a company in which a process should be finished. Throughput time analysis allows to verify whether these targets have been met.

**Assumptions:** An event log is required for testing the support of the case. The traces in the event log need to contain information about the name of the activities executed and their start and end timestamps.

**Acceptance criteria:** One requirement for supporting this use case is to provide the throughput time corresponding to each case in the process. Alternatives include the ability of providing the average throughput time of cases and the minimum and maximum values calculated. Both options are accepted.

**Use case 13: Slowest activities**

**Description:** Discover potential time problems by looking at the activities in the process that take a long time.

**Example:** The user can identify where problems might exist by considering the activities that take most time.

**Assumptions:** An event log is required for testing the support of the case. The traces in the event log need to contain information about the name of the activities executed and their start and end timestamps.

**Acceptance criteria:** A basic requirement for a tool that supports this use case is to be able to compute, based on the information stored in the event log, the average service time for each activity of the process. This criterion is used to decide the support of use case 13. Additionally, tools could also provide mechanisms of highlighting the activities that take longer that a certain amount of time. This amount of time can be either specified as input by the user or can be automatically determined by the system relative to the durations of all the activities of the process.

**Use case 14: Longest waiting times**

**Description:** Determine delays between activities by analyzing the waiting times before each activity.

**Example:** Performance problems caused by a bad synchronization of tasks or by an unsuitable resource allocation are reflected by the long waiting times specific to some activities.

**Assumptions:** An event log is required for testing the support of the case. The traces in the event log need to contain information about the name of the activities executed and their start and end timestamps.
**Acceptance criteria:** The minimal requirement for a tool that supports this use case is to be able to compute, based on the information stored in the event log, the waiting times before each activity of the process. This criterion is used when evaluating the support for this use case. Next to this basic functionality, tools can also highlight the problem spots of the process caused by high waiting time values. A waiting time value can be considered high in relation to a user input threshold or in relation to a value automatically determined by the system.

**Use case 15: Cycles**

**Description:** Learn whether additional delays occur in the process due to cycles.

**Example:** This kind of behavior observed within a process might be an indicator that in practice things are not happening as they should happen. Loops introduce additional time delays, decrease the efficiency of the overall process, might be a sign that an error was made, and can cause the occurrence of bottlenecks.

**Assumptions:** An event log is required for testing the support of the case. The traces in the event log need to contain information about the name of the activities executed and their start and end timestamps.

**Acceptance criteria:** The tools supporting this pattern should be able to identify the paths having throughput time higher than a given value or higher than a value automatically identified by the system based on the throughput times of other paths. Having identified these paths, the tool should check whether they contain any cycles (repetitions of a set of activities) and highlight them.

**Use case 16: Arrival rate of cases**

**Description:** Determine the frequency with which new cases arrive in the process.

**Example:** The pattern followed by cases arriving in the process is important information that is taken into consideration when a new process redesign is applied.

**Assumptions:** An event log is required for testing the support of the case. The traces in the event log need to contain information about the name of the activities executed and their start and end timestamps.

**Acceptance criteria:** The tools supporting this pattern should provide information about the distribution which describes the arrival pattern of cases entering the process.

**Use case 17: Resource utilization rate**

**Description:** Determine what are the utilization rates of the resource i.e, measure the fraction of time that a resource is busy.

**Example:** This information provides insight from two perspectives. First, the existence of overutilized resources might also indicate possible bottlenecks in the process. Opposite, resources with a very low utilization are a sign that the organization of the process is inefficient and that a redesign should be considered.

**Assumptions:** An event log is required for testing the support of the case. The traces in the event log need to contain information about the name of the activities executed, their start and end timestamps, and the identity of the resources performing them.
Acceptance criteria: The tools supporting this pattern should provide information about the utilization rate characteristic to each resource.

Use case 18: Time sequence of events

Description: Get a deeper understanding on the organization of a process by looking at the time sequence of activities for a specific case. (e.g. Gantt-graph for activities).

Example: By having the possibility of visualizing the time sequence of events, the user can get a better overview on the overlapping activities in the process. This representation captures extra-information about the process that is lost in the control flow model representation.

Assumptions: An event log is required for testing the support of the case. The traces in the event log need to contain information about the name of the activities executed and their start and end timestamps.

Acceptance criteria: In order to support this use case, a tool should provide a mechanism of representing the chronological sequence of the activities of a specific process instance. The diagram depicting this sequence captures information on both the start and end time of each activity and makes it possible to see the overlapping activities.

Use case 19: Business rules

Description: Discover what are the process attributes that influence the choice points and what are the conditions for following one branch or another.

Motivation: The user can discover which data fields stored in the log determine the choice between different branches in the process. The choices made can be explained by identifying rules which take into account the characteristics of a case.

Assumptions: An event log is required for testing the support of the case. The traces in the event log need to contain information about the name of the activities executed and all attributes that influence the routing decision at choice points.

Acceptance criteria: The tools supporting this pattern should highlight the decisions points and the attributes values for following each of the possible paths.
Appendix B

Interviews Summary

Interview 1

Role: Process Analyst
Industry: Banking

1. Introduction
   • **Experience with Process Mining**
     The interviewee was involved in a pilot process mining project within a Dutch bank. The purpose of the project was to analyze an internal HR process using two process mining tools: ProM and Futura Reflect. The process was simple and well-defined. At the end, the results obtained from the two systems were compared.
   • **Benefits brought by Process Mining**
     According to the interviewee the main benefits that process mining brings are the possibility of visualizing the actual process and the detection of bottlenecks. He also mentioned the role of process mining in the objective analysis of the AS-IS situation and as a basis for redesign and the possibility of analyzing different parts of a process in different levels of depth.

2. Results ranking list of use cases

   Table B.1: Results ranking - Interview 1

<table>
<thead>
<tr>
<th>Order number</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use case number</td>
<td>1</td>
<td>3</td>
<td>15</td>
<td>13</td>
<td>4</td>
<td>12</td>
<td>2</td>
<td>7</td>
<td>14</td>
<td>16</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Order number</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use case number</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>5</td>
<td>6</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>8</td>
</tr>
</tbody>
</table>

3. **Additional insights**
   In the interviewee’s opinion process mining should provide a mechanism of performing root-cause analysis when problems are discovered. At the moment process mining stays in an experimental phase because a link between objective insights and further follow-up is required. It is often the case that managers or process analysts get a large amount of information about the analyzed process, but this information is not translated into solutions. The interviewee also mentioned the
possibility of combing and comparing the objective data obtained by applying process mining techniques with subjective data with the purpose of gaining more accuracy when analyzing a process. On a different perspective, people seem more enthusiastic for use on the long term (for projects) than on a daily basis. No missing use cases were identified.

As a conclusion of this interview, process mining can be used in three different directions: start point for a structural change, auditing, and as a stand-alone functionality (similar to the concept of quick-scan).

**Interview 2**

Role: Process Analyst  
Industry: Business Process Management Consultancy

1. **Introduction**
   - **Experience with Process Mining**
     The experience of the interviewee in the process mining context consists of attending a presentation on Futura Reflect. He has however no practical experience with using it.
   - **Benefits brought by Process Mining**
     The most significant benefit of process mining is that its results are based on evidence and facts, making it an accurate technique. Additional advantages are related to the time and cost perspectives: process mining is faster and cheaper than the other techniques employed in the process analysis phase.

2. **Results ranking list of use cases**

   Table B.2: Results ranking - Interview 2

<table>
<thead>
<tr>
<th>Order number</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use case number</td>
<td>3</td>
<td>12</td>
<td>6</td>
<td>15</td>
<td>1</td>
<td>13</td>
<td>2</td>
<td>4</td>
<td>14</td>
<td>18</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Order number</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use case number</td>
<td>10</td>
<td>5</td>
<td>17</td>
<td>19</td>
<td>16</td>
<td>9</td>
<td>7</td>
<td>8</td>
<td>11</td>
</tr>
</tbody>
</table>

3. **Additional insights**
   No missing use cases were identified. According to the interviewee process mining has a lot of potential, but many people do not see it now.

**Interview 3**

Role: Process Analyst  
Industry: Healthcare

1. **Introduction**
   - **Experience with Process Mining**
     The interviewee was involved in the project aimed at analyzing a gynecology-oncology process using techniques implemented in ProM. The process spans over several departments in
the hospital (multi-disciplinary), contains a lot of diagnosis steps and decision making and covers a large number of patients. The analysis showed differences between the protocol and what happens in practice. The first step of the analysis was to look at the general structure of the process and then in the second phase problems related to throughput time, repetitions and resource allocation were deeper inspected. The focus of the analysis was on the most frequent behavior.

- **Benefits brought by Process Mining**
  Process mining takes less time and gives insight into the process. These are according to the interviewee the main advantages. He also mentioned the ease of detecting patterns compared to the manual operation and the benefits provided by the simulation aspect of process mining.

2. **Results ranking list of use cases**

Table B.3: Results ranking - Interview 3

<table>
<thead>
<tr>
<th>Order number</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use case number</td>
<td>1</td>
<td>7</td>
<td>3</td>
<td>4</td>
<td>15</td>
<td>10</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>17</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Order number</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use case number</td>
<td>2</td>
<td>11</td>
<td>19</td>
<td>18</td>
<td>16</td>
<td>9</td>
<td>8</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>

3. **Additional insights**
No missing use cases were identified. According to the interviewee process mining can also be viewed as a communication tool. Once again, he mentioned the usefulness of coupling simulation with process mining.

**Interview 4**

Role: Auditor
Industry: Public Sector

1. **Introduction**

- **Experience with Process Mining**
The experience of the interviewee with process mining techniques consists of the analysis of the TBU (Tegemoetkoming Buitengewone Uitgaven) process with the purpose of getting better insight and checking whether what happens in practice is 100% conform to the theoretical process. The project had three goals: to identify that cases that do not complete, but get stuck in the process, to check its correctness, and to look into the time aspect. The process was spread over multiple IT systems, so gathering the data and linking all systems was a bit of a problem before starting the actual analysis.

- **Benefits brought by Process Mining**
The most important advantages specific to process mining are the complete transparency it offers and the full insight into processes. In the future process mining might be also used as a tool for continuous monitoring.

2. **Results ranking list of use cases**
Table B.4: Results ranking - Interview 4

<table>
<thead>
<tr>
<th>Order number</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use case number</td>
<td>5</td>
<td>14</td>
<td>19</td>
<td>18</td>
<td>6</td>
<td>7</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Order number</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use case number</td>
<td>11</td>
<td>15</td>
<td>13</td>
<td>10</td>
<td>16</td>
<td>12</td>
<td>17</td>
<td>8</td>
<td>3</td>
</tr>
</tbody>
</table>

3. **Additional insights**
The ease of using a process mining tool represents an important aspect for the interviewee. In his opinions, systems should provide ways of representing the results in a fashion that is more presentable to managers. No missing use cases were found.

**Interview 5**

Role: Auditor  
Industry: Public Sector

1. **Introduction**
   - **Experience with Process Mining**
     The interviewee has no practical experience with using process mining. However, he supervised interviewee 6 in the project made for analyzing a purchase process from a SAP system from an auditing perspective. The control flow of the process and conformance checking aspects were investigated. The interviewee also collaborated with interviewee 4 in the project that analyzed the TBU process.
   - **Benefits brought by Process Mining**
     According to the interviewee, process mining enables its users to see what is exactly happening in a process. This way one can discover strange things taking place in practice, which can constitute exceptions from the normal path of events. Moreover, using the filtering functionalities, one could choose to look into cases which are more important of significant than others.

2. **Results ranking list of use cases**

Table B.5: Results ranking - Interview 5

<table>
<thead>
<tr>
<th>Order number</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use case number</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>7</td>
<td>18</td>
<td>19</td>
<td>15</td>
<td>5</td>
<td>14</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Order number</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use case number</td>
<td>13</td>
<td>16</td>
<td>8</td>
<td>6</td>
<td>17</td>
<td>12</td>
<td>11</td>
<td>10</td>
<td>9</td>
</tr>
</tbody>
</table>

3. **Additional insights**
Additional features that a process mining tool should provide, in the interviewee’s opinion, are the continuous monitoring of a process and the possibility of receiving alerts every time something is not working as it should. He also mentioned the importance of the user interface and the large
amount of time it takes to prepare the log for the analysis phase. No additional use case was identified.

**Interview 6**

Role: Auditor  
Industry: Public Sector  

1. **Introduction**
   - **Experience with Process Mining**  
     The interviewee analyzed from an auditing perspective a purchase process from a SAP system using process mining techniques. This was done in the context of the interviewee’s master thesis “Automated Analysis of Business Processes for IT auditing”. As an IT-auditor working in the public sector, the interviewee is still using process mining to investigate processes.
   - **Benefits brought by Process Mining**  
     As benefits of process mining the interviewee mentioned the possibility of seeing what is actually happening in practice, and from an auditing perspective the ability of comparing the explicit process with the one discovered from the event log.

2. **Results ranking list of use cases**

   Table B.6: Results ranking - Interview 6

<table>
<thead>
<tr>
<th>Order number</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use case number</td>
<td>1</td>
<td>4</td>
<td>7</td>
<td>3</td>
<td>6</td>
<td>5</td>
<td>2</td>
<td>15</td>
<td>12</td>
<td>13</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Order number</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use case number</td>
<td>14</td>
<td>18</td>
<td>19</td>
<td>17</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>16</td>
<td>11</td>
</tr>
</tbody>
</table>

3. **Additional insights**

   According to the interviewee process mining could be used in the context of continuous monitoring, but setting up the environment for this still requires some manual steps. Important features that process mining software should incorporate are preprocessing of data and documentation and help functions. The interviewee also mentioned the necessity of repeating identical steps when performing the same type of analysis on different events log and suggested the possibility of recording the actions performed when setting up an analysis, saving them, and then using the resulting sequence of saved actions for further setting analysis, without the need of manually performing them again for every new analysis.
Appendix C

Survey Contents

1. In which industry do you work?

2. Which role fits you best?
   a) Process Analyst
   b) Auditor
   c) Process Manager
   d) Researcher
   e) Other (Please specify which)

3. Do you have any practical experience with process mining?
   a) No, not yet.
   b) Yes, I have used it. (Please specify also the number of projects)

4. If you answered Yes to Question No. 3, for which purpose did you use process mining?

5. What do you think is the biggest benefit of process mining?

6. Please rank all the following process mining use cases according to their importance/relevance to you. To do this: first read the complete list of use cases and identify the most important use case. Then, drag it to the right side and repeat the process for the remaining use cases.

   1 Determine the structure of an unknown process or discover how a process looks like in practice.
   2 Get a deeper understanding of the process by looking at the probabilities of following one path or another after a choice point.
   3 Discover what is the path in the process that is followed by the highest percentage of cases.
   4 Discover common and uncommon behavior in the process by looking at the distribution of cases over the possible paths in the process.
   5 Discover the outliers of the process by looking at the exceptional behavior observed in practice.
   6 Check whether the rules and regulations related to the process are obeyed.
   7 Compare the documented process model with the real process as observed in the event log.
8. Learn who has the responsibility for performing an activity.
9. Discover the group of resources involved in solving a particular case.
10. Manage resource location or determine possible causes for quality and time issues by looking at how work is transferred between resources.
11. Determine who the central resources for a process are by looking at the resources that get work transferred to them most often.
12. Determine the average time that passed since the start of a case in process until its completion.
13. Discover potential time problems by looking at the slowest activities in the process.
14. Determine possible synchronization problems between activities by analyzing the waiting times before each activity.
15. Learn whether additional delays occur in the process due to loopbacks.
16. Determine the frequency with which new cases arrive in the process.
17. Determine what are the utilization rates of the resources / what is the fraction of time that a resource is busy.
18. Get a deeper understanding on the organization of a process by looking at the timely sequence of activities for a specific case. (e.g. Gant-graph for activities).
19. Discover what are the process attributes that influence the choice points and what are the conditions for following one branch or another.

7. Do you think there is any use case missing?
   a) No
   b) Yes (Please specify which)

8. How many of the processes in your area contain parallelism (activities that can be performed in parallel)?
   a) None
   b) Up to 20%
   c) Between 20 and 50%
   d) Between 50 and 80%
   e) More than 80%

9. A typical process in your area has the following characteristics:
   a) Number of cases (process instances) per year:
   b) Number of activities per case:

10. Which additional functionalities, not covered in the use cases presented in question 6, do you consider important for a process mining tool? (for example specific import or export capabilities)

11. Is there anything else you would like to add?
Figure D.1: Question 1: In which industry do you work?
Figure D.2: Question 2: Which role fits you best?

Figure D.3: Question 3: Do you have any practical experience with process mining?
### Table D.1: Question 4: For which purpose did you use process mining?

<table>
<thead>
<tr>
<th>Process Mining Class</th>
<th>Functionality</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Discovery</strong></td>
<td>Discover and show the true business flow within companies</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Find the frequent patterns in the event log</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Discover processes from log traces</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Mine the actual used process model out the history data (Costs)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Automated business process discovery</td>
<td>3</td>
</tr>
<tr>
<td><strong>Conformance checking</strong></td>
<td>Conformance between event logs and Petri net specifications</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Compliance analysis</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Evaluate the compliance of processes with best practices</td>
<td>2</td>
</tr>
<tr>
<td><strong>Performance analysis</strong></td>
<td>Identify inefficiency</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Resource usage analysis</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Duration analysis</td>
<td>4</td>
</tr>
<tr>
<td><strong>Organizational analysis</strong></td>
<td>Show the maximum number of involvement of resources</td>
<td>3</td>
</tr>
<tr>
<td><strong>Monitoring</strong></td>
<td>Develop a real-time process monitoring solution</td>
<td>2</td>
</tr>
</tbody>
</table>

### Table D.2: Question 5: What do you think is the biggest benefit of process mining?

<table>
<thead>
<tr>
<th>Benefits</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Functional qualities</strong></td>
<td></td>
</tr>
<tr>
<td>Visualization of processes, analysis</td>
<td>4</td>
</tr>
<tr>
<td>Extracting process models</td>
<td>5</td>
</tr>
<tr>
<td>Exploring and discovering processes</td>
<td>3</td>
</tr>
<tr>
<td>Process performance</td>
<td>4</td>
</tr>
<tr>
<td>Get real key figures about your process</td>
<td>2</td>
</tr>
<tr>
<td><strong>Non-functional qualities</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Objectivity</strong></td>
<td></td>
</tr>
<tr>
<td>Objective unbiased information</td>
<td>2</td>
</tr>
<tr>
<td>Objectivity on how the process is executed in real-life</td>
<td>3</td>
</tr>
<tr>
<td><strong>Accuracy</strong></td>
<td></td>
</tr>
<tr>
<td>Accurate information based on actual facts</td>
<td>2</td>
</tr>
<tr>
<td><strong>Speed</strong></td>
<td></td>
</tr>
<tr>
<td>Speed of getting results</td>
<td>2</td>
</tr>
<tr>
<td>Saving time</td>
<td>2</td>
</tr>
<tr>
<td><strong>Transparency</strong></td>
<td></td>
</tr>
<tr>
<td>Making real processes transparent</td>
<td>1</td>
</tr>
<tr>
<td>Transparent view and analysis of business processes</td>
<td>1</td>
</tr>
</tbody>
</table>


Table D.3: Question 6: Overall results use cases rankings

<table>
<thead>
<tr>
<th>Rank</th>
<th>Use Case</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>U1. Structure of the process</td>
<td>2.3</td>
</tr>
<tr>
<td>2</td>
<td>U7. Compliance to the explicit process</td>
<td>5.9</td>
</tr>
<tr>
<td>3</td>
<td>U3. Most frequent path in the process</td>
<td>7.1</td>
</tr>
<tr>
<td>4</td>
<td>U4. Distribution of cases over paths</td>
<td>7.2</td>
</tr>
<tr>
<td>5</td>
<td>U2. Routing probabilities</td>
<td>8.2</td>
</tr>
<tr>
<td>6</td>
<td>U5. Exceptions from the normal path</td>
<td>8.3</td>
</tr>
<tr>
<td>7</td>
<td>U6. The degree in which the rules are obeyed</td>
<td>8.7</td>
</tr>
<tr>
<td>8</td>
<td>U15. Cycles</td>
<td>8.8</td>
</tr>
<tr>
<td>9</td>
<td>U12. Throughput time of cases</td>
<td>9.4</td>
</tr>
<tr>
<td>11</td>
<td>U18. Time sequence of events</td>
<td>11.5</td>
</tr>
<tr>
<td>12</td>
<td>U13. Slowest activities</td>
<td>11.6</td>
</tr>
<tr>
<td>13</td>
<td>U14. Longest waiting times</td>
<td>11.9</td>
</tr>
<tr>
<td>14</td>
<td>U17. Resource utilization rate</td>
<td>12.4</td>
</tr>
<tr>
<td>15</td>
<td>U10. Work handovers</td>
<td>12.5</td>
</tr>
<tr>
<td>16</td>
<td>U8. Resources per task</td>
<td>12.6</td>
</tr>
<tr>
<td>17</td>
<td>U9. Resources involved in a case</td>
<td>12.7</td>
</tr>
<tr>
<td>18</td>
<td>U11. Central employees</td>
<td>12.8</td>
</tr>
<tr>
<td>19</td>
<td>U16. Arrival rate of cases</td>
<td>13.8</td>
</tr>
</tbody>
</table>

Figure D.4: Use cases ranking results based on respondents activity domains
Figure D.5: Question 7: Do you think there is any use case missing?

<table>
<thead>
<tr>
<th>Use case</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Measure different KPIs</strong></td>
<td></td>
</tr>
<tr>
<td>Measure key performance indicators (costs, quality, flexibility)</td>
<td>1</td>
</tr>
<tr>
<td>Calculating costs based on events (Activity Based Costing etc.)</td>
<td>1</td>
</tr>
<tr>
<td>Analysis of attributes (Costs)</td>
<td>1</td>
</tr>
<tr>
<td><strong>Build a simulation model</strong></td>
<td></td>
</tr>
<tr>
<td>Build simulation models</td>
<td>3</td>
</tr>
<tr>
<td>Use process mining in combination with simulation for forward decisional support analysis</td>
<td>1</td>
</tr>
<tr>
<td><strong>Use model to make predictions</strong></td>
<td></td>
</tr>
<tr>
<td>Predicting the result of a process</td>
<td>1</td>
</tr>
<tr>
<td>Analyze an event log online to make predictions for the future</td>
<td>1</td>
</tr>
<tr>
<td><strong>Preprocessing</strong></td>
<td></td>
</tr>
<tr>
<td>Trace clustering</td>
<td>1</td>
</tr>
<tr>
<td>Filtering irrelevant cases out of the event log</td>
<td>2</td>
</tr>
</tbody>
</table>
Figure D.6: Question 8: How many of the processes in your area contain parallelism?

Figure D.7: Question 9a: Typical process - Number of cases (process instances) per year
Table D.5: Question 10: Which additional functionalities do you consider important for a process mining tool?

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input and output capabilities</strong></td>
<td></td>
</tr>
<tr>
<td>Export to excel</td>
<td></td>
</tr>
<tr>
<td>XPDL export</td>
<td></td>
</tr>
<tr>
<td>Import/export enhancements are always welcome</td>
<td></td>
</tr>
<tr>
<td>Export capabilities for figures and results</td>
<td></td>
</tr>
<tr>
<td>Easy export of output, both graphical as results</td>
<td></td>
</tr>
<tr>
<td>Event log to EPC or BPMN export results as pdf or html</td>
<td>13</td>
</tr>
<tr>
<td>Export to BPM tools</td>
<td></td>
</tr>
<tr>
<td><strong>Filter and cluster data</strong></td>
<td>3</td>
</tr>
<tr>
<td>Clustering to identify similar patterns in the process</td>
<td></td>
</tr>
<tr>
<td>Intelligent filters</td>
<td></td>
</tr>
<tr>
<td><strong>Integration with external systems</strong></td>
<td>10</td>
</tr>
<tr>
<td>Plugins for existing BPMS, such as Oracle</td>
<td></td>
</tr>
<tr>
<td>Database connection</td>
<td></td>
</tr>
<tr>
<td>Integration with statistical tools (for example: R)</td>
<td></td>
</tr>
<tr>
<td>Integrated data mining tools to analyze the patterns in a process</td>
<td></td>
</tr>
<tr>
<td>Integration with ERP, CRM, etc systems</td>
<td></td>
</tr>
<tr>
<td><strong>Animation capabilities</strong></td>
<td>2</td>
</tr>
<tr>
<td>Replay</td>
<td></td>
</tr>
<tr>
<td>See the animation of a process</td>
<td></td>
</tr>
<tr>
<td><strong>Support for large input event logs</strong></td>
<td>2</td>
</tr>
<tr>
<td>Visualization for the large-volume of process instance data</td>
<td></td>
</tr>
<tr>
<td>Support for high amount of cases</td>
<td></td>
</tr>
</tbody>
</table>
Appendix E

Generated Event Logs

Table E.1: Event log E1 - Sequence

<table>
<thead>
<tr>
<th># of cases</th>
<th># of events</th>
<th>Trace</th>
<th>% trace</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>3000</td>
<td>A-B-C</td>
<td>100%</td>
</tr>
</tbody>
</table>

Table E.2: Event log E2 - XOR split

<table>
<thead>
<tr>
<th># of cases</th>
<th># of events</th>
<th>Trace</th>
<th>% trace</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>3000</td>
<td>A-B-D</td>
<td>51.7%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-C-D</td>
<td>58.3%</td>
</tr>
</tbody>
</table>

Table E.3: Event log E3 - OR split

<table>
<thead>
<tr>
<th># of cases</th>
<th># of events</th>
<th>Trace</th>
<th>% trace</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>3109</td>
<td>A-C-D</td>
<td>26%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-C-D</td>
<td>25.4%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-C-B-D</td>
<td>24.4%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-D</td>
<td>24.2%</td>
</tr>
</tbody>
</table>

Table E.4: Event log E4 - Parallel activities

<table>
<thead>
<tr>
<th># of cases</th>
<th># of events</th>
<th>Trace</th>
<th>% trace</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>4000</td>
<td>A-B-C-D</td>
<td>51.3%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-C-B-D</td>
<td>58.7%</td>
</tr>
</tbody>
</table>
Figure E.1: Highly parallel process model with random activity durations

Figure E.2: Highly parallel process model with constant activity durations

Table E.5: Event log E5 - Structured loop - Repeat

<table>
<thead>
<tr>
<th># of cases</th>
<th># of events</th>
<th>Trace</th>
<th>% trace</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>5520</td>
<td>A-B-C-D</td>
<td>61.6%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-C-B-C-B-C-D</td>
<td>24.9%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-C-B-C-B-C-B-C-D</td>
<td>7%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-C-B-C-B-C-B-C-B-C-D</td>
<td>4%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-C-B-C-B-C-B-C-B-C-B-C-D</td>
<td>1%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-C-B-C-B-C-B-C-B-C-B-C-D</td>
<td>0.8%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-C-B-C-B-C-B-C-B-C-B-C-D</td>
<td>0.7%</td>
</tr>
</tbody>
</table>
Table E.6: Event log E6 - Structured loop - Do while

<table>
<thead>
<tr>
<th># of cases</th>
<th># of events</th>
<th>Trace</th>
<th>% trace</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>4733</td>
<td>A-B-D</td>
<td>53.6%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-C-B-D</td>
<td>27%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-C-B-C-B-D</td>
<td>9.2%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-C-B-C-B-C-B-D</td>
<td>5.2%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-C-B-C-B-C-B-C-B-D</td>
<td>2.1%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-C-B-C-B-C-B-C-B-C-B-D</td>
<td>1.7%</td>
</tr>
</tbody>
</table>

Table E.7: Event log E7 - Arbitrary loop event

<table>
<thead>
<tr>
<th># of cases</th>
<th># of events</th>
<th>Trace</th>
<th>% trace</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>6108</td>
<td>A-B-D-E</td>
<td>29.7%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-C-E</td>
<td>24.5%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-D-E-F-D-E</td>
<td>13.7%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-C-E-F-D-E</td>
<td>11.2%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-D-E-F-D-E-F-D-E</td>
<td>6.2%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-C-E-F-D-E-F-D-E</td>
<td>5%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-D-E-F-D-E-F-D-E-F-D-E</td>
<td>2.5%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-C-E-F-D-E-F-D-E-F-D-E</td>
<td>2.3%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-D-E-F-D-E-F-D-E-F-D-E</td>
<td>1.3%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-C-E-F-D-E-F-D-E-F-D-E</td>
<td>1%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-D-E-F-D-E-F-D-E-F-D-E</td>
<td>0.8%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-D-E-F-D-E-F-D-E-F-D-E</td>
<td>0.7%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-D-E-F-D-E-F-D-E-F-D-E</td>
<td>0.6%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 Others</td>
<td>0.5%</td>
</tr>
</tbody>
</table>

Table E.8: Event log E8 - Skipped tasks

<table>
<thead>
<tr>
<th># of cases</th>
<th># of events</th>
<th>Trace</th>
<th>% trace</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>2500</td>
<td>A-C</td>
<td>50.2%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-C</td>
<td>49.8%</td>
</tr>
</tbody>
</table>

Table E.9: Event log E9 - Duplicate tasks

<table>
<thead>
<tr>
<th># of cases</th>
<th># of events</th>
<th>Trace</th>
<th>% trace</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>7000</td>
<td>A-B-C-E-F-B-G</td>
<td>51.5%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-C-D-F-B-G</td>
<td>48.5%</td>
</tr>
</tbody>
</table>

Table E.10: Event log E10 - Routing probabilities

<table>
<thead>
<tr>
<th># of cases</th>
<th># of events</th>
<th>Trace</th>
<th>% trace</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>6284</td>
<td>A-B-C-F-G-H</td>
<td>38.7%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-C-F-G-I</td>
<td>33.5%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-C-D-E-G-I</td>
<td>14.2%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-C-D-E-G-H</td>
<td>13.6%</td>
</tr>
</tbody>
</table>
Table E.11: Event log E11 - Frequent

<table>
<thead>
<tr>
<th># of cases</th>
<th># of events</th>
<th>Trace</th>
<th>% trace</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>6108</td>
<td>A-B-F-G-K</td>
<td>18.6%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-C-F-H-K</td>
<td>12.6%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-C-F-J-K</td>
<td>11%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-C-F-G-K</td>
<td>9.9%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-C-F-I-K</td>
<td>7.6%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-D-F-I-K</td>
<td>6.3%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-E-F-G-K</td>
<td>6%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-E-F-H-K</td>
<td>5.5%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-E-F-J-K</td>
<td>5.4%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-D-F-H-K</td>
<td>4.7%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-D-F-J-K</td>
<td>4.5%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-E-F-I-K</td>
<td>4.1%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-D-F-G-K</td>
<td>3.8%</td>
</tr>
</tbody>
</table>

Table E.12: Event log E12 - Conformance checking

<table>
<thead>
<tr>
<th># of cases</th>
<th># of events</th>
<th>Trace</th>
<th>% trace</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>7140</td>
<td>A-B-C-D-H-I-J</td>
<td>41.6%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-E-F-G-H-I-J</td>
<td>29.3%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-C-D-H-J</td>
<td>16.4%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-E-F-G-H-J</td>
<td>10.4%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-E-G-F-H-I-J</td>
<td>0.8%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-D-C-H-I-J</td>
<td>0.7%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-D-C-H-J</td>
<td>0.4%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-E-G-F-H-J</td>
<td>0.4%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Originator/Task</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
</tr>
</thead>
<tbody>
<tr>
<td>Susan</td>
<td>1000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Dexter</td>
<td>0</td>
<td>497</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Harry</td>
<td>0</td>
<td>503</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Fiona</td>
<td>0</td>
<td>0</td>
<td>211</td>
<td>207</td>
<td>142</td>
<td>143</td>
<td>123</td>
<td>0</td>
<td>0</td>
<td>301</td>
</tr>
<tr>
<td>Albert</td>
<td>0</td>
<td>0</td>
<td>194</td>
<td>203</td>
<td>123</td>
<td>124</td>
<td>131</td>
<td>0</td>
<td>0</td>
<td>345</td>
</tr>
<tr>
<td>Jim</td>
<td>0</td>
<td>0</td>
<td>209</td>
<td>204</td>
<td>121</td>
<td>119</td>
<td>132</td>
<td>0</td>
<td>0</td>
<td>354</td>
</tr>
<tr>
<td>Margaret</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Nancy</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>682</td>
<td>0</td>
</tr>
<tr>
<td># of cases</td>
<td># of events</td>
<td>Trace</td>
<td>% trace</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
<td>---------------------------</td>
<td>---------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>7140</td>
<td>A-B-C-D-H-I-J</td>
<td>42%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-E-F-G-H-I-J</td>
<td>29,6%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-C-D-H-J</td>
<td>18,2%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-E-F-G-H-J</td>
<td>12,9%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Originator/Task</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
</tr>
</thead>
<tbody>
<tr>
<td>Susan</td>
<td>1000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Dexter</td>
<td>0</td>
<td>497</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Harry</td>
<td>0</td>
<td>503</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Fiona</td>
<td>0</td>
<td>0</td>
<td>211</td>
<td>207</td>
<td>142</td>
<td>143</td>
<td>123</td>
<td>0</td>
<td>0</td>
<td>301</td>
</tr>
<tr>
<td>Albert</td>
<td>0</td>
<td>0</td>
<td>194</td>
<td>203</td>
<td>123</td>
<td>124</td>
<td>131</td>
<td>0</td>
<td>0</td>
<td>345</td>
</tr>
<tr>
<td>Jim</td>
<td>0</td>
<td>0</td>
<td>209</td>
<td>204</td>
<td>121</td>
<td>119</td>
<td>132</td>
<td>0</td>
<td>0</td>
<td>354</td>
</tr>
<tr>
<td>Margaret</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Nancy</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>682</td>
<td>0</td>
</tr>
</tbody>
</table>
Table E.14: Event log E14 - Cycles

<table>
<thead>
<tr>
<th># of cases</th>
<th># of events</th>
<th>Trace</th>
<th>% trace</th>
<th>Avg throughput time (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>4733</td>
<td>A-B-C-E-H-F-G-I-J</td>
<td>15.3%</td>
<td>1.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-D-E-F-H-G-I-J</td>
<td>15%</td>
<td>1.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-D-E-H-F-G-I-J</td>
<td>13.9%</td>
<td>2.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-C-E-F-H-G-I-J</td>
<td>13.4%</td>
<td>2.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-D-E-H-F-G-I-H-F-G-I-J</td>
<td>1.9%</td>
<td>2.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-C-E-H-F-G-I-F-H-G-I-J</td>
<td>1.9%</td>
<td>2.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-C-E-B-D-E-H-F-G-I-J</td>
<td>1.7%</td>
<td>2.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-D-E-B-C-E-F-H-G-I-J</td>
<td>1.7%</td>
<td>2.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-D-E-B-C-E-H-F-G-I-J</td>
<td>1.7%</td>
<td>2.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-D-E-H-F-H-G-I-F-H-G-I-J</td>
<td>1.6%</td>
<td>3.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-C-E-F-H-G-I-F-H-G-I-J</td>
<td>1.5%</td>
<td>3.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-D-E-H-F-G-I-F-H-G-I-J</td>
<td>1.5%</td>
<td>3.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-C-E-B-C-E-F-H-G-I-J</td>
<td>1.4%</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-D-E-B-D-E-H-F-G-I-J</td>
<td>1.4%</td>
<td>3.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-C-E-F-H-G-I-H-F-G-I-J</td>
<td>1.4%</td>
<td>3.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-C-E-B-C-E-H-F-G-I-J</td>
<td>1.3%</td>
<td>3.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-C-E-B-D-E-F-H-G-I-J</td>
<td>1.2%</td>
<td>2.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-D-E-B-D-E-F-H-G-I-J</td>
<td>1%</td>
<td>2.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-D-E-A-B-D-E-H-F-G-I-J</td>
<td>1%</td>
<td>3.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-C-E-H-F-G-I-H-F-G-I-J</td>
<td>1%</td>
<td>3.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-D-E-A-B-C-E-H-F-G-I-J</td>
<td>1%</td>
<td>3.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-C-E-A-B-D-E-H-F-G-I-J</td>
<td>0.8%</td>
<td>3.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-C-E-A-B-D-E-F-H-G-I-J</td>
<td>0.7%</td>
<td>3.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-D-E-A-B-D-E-F-H-G-I-J</td>
<td>0.6%</td>
<td>3.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-C-E-A-B-C-E-F-H-G-I-J</td>
<td>0.5%</td>
<td>3.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-C-E-A-B-C-E-H-F-G-I-J</td>
<td>0.5%</td>
<td>3.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-D-E-A-B-C-E-F-H-G-I-J</td>
<td>0.5%</td>
<td>3.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>others</td>
<td>12.6%</td>
<td>Time increases until 6.3</td>
</tr>
</tbody>
</table>
Table E.15: Event log E15 - Data perspective

<table>
<thead>
<tr>
<th># of cases</th>
<th># of events</th>
<th>Trace</th>
<th>% trace</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>6284</td>
<td>A-B-C-F-G-H</td>
<td>38.7%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-C-F-G-I</td>
<td>33.5%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-C-D-E-G-I</td>
<td>14.2%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-B-C-D-E-G-H</td>
<td>13.6%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Decision Point</th>
<th>Attribute used</th>
<th>Value of the attribute</th>
<th>Path followed</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Value damage</td>
<td>&lt;5000</td>
<td>D-E</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&gt;5000</td>
<td>F</td>
</tr>
<tr>
<td>G</td>
<td>History damages</td>
<td>&lt;15000</td>
<td>H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&gt;15000</td>
<td>I</td>
</tr>
</tbody>
</table>
Appendix F

Use Case-based Evaluation Results

Use case 1: Structure of the process

Description: Determine the structure of an unknown process or discover how a process looks like in practice.

Sub use case 1: Sequential activities

Acceptance criteria: The tools supporting this sub use case provide a control flow representation that suggests that a set of activities is executed in sequential order, meaning that one activity cannot start before the previous one is completed.

Event log used: Event log E1 (Table E.1). The event log consists of a sequence of three activities: A-B-C.

ARIS PPM Both the EPC and the activity sequence mining approaches support the Sequential activities sub use case. Figure F.1 depicts the EPC representation of the process model discovered.

Figure F.1: ARIS PPM supports the discovery of sequential activities
**Futura Reflect** The ability of discovering sequential activities was tested using both the *Mine* and the *Explore* functionalities. In both cases, the sequence of activities A-B-C (Figures F.2(a) and F.2(b)) was discovered. Therefore, the **Sequential activities** sub use case is **supported** by the system.

![Sequential activities - Futura Reflect - Mine](image1)

(a) Sequence - Futura Reflect - Mine

![Sequential activities - Futura Reflect - Explore](image2)

(b) Sequence activities - Futura Reflect - Explore

Figure F.2: Futura Reflect supports the discovery of sequential activities in both miners

**ProcessAnalyzer** The sequence of activities A-B-C (Figure F.3) was discovered based on the information from the event log. The **Sequential activities** sub use case is thus **supported**.

![Sequential activities - ProcessAnalyzer](image3)

Figure F.3: ProcessAnalyzer supports the discovery of sequential activities

**Sub use case 2: XOR splits and joins**

**Acceptance criteria:** The tools supporting this sub use case provide a control flow representation that suggests that after the completion of an activity that is considered to be a XOR split, the thread of control is passed to exactly one of the outgoing branches. The convergence of the multiple branches into one subsequent branch is, then, done by executing an activity, considered to be a XOR join.

**Event log used:** Event log E2 (Table E.2). The event log consists of activity A, followed by the possibility of executing one of the activities B or C, and activity D at the end.

**ARIS PPM** The XOR split pattern can be discovered by both mining approaches implemented in ARIS PPM (EPC and activity sequence). Figure F.4 shows the EPC process model. Therefore, the system **supports** the **XOR splits and joins** sub use case.

**Futura Reflect** The **XOR splits and joins** sub use case is **supported** by both *Mine* and *Explore* process discovery functionalities. Both discovered models (Figures F.5(a) and F.5(b)) depict an exclusive choice construct after activity A.
APPENDIX F. USE CASE-BASED EVALUATION RESULTS

Figure F.4: ARIS PPM supports the discovery of XOR splits and joins

(a) XOR split - Futura Reflect - Mine  (b) XOR split - Futura Reflect - Explore

Figure F.5: Futura Reflect supports the discovery of XOR splits and joins in both miners

**ProcessAnalyzer** The process model represented in Figure F.6 contains an exclusive choice pattern after activity A. The **XOR splits and joins** sub use case is thus supported by ProcessAnalyzer.

**Sub use case 2: OR splits and joins**

**Acceptance criteria:** This sub use case is supported if the system provides a control flow representation that suggests the fact that after the completion of an activity, considered to be the OR split, the thread of control can be passed to one or more of the outgoing branches. The convergence of the multiple branches into one subsequent branch is, then, done by executing an activity, considered to be a OR join.

**Event log used:** Event log E3 (Table E.3). The event logs consists of traces starting with activity A,
Figure F.6: ProcessAnalyzer supports the discovery of XOR splits and joins

followed by the possibility of executing any combination of activities B and C (B, C, or both B and C), and ending with activity D.

**ARIS PPM** The OR splits and joins sub use case is not supported. The discovered model (Figure F.7) suggest that activity C, for instance, can be either triggered by activity A, either by B, which is not the case in practice. The executions of activities B and C are independent and are enabled only after the completion of activity A.

Figure F.7: ARIS PPM does not support the discovery of OR splits and joins

**Futura Reflect** The ability of discovering the OR splits construct was checked for both Mine and Explore functionalities. None of the discovered models (Figures F.8(a) and F.8(b)) contains a control flow representation suggesting that any combination of activities B and C can be executed. Instead,
the models try to cover all paths from the event log. Therefore, the **OR splits and joins** sub use case is not supported.

![Diagram](image1.png)

(a) OR split - Futura Reflect - Mine
(b) OR split - Futura Reflect - Explore

Figure F.8: Futura Reflect does not support the discovery of OR splits and joins in any of the miners

**ProcessAnalyzer** The discovered model contains no control flow representation capturing the ability of executing activity B only, activity C only, or both of them. Instead, the models try to cover all paths from the event log. The conclusion is that the **OR splits and joins** sub use case is not supported.

![Diagram](image2.png)

Figure F.9: ProcessAnalyzer does not support the discovery of OR splits and joins

**Sub use case 4: Parallel activities**

**Acceptance criteria:** The tools supporting this sub use case provide a control flow representation suggesting the divergence of a branch into multiple branches that execute concurrently.

**Event log used:** Event log E4 (Table E.4). The event log consists of traces starting with activity A, followed by the sequences B-C or C-B (activities B and C are executed in parallel, therefore), ending with activity D.

**ARIS PPM** The discovery of **Parallel activities** is supported by ARIS PPM in the EPC representation (Figure F.10). There is, however, the condition that the activities involved in the concurrent construct need to have the same start and completion timestamps. The representation using activity sequences does not support this sub use case.
Futura Reflect The **Parallel activities** sub use case is supported by Mine, but not by Explore. The model built by the Explore miner (Figure F.11(b)) shows no parallelism, but a combination of the two traces in the input event log. The algorithm used by Mine is, however, able to detect that activities B and C can be performed in parallel (Figure F.11(a)). The use of the special “+” symbol in the boxes representing activities A and D denotes a AND split, respectively AND join, semantic.

ProcessAnalyzer The model discovered (Figure F.12) contains no representation that might suggest
that the execution of activities B and C is concurrent in practice. Instead, the model covers the two possible paths from the event log (A-B-C-D and A-C-B-D) by displaying the exact two sequences. Moreover, the model allows for additional behavior (e.g. A-C-D, A-B-D, etc), not observed in the log. Therefore, the **Parallel activities** sub use case is not supported.

![Figure F.12: ProcessAnalyzer does not support the discovery of parallel activities](image)

**Sub use case 6: Structured loops - the “repeat” variant**

**Acceptance criteria:** The tools supporting this sub use case provide a control flow representation suggesting the possibility of executing an activity or a set of activities repeatedly. In particular for this sub use, from the discovered model it should be clear that the activities involved in the loop are executed at least once.

**Event log used:** Event log E5 (Table E.5). The event log consists of traces starting with activity A, followed by a repeat construct that involves activities B and C, and ending with activity D.

**ARIS PPM** In the EPC representation of the model discovered (Figure F.13) activities B and C can be executed multiple times, and are executed at least once, which suggests the occurrence of a repeat construct. The **Structured loops - the “repeat” variant** sub use case is, thus, supported.

**Futura Reflect** The capability of discovering the repeat variant of the structured loops was tested in both *Mine* and *Explore* miners. The discovered models (Figures F.14(a) and F.14(b)) show a loop that consists of activities B and C, both of them being executed at least once. We can conclude thus, that the **Structured loops - the “repeat” variant** sub use case is supported by both miners.

**ProcessAnalyzer** The discovered model (Figure F.15) shows a loop that consists of activities B and C, both of them being executed at least once. Therefore, the **Structured loops - the “repeat variant** sub use case is supported by the tool.

**Sub use case 6: Structured loops - the “do’while” variant**

**Acceptance criteria:** The tools supporting this sub use case provide a control flow representation suggesting the possibility of executing an activity or a set of activities repeatedly. In particular for this sub use, from the discovered model it should be clear that a subset of the activities involved in the loop can be skipped.

**Event log used:** Event log E6 (Table E.6). The event log consists of traces starting with activity A, followed by a do-while construct that involves activities B and C, and ending with activity D. After executing B it is possible to (re)enter the do-while construct by executing C, or to skip it by executing activity D.
Figure F.13: ARIS PPM supports the discovery of the “repeat” variant of structured loops

![Diagram showing ARIS PPM workflow](image)

(a) Repeat - Futura Reflect - Mine  
(b) Repeat - Futura Reflect - Explore

Figure F.14: Futura Reflect supports the discovery of the “repeat” variant of structured loops in both miners

![Diagram showing Futura Reflect workflow](image)

ARIS PPM  In the model discovered (Figure F.16) activities B and C can be executed multiple times,
APPENDIX F. USE CASE-BASED EVALUATION RESULTS

Figure F.15: ProcessAnalyzer supports the discovery of the “repeat” variant of structured loops but there is also the possibility of not executing activity C at all by performing activity D after B, which suggests the occurrence of a do while construct. Therefore, Structured loops - the “do while” variant sub use case is supported.

Figure F.16: ARIS PPM supports the discovery of the “do-while” variant of structured loops

Futura Reflect The discovered models (Figures F.17(a) and F.17(b)) show a loop that consists of activities B and C. In this case it is also visible that activity C can be skipped, by executing activity D immediately after B. This clearly marks the distinction between the repeat and the do-while variants. Therefore, one can conclude that the Structured loops - the “do while” variant sub use case is supported.

ProcessAnalyzer The discovered model (Figure F.18) shows a loop that consists of activities B and C. In this case, it is also visible that activity C can be skipped, by executing activity D immediately after B. The Structured loops - the “do while” variant sub use case is supported.

Sub use case 7: Arbitrary loops

Acceptance criteria: The tools supporting this sub use case provide the ability to discover processes that contain cycles which have more than one entry of exit point.
Figure F.17: Futura Reflect supports the discovery of the “do-while” variant of structured loops in both miners.

Figure F.18: ProcessAnalyzer supports the discovery of the “do-while” variant of structured loops.

**Event log used:** Event log E7 (Table E.7). The event log consists of traces starting with activity A, followed by the possibility of executing activities B or C. If B is executed, the trace continues with activities D and E. After executing E, there is the possibility of continuing the process by executing F, followed again by the sequence D-E or to stop. If C is executed, instead of B, then the next activity in the trace is E. The corresponding process represents, thus, an arbitrary loop, with two entry points (before and after executing D) and one exit point (after executing E).

**ARIS PPM** The model depicted in Figure F.19 presents the characteristics of an arbitrary loop. As such, the loop consists of executing activity E multiple times, the entry points in the loop are before and after executing activity D and the exit point is after executing E. The Arbitrary loops sub use case is thus supported.

**Futura Reflect** The models (Figures F.20(a) and F.20(b)) obtained after applying the two different discovery algorithms are identical and present the characteristics of an unstructured loop, namely multiple entry and exit points in and out the loop. The loop consists of executing activity E multiple times, the entry points in the loop are before and after executing activity D and the exit point is after executing E. Based on the models discovered by the two different mining algorithms, it can be concluded the Arbitrary loops sub use case is supported.

**ProcessAnalyzer** The model discovered (Figure F.21) depicts an arbitrary loop consisting of the possibility of executing activity E multiple times, with two entry points (before and after executing activity D) and one exit point (after executing activity E). Hence, the Arbitrary loops sub use case
Figure F.19: ARIS PPM supports the discovery of arbitrary loops

Figure F.20: Futura Reflect supports the discovery of arbitrary loops in both miners
is supported.

**Sub use case 8: Skipped activities**

**Acceptance criteria:** The tools supporting this sub use case have the ability to detect the possibility of skipping a certain activities or set of activities in discovered process.

**Event log used:** Event log E8 (Table E.8). The event log consists of traces starting with activity A, followed by activity B, or by activity C. If B is executed, then is C is executed afterwards.

**ARIS PPM** The model (Figure F.22) discovered shows the possibility of skipping the execution of task B. Therefore, the **Skipped activities** sub use case **is supported.**

**Futura Reflect** The models (Figures F.23(a) and F.23(b)) discovered by both *Mine* and *Explore* miners show the possibility of skipping the execution of task B. The **Skipped activities** sub use case is, thus, **supported.**
APPENDIX F. USE CASE-BASED EVALUATION RESULTS

Figure F.23: Futura Reflect supports the discovery of skipped activities in both miners

ProcessAnalyzer The model (F.24) discovered shows the possibility of skipping the execution of task B. Therefore, the Skipped activities sub use case is supported.

Figure F.24: ProcessAnalyzer supports the discovery of skipped activities

Sub use case 9: Duplicate activities

Acceptance criteria: The tools supporting this sub use case have the ability to differentiate between two activities having identical names, but being executed in completely different contexts. The result of this distinction is the representation in the discovered model of two separate activities.

Event log used: Event log E9 (Table E.9). The process that this event log corresponds to is a simple one, which contains two activities with the same name (B), one close to the beginning of the process and the other towards the end. The execution of the two activities in completely different contexts suggests there is no connection between them and that they actually represent completely different tasks.

ARIS PPM The model depicted in Figure F.25 shows a loop from activity F back to activity B. This is sign of the inability of distinguishing between the occurrences of the activity B in different contexts. As a consequence, the Duplicate activities sub use case is not supported.

Futura Reflect The models (Figures F.26(a) and F.26(b)) discovered by both the Mine and the Explore functionality show the inability of distinguishing between the occurrences of the activity labeled B
Figure F.25: ARIS PPM does not support the discovery of duplicate activities in different contexts. As a consequence of this inability, the models depict a loop back to activity B. The conclusion is that the **Duplicate activities** sub use case is **not supported** by the tool.

![Diagram](image)

(a) Duplicate activities - Futura Reflect - Mine  
(b) Duplicate activities - Futura Reflect - Explore

Figure F.26: Futura Reflect does not support the discovery of duplicate activities in any of the miners

**ProcessAnalyzer** The model (Figure F.27) discovered shows the inability of distinguishing between the occurrences of the activity labeled B in different contexts. This results in the occurrence of a loop back to activity B. The **Duplicate activities** sub use case is, therefore, **not supported**.
Use case 2: **Routing probabilities**

**Description:** Get a deeper understanding of the process by looking at the probabilities of following one path or another after a choice point.

**Event log used:** Event log E10 (Table E.10). The event log corresponds to a process with two XOR splits (activities C and G). After executing C, there is the possibility of performing activity F or the sequence D-E. After executing G, the possible activities to be executed are H and I.

**Acceptance criteria:** The tools supporting this use case should provide a mechanism of visualizing the routing probabilities specific to each choice point in the process. Each outgoing arc of a choice point has associated such a routing probability.

**ARIS PPM** In the EPC representation, an arc connecting two activities has associated the probability that a case follows it, based on the total number of cases and the number of cases following that arc (Figure F.28). In the activity sequence representation, it is also possible to display this type of information on the arcs. Therefore, the **Routing probabilities** use case is supported.

**Futura Reflect** Only the Explore functionality provides the possibility of displaying different kind of information as arc labels. In order to show the percentage of cases following one arc or another in the process, one needs to check the “Show performance metrics” box, to select “Count” as one of the arc metrics and to check the “As percentage” box. This will display on every arc the percentage of cases following it (Figure F.29). For the activities that constitute choice points in the process, these percentages represent routing probabilities. One can thus conclude that the **Routing probabilities** use case is supported.

**ProcessAnalyzer** By enabling the “Show flow labels” checkbox from the “Visibility settings” pane, for every arc in the model the percentage of cases following it is displayed (Figure F.30). For the activities that constitute choice points in the process, these percentages represent routing probabilities. The **Routing probabilities** use case is therefore supported.

Use case 3: **Most frequent path in the process**

**Description:** Discover the path in the process that is followed by the highest percentage of cases. A path is considered to be a sequence of activities.

**Event log used:** Event log E11 (Table E.11). The event log consists of traces starting with activity A. After executing A, one of the activities B, C, D, E can be exclusively executed. The following
Figure F.28: ARIS PPM supports the Routing probabilities by displaying the proportion of cases following each arc

Figure F.29: Futura Reflect by displaying the percentage of cases following each arc
Appendix F. Use Case-based Evaluation Results

Figure F.30: ProcessAnalyzer by displaying the percentage of cases following each arc.

Activity in the traces is F. After the execution of F, one of the activities G, H, I, J can be exclusively executed. The traces end with activity K. The most frequent path in the process is A-B-F-G-K.

Acceptance criteria: The tools supporting this pattern should provide the possibility of highlighting or pointing out which is the most frequent path in the process.

ARIS PPM ARIS PPM provides the possibility of highlighting “the most probable path” on a mined model. In our case, it suggests that this path is A-C-F-G-K (Figure ??), which according to the description of event log used is not the most frequent path in the process. The conclusion is that the computation of the “most probable path” is not done at the process level, but at the level of each split point. Therefore, the Most frequent path in the process use case is not supported.

Figure F.31: ARIS PPM does not highlight the most frequent path at the process level, but at the level of each XOR split.
**Futura Reflect** In the "Explore" functionality there is the possibility of representing arcs with different thicknesses in the sense that the more frequent paths (arcs) are thicker than the less frequent ones. Activities are as well symbolized with different colors, based on the number of their occurrence in the event log. A purple box suggests that the activity it represents is more frequent than an activity represented through a light purple/blue box. Additionally, the software that computes the discovered model tries to lay out the model such that the most frequent path is a straight line, and the rest is laid out around it. This fact is also visible in Figure ??, where the central line of the model is exactly the most frequent path (A-B-F-G-K). Having this considered, it can be argued that the **Most frequent path in the process** use case is supported.

![Futura Reflect model](image)

**ProcessAnalyzer** The arcs in the model are represented with different thicknesses in the sense that the more frequent paths (arcs) are thicker than the less frequent ones. Activities are as well symbolized with different colors, based on the number of their occurrence in the event log. However, in the current case the most frequent path is not the sequence of the most frequent arcs and the tool does not highlight the fact that the most frequent path is A-B-F-G-K (Figure F.33). Having this considered, it can be argued that the **Most frequent path in the process** use case is not supported.

**Use case 4: Distribution of cases over paths**

**Description:** Discover common and uncommon behavior in the process by looking at the distribution of cases over the possible paths in the process. A path is considered to be a sequence of activities.

**Event log used:** Event log E11 (Table E.11). The event log consists of traces starting with activity A. After executing A, one of the activities B, C, D, E can be exclusively executed. The following activity in the traces is F. After the execution of F, one of the activities G, H, I, J can be exclusively executed. The traces end with activity K. The most frequent path in the process is A-B-F-G-K.

**Acceptance criteria:** The tools supporting this pattern should have the ability to associate to each path in the process the percentage or the number of cases that follow it. Both possibilities are accepted.
Figure F.33: ProcessAnalyzer does not highlight the most frequent path at the process level, but at the level of each XOR split.

**ARIS PPM** The tool provides no mechanism of visualizing the number of cases or the percentage of cases follow the paths in the process. The **Distribution of cases over paths** use case is thus **not supported**.

**Futura Reflect** One can visualize statistics about the different paths in the process by pressing the “Download path information” button in the menu bar of the Explore miner. This saves a Microsoft Office Excel file containing information like the absolute frequency, the relative frequency of the activities of all paths identified in the process (Figure F.34). The tool offers thus **support** for executing the **Distribution of cases over paths** use case.

Figure F.34: Futura Reflect computes statistics about the different paths in the process.

**ProcessAnalyzer** Statistics about the different paths in the process are available in the **Variations** functionality. The results are depicted in Figure F.35. The **Distribution of cases over paths** use case is thus **supported**.

**Use case 5: Exceptions from the normal path**

**Description:** Discover the outliers of the process by looking at the least frequent behavior observed in practice. A path is considered to be a sequence of activities.
Event log used: Event log E12 (Table E.12). The event log stores for each trace, besides the name of the activities, also the identity of the originators of the activities. The most frequent behavior observed in the event log (97.7%) covers the traces A-B-C-D-H-I-J, A-B-E-F-G-H-I-J, A-B-C-D-H-J, and A-B-E-F-G-H-J. The rest of observed behavior (2.3%) consists of the traces A-B-E-G-F-H-I-J, A-B-D-C-H-I-J, A-B-D-C-H-J, and A-B-E-G-F-H-J.

Acceptance criteria: The tools supporting this pattern should be able to identify any parts of the process that might constitute exceptions from the normal behavior and then provide a mechanism of highlighting these parts in the process model. The decision whether a pattern in the process model represents an exception or not is based on comparing its frequency of occurrence with other frequent behaviors observed in the rest of the event log. The tool can either establish its own threshold to be used in classifying the patterns as frequent or infrequent based on the specific characteristics of the input event log or it can use a threshold value given by the user. If at least one approach is provided by a tool, the use case is supported.

ARIS PPM The identification of the exceptional paths is possible in the activity sequence representation. One can configure a filter to show only those connections below a certain user specified frequency (Figure F.36). Another possibility of solving this use case is by defining a KPI giving the relative frequency of transition and highlighting the low frequent cases (threshold specified by user). This suggests that the Exceptions from the normal path use case is supported.

Futura Reflect In both Mine and Explore functionalities, there is the possibility of adding a filter based on the frequency of paths, the “Unique path filter”. The result obtained by selecting a high percentage of cases using the slider and then by checking the option “Discard the selected percentage of cases”, is the set of cases that constitute the least frequent behavior. The algorithms employed by the two functionalities will discover a new process model based only on the information contained in the selected cases. By looking at the discovered model, one can visualize the paths representing exceptions from the rest of the observed behavior. For the input event log used, a selection of the least 2% infrequent cases displays the process models (Figures F.37(a) and F.37(b)), which contains indeed the least frequent behavior observed in the log. One can thus conclude that the Exceptions from the normal path use case is supported.

ProcessAnalyzer The tool provides no mechanism of highlighting or identifying the parts of the process that might constitute exceptions. Thus, the Exceptions from the normal path use case is not
Figure F.36: ARIS PPM can display only the relations with a frequency lower than a user specified frequency threshold supported.
Use case 6: The degree in which the rules are obeyed

Description: Check whether the rules and regulations related to the process are obeyed.

Event log used: Event log E12 (Table E.12). The event log stores for each trace, besides the name of the activities, also the identity of the originators of the activities. The most frequent behavior observed in the event log (97.7%) covers the traces A-B-C-D-H-I-J, A-B-E-F-G-H-I-J, A-B-C-D-H-J, and A-B-E-F-G-H-J. The rest of observed behavior (2.3%) consists of the traces A-B-E-G-F-H-I-J, A-B-D-C-H-I-J, A-B-D-C-H-J, and A-B-E-G-F-H-J.

Sub use case 1: First level

Acceptance criteria: The First level corresponds to the possibility of evaluating whether the checking of a specific type of rule is supported. The first level criterion consists of establishing the existence mechanisms capable of checking the compliance to the four-eye principle, also known as the separation of duties rule, the possibility of checking whether an activity has been executed after a certain amount of time after another activity and so on. To sum up, any hard coded functionality that enables the possibility of verifying a certain kind of rule is accepted.

Note: The First level criterion is automatically supported if the Second level criterion is supported.
ARIS PPM There are no hard coded features that check rules. All rules to be checked are configured when importing an event log. The First level criterion is, therefore, automatically supported (see note above).

Futura Reflect The first level criterion is supported by using the “Separation of Duties Violations” filter. By using this filter, with the option “Only check SoD violations within cases” enabled, the tool filters the cases that do not comply with the four-eye principle rule applied on activities B and H. Therefore, the First level sub use case is supported.

Sub use case 2: Second level

Acceptance criteria: In the Second level a more generic functionality is evaluated. This functionality refers to the possibility that users define their own rules to be checked.

ARIS PPM The tool supports the Second level sub use case. While making the configurations needed before the import of the event process, a rule can be defined and a dimension can associated to it (for instance, a boolean stating whether the rule is obeyed or not). This rule would be then checked for every process instance and the boolean dimension would be updated accordingly. If we consider an example in which the separation of duties rule is checked for activities B and H (The resource executing activity B cannot execute activity H in the same case), then possible problems can be identified by looking at the sequence diagram from Figure F.39(a) Next, it is possible to select the process instances for which the rules is not respected (Figure F.39(b)) or display the value of the boolean dimension (Figure F.39(c)) for all the instances.

Futura Reflect The metrics and filtering together allow a user to check for rules that are not hard coded in the tools. With the individual building blocks (several types of filters, several types of metrics), more complicated rules can be checked that were not entirely predefined. For example, it’s easy to check whether/which cases that were started on a Monday have a throughput time of more than 2 weeks, etc. This suggests the fact that the Second level sub use case is supported.
Figure F.39: ARIS PPM provides three different ways of checking the compliance with any rule specified at import time.

**ProcessAnalyzer** The tool provides no mechanism of defining any rules that might be checked for a process. Therefore, the Second level sub use case is not supported.

**Use case 7: Compliance to the explicit process**

**Description:** Compare the documented process model with the real process, either given by the event log or given by a explicit model.

**Event log used:** Event log E12 (Table E.12). The event log stores for each trace, besides the name of the activities, also the identity of the originators of the activities. The most frequent behavior observed in the event log (97.7%) covers the traces A-B-C-D-H-I-J, A-B-E-F-G-H-I-J, A-B-C-D-H-J, and A-B-E-F-G-H-J. The rest of observed behavior (2.3%) consists of the traces A-B-E-G-F-H-I-J, A-B-D-C-H-I-J, A-B-D-C-H-J, and A-B-E-G-F-H-J.

**Sub use case 1: Qualitative criterion**

**Acceptance criteria:** The tools supporting this pattern provide the possibility of comparing two process models (the discovered one and the explicit one) in a visual fashion and are able to highlight the parts of the process where differences occur.

**ARIS PPM** The Qualitative criterion sub use case is supported by using the visual comparison of two selected processes, which in this case can be the discovered process and the to-be one. Additional
features related to the qualitative criterion are available in the ARIS Modeling Tool.

**Futura Reflect** Figure F.40(a) depicts the explicit process model, while Figure F.40(b) represents the result of comparing the explicit model with the model discovered from the event log. The results are presented in a visual fashion, by building a process model that contains the reunion of the elements of the compared models. The arcs of the resulting process model are symbolized using different colors, as follows: light grey arcs are present in the reference model, but not in the discovered model, orange arcs are present in the discovered model, but not in the reference one, and light blue are belong to both models. The result of the comparison performed for the evaluation of this use case contains orange and light blue arcs, but the focus is to check whether indeed the orange arcs only belong to the discovered process model. By looking at the event log 12 and the reference model, depicted in figure F.40(a), one can observe, the possibility of executing the tasks C and D, respectively F and G in any order. This kind of behavior is not conform to the reference model, since there is only one possible order of performing the mentioned activities. The inconsistencies indicated in the comparison of the models are thus valid. One can conclude that the **Qualitative criterion is supported**.

![Diagram](image)

(a) Explicit model - Futura Reflect
(b) Compared model - Futura Reflect

Figure F.40: Futura Reflect compares two process models by merging them into one model and highlighting the arcs constituting differences.
**ProcessAnalyzer**  The tool provides no mechanism of visually comparing the discovered process model with the reference one. Therefore, the **Qualitative criterion is not supported**.

**Sub use case 2: Quantitative criterion**

**Acceptance criteria:** The tools supporting this pattern should provide the possibility of measuring the percentage of fit between the discovered process model and the event log, based on predefined metrics.

**Results:** None of the analyzed tools provides a mechanism of comparing a reference model to the discovered one by measuring the percentage of fit between them. The **Quantitative criterion is thus not supported**.

**Use case 8: Resources per task**

**Description:** Discover the relation between resources and tasks.

**Event log used:** Event log E13 (Table E.13). The event log stores for each trace, besides the name of the activities, also the identity of the originators of the activities and the activities timestamps. The possible traces are: A-B-C-D-H-I-J, A-B-E-F-G-H-I-J, A-B-C-D-H-J, and A-B-E-F-G-H-J.

**Acceptance criteria:** In order to support this use case, a tool should be able to output associations between tasks and resources based on the information stored in the event log. These associations are not specific to a case or set of cases, but apply to the entire process.

**ARIS PPM**  The table depicted in Figure F.41 shows the relations between the resources involved in the process and the activities of the process. The number in a box represents the number of times the activity (from the column) has been executed by the resource (from the row). Thus, the **Resources per task use case is supported**.

**Futura Reflect**  The **Resources per task** use case is supported by defining a chart. For this, one needs to select “Activity” for the X-axis attribute and enable the “Split on attribute” option where the Originator attribute needs to be chosen. On the data series one has to use “Count”. In the resulting chart (Figure F.42), one will see in the associations between resources and activities. The data behind the chart can be downloaded as an Excel file.

**ProcessAnalyzer**  Using the swimlane flowchart representation for the model discovered by the **Process Analysis** functionality, one can see the activities of the process organized into lanes based on the identity of their originator. However, if the event log contains the identity of individual resources
APPENDIX F. USE CASE-BASED EVALUATION RESULTS

Figure F.42: Futura Reflect displays associations between activities and their executors in a chart as originators and not the organizational unit, the representation brings limitations. For an activity that can be undertaken by more than one resource it is not possible to see the entire set of resources that are able to execute it, since in the model an activity is present only once and can hence appear in only one swimlane. As a consequence, the flowchart might incorrectly display empty swimlanes. Therefore, the Resources per task use case is not supported since the acceptance criteria states the need of having task-resource associations and not task-organizational unit associations.

Use case 9: Resources involved in a case

Description: Discover the group of resources involved in executing a particular case.

Event log used: Event log E13 (Table E.13). The event log stores for each trace, besides the name of the activities, also the identity of the originators of the activities and the activities timestamps. The possible traces are: A-B-C-D-H-I-J, A-B-E-F-G-H-I-J, A-B-C-D-H-J, and A-B-E-F-G-H-J.

Acceptance criteria: The tools supporting this pattern should present the group of resources involved in a specific selected process instance.

ARIS PPM In the “Process instances” table, by selecting one specific process instance, on the EPC diagram representing the control flow for that instance, one can also see the resources performing each task in the process (Figure F.44). The Resources involved in a case use case is supported.

Futura Reflect In the Overview functionality, in the “Detailed case information” tab, one can select any process instance and view its specific details. In order to see the group of resources involved in a particular case, before selecting the case from the list, the “Select attributes” button should be pressed and the “Originator” box should be checked in the prompted window. The result of this will be the display of the list of activities of the select process instances and the resources
ProcessAnalyzer does not output correct associations between activities and their executors due to the flowchart representation.

ARIS PPM displays the name of the originator under each activity in a process instance performing them. This suggests that the Resources involved in a case use case is supported.

Futura Reflect displays the name of the resources involved in a process instance in a table.

**ProcessAnalyzer** In the “Cases” sheet, every line represents a case from the input event log. Among other information related to time and activities, one can also read the set of resources involved in a specific process instance. There is a column for each resource in the process and
every time one resource worked on a certain case, there is a 1 in the cell corresponding to the case and to the resource. Therefore, one could conclude that the **Resources involved in a case** use case is supported.

![Figure F.46: ProcessAnalyzer displays the number of activities executed by each originator in a process instance](image)

**Use case 10: Work handovers**

**Description:** Manage resource location or determine possible causes for quality and time issues by looking at how work is transferred between resources.

**Event log used:** Event log E13 (Table E.13). The event log stores for each trace, besides the name of the activities, also the identity of the originators of the activities and the activities timestamps. The possible traces are: A-B-C-D-H-I-J, A-B-E-F-G-H-I-J, A-B-C-D-H-J, and A-B-E-F-G-H-J.

**Acceptance criteria:** A tool that supports this pattern is capable of constructing a social network of interactions between the resources involved in the execution of a process, based on the handovers of work that occur.

**ARIS PPM** The system is capable of constructing the social network of resources based on the handovers of work that take place using the activity sequence representation. On the arrows connecting the nodes (which represent the resources), the number of transfers of work is displayed (Figure F.47(a)). Moreover, the user also has the possibility to aggregate several nodes and display the resulting network (Figure F.47(b)). One can thus conclude that the **Work handovers use case** is **supported**.

**Futura Reflect** In the *Explore* functionality, in the “Options” pane, the “Attribute” should be set on “Originator” and the layout as “Social Network”. The result is a resource social network based on the handovers of work, depicted in Figure F.48. The thickness of the arcs is given by the number of transfers between two resources. Therefore, one can conclude that the **Work handovers use case** is **supported** in the *Explore* functionality.

**ProcessAnalyzer** The handovers of work between organizational units are visible by looking at the arrows connecting activities in the swimlane representation (Figure F.49). However, this representation brings the limitation that any activity, even the ones executed by multiple resources, can only be present in one swimlane, causing some of the swimlanes to be displayed as empty. Therefore, the information about the complete set of resources that can undertake it is lost. The conclusion is that the **Work handovers use case** is **not supported** by ProcessAnalyzer.
Use case 11: Central employees

**Description:** Determine who the central resources for a process are by analyzing the social network based on handovers of work.

**Event log used:** Event log E13 (Table E.13). The event log stores for each trace, besides the name of the activities, also the identity of the originators of the activities and the activities timestamps. The possible traces are: A-B-C-D-H-I-J, A-B-E-F-G-H-I-J, A-B-C-D-H-J, and A-B-E-F-G-H-J.

**Acceptance criteria:** The tools supporting this pattern should point out who are the central employees of a process based on number of work items transferred to them.
APPENDIX F. USE CASE-BASED EVALUATION RESULTS

Figure F.49: ProcessAnalyzer does not display the complete information about the handovers of work due to the limitations of the flowchart representation (e.g. empty swimlanes)

**ARIS PPM** The use case can be achieved by using KPIs. Based on the KPIs, a chart with the number of received handovers can be generated (Figure F.50). The employees with the highest number of work handovers transferred to them are the central employees of the process. The **Central employees** use case is, thus, **supported**.

Figure F.50: ARIS PPM displays a chart with the number of received handovers of each employee
Use case 12: Throughput time of cases

Description: Determine the time that passed since the start of a case in process until its completion.

Event log used: Event log E13 (Table E.13). The event log stores for each trace, besides the name of the activities, also the identity of the originators of the activities and the activities timestamps. The possible traces are: A-B-C-D-H-I-J, A-B-E-F-G-H-I-J, A-B-C-D-H-J, and A-B-E-F-G-H-J.

Acceptance criteria: The minimum requirement for supporting this use case is to provide the throughput time corresponding to each case in the process. Alternatives include the ability of providing the average throughput time of cases and the minimum and maximum values calculated. Both options are accepted.

ARIS PPM The throughput time of cases is represented by the KPI called “Process cycle time”. One can choose to display in the process instances table, next to the identifier of each process instance, also its throughput time (Figure 5.16). The use case Throughput time of cases is therefore supported.

Futura Reflect Using the Overview functionality, in the “Attributes/Metrics” tab, one can add attributes or metrics to the process instances in the event log. The user can select from a range of predefined attributes. By choosing the metric “Simple throughput time” from the “Throughput time” menu in the prompted window, information about the time passed since the beginning of a case until its completion is attached to it in the dataset. After this operation, the newly added metric and its minimum and maximum values are visible in the table containing all the attributes and metrics of the process (Figure F.52). This suggests that the Throughput time of cases use case is supported.

ProcessAnalyzer In the “Cases” sheet it is possible to see for each case in the event log the duration in hours, days or month (Figure F.53). By sorting one of the duration columns, one can also see the minimum and the maximum value of the throughput time and the corresponding cases. Therefore, the Throughput time of cases use case is supported.
APPENDIX F. USE CASE-BASED EVALUATION RESULTS

Use case 13: Slowest activities

**Description:** Discover potential time problems by looking at the activities in the process that take a long time.

**Event log used:** Event log E13 (Table F.13). The event log stores for each trace, besides the name of the activities, also the identity of the originators of the activities and the activities timestamps. The possible traces are: A-B-C-D-H-I-J, A-B-E-F-G-H-I-J, A-B-C-D-H-J, and A-B-E-F-G-H-J.

**Acceptance criteria:** A basic requirement for a tool that supports this use case is to be able to compute, based on the information stored in the event log, the average service time for each activity of the process. This criterion is used to decide the support of use case 13. Additionally, tools could also provide mechanisms of highlighting the activities that take longer than a certain amount of time. This amount of time can be either specified as input by the user or can be automatically determined by the system relative to the durations of all the activities of the process.

**ARIS PPM** The average duration of an activity in the process is given by the KPI called *Processing time (function)*. Figure F.54 depicts the graph of durations of all the activities in the process. This suggests that the Slowest activities use case is supported.

**Futura Reflect** Proceeding in a similar fashion to the steps described in Use case 12, one can define an additional metric representing the processing time of activities. The predefined metric is called “Processing time” and can be found in the menu with the same name. However, this is possible only provided that all activities in the event log have both a start and an end timestamp and when loading the dataset into the tool. After having defined the metric giving the processing time of an activity, the average specific to each activity can be visualized on the process model (Figure F.55) discovered by the *Explore* functionality, by selecting “Average” and “Processing time” on one of the performance metrics specific to activities. This implies, thus, that the Slowest activities use case is supported.
ProcessAnalyzer One way to obtain information about the average duration of each activity in the process is to enable the “Show activity duration” check box in the Process Analysis sheet. Then, the discovered model will display the average service time in the boxes representing activities (Figure F.56). Another way to analyze this information is by looking at the tables displayed under the discovered model on the same sheet. Thus, the Slowest activities use case is supported.

Use case 14: Longest waiting times

Description: Determine delays between activities by analyzing the waiting times before each activity.

Event log used: Event log E13 (Table E.13). The event log stores for each trace, besides the name of the activities, also the identity of the originators of the activities and the activities timestamps. The possible traces are: A-B-C-D-H-I-J, A-B-E-F-G-H-I-J, A-B-C-D-H-J, and A-B-E-F-G-H-J.

Acceptance criteria: The minimal requirement for a tool that supports this use case is to be able to compute, based on the information stored in the event log, the waiting times before each activity of the process. This criterion is used when evaluating the support for this use case. Next to this basic functionality, tools can also highlight the problem spots of the process caused by high waiting time values. A waiting time value can be considered high in relation to a user input threshold or in relation to a value automatically determined by the system.

ARIS PPM The waiting time is given by the KPI Function wait time. Figure F.57 depicts the graph presenting the waiting times before each activity. This implies that the Longest waiting times use case is supported.

Futura Reflect The metric giving the waiting times before activities can be obtained similarly with the metrics defined for use cases 12 and 13. The predefined metric corresponding to the waiting time is called “Idle time” and is located in the “Processing time” menu. In order to view the waiting times before activities, one needs to use the Explore miner and set one of the arcs metrics on “Average” and “Waiting time”. A waiting time displayed on an arc corresponds to the waiting time before the target activity of the arc (Figure F.58). It can be therefore concluded that the Longest waiting times use case is supported.

ProcessAnalyzer In the Process Analysis functionality, the arrows connecting activities have associated duration in days, hours, minutes or seconds (Figure F.59). This duration is the time that passes
APPENDIX F. USE CASE-BASED EVALUATION RESULTS

Figure F.57: ARIS PPM displays a chart with the waiting times before each activity

Figure F.58: Futura Reflect displays the waiting times on the arcs in the Explore miner between the two activities and can be therefore be interpreted as waiting time before the second activity. Thus, the Longest waiting times use case is supported.

Figure F.59: ProcessAnalyzer displays the waiting times on the arcs

Use case 15: Cycles

**Description:** Learn whether additional delays occur in the process due to cycles.

**Event log used:** Event log E14 (Table E.14). The event log belongs to a process in which multiple loops back are possible, leading to the repetition of several sequences of activities.

**Acceptance criteria:** The tools supporting this pattern should be able to identify the paths having throughput time higher than a given value or higher than a value automatically identified by the system based on the throughput times of other paths. Having identified these paths, the tool should check whether they contain any cycles (repetitions of a set of activities) and highlight them.

**ARIS PPM** Using the KPI that gives the throughput time of the cases (Process cycle time) it is possible to aggregate the values, creating intervals. The graph from the left of the figures displays the number of process instances that have the throughput time inside the intervals from the x-axis. By selecting the first bar (the smallest interval of throughput time and the highest number of process instances), one can visualize in the right side the EPC model corresponding to those process
instances (Figure F.60(a)). Figure F.60(b) depicts the situation in which the bar representing the cases with the highest throughput time is selected. The corresponding EPC model displays a loop back in the process, which is the root cause for the long throughput times. Therefore, the Cycles use case is supported.

**Futura Reflect** Filtering based on the “Throughput time” attribute is done by specifying the range of values to be included. In Explore model one can see the model built using the selected cases, model which included the loops causing the delays. By looking at the waiting times on the arcs, a user can get an idea of how much time is spent in a certain loop. Figure F.61 depicts the model and the corresponding “Throughput time” filter applied to the event log. The conclusion is that the Cycles use case is supported.

**ProcessAnalyzer** By filtering one of the duration columns from the “Case sheet”, one can include in the analysis only the cases with throughput times higher than a certain value. In the model depicted in Figure F.62 (corresponding to cases with throughput times higher than 5 hours), one can see the loops back in the process (from E to A or B, and from I to F). Therefore, the Cycles use case is supported.

### Use case 16: Arrival rate of cases

**Description:** Determine the frequency with which new cases arrive in the process.

**Event log used:** Event log E13 (Table E.13). The event log stores for each trace, besides the name of the activities, also the identity of the originators of the activities and the activities timestamps. The possible traces are: A-B-C-D-H-I-J, A-B-E-F-G-H-I-J, A-B-C-D-H-J, and A-B-E-F-G-H-J.

**Acceptance criteria:** The tools supporting this pattern should provide information about the distribution which describes the arrival pattern of cases entering the process.

**ARIS PPM** The graph depicted in Figure F.63 represents the evolution of the number of cases starting the process (given by the Starting frequency KPI on the y axis) in time. The Arrival rate of cases use case is, thus, supported.

**Futura Reflect** The system provides no possibility of identifying the arrival pattern of new cases. Therefore, the Arrival rate of cases use case is not supported.

**ProcessAnalyzer** The system provides no possibility of identifying the arrival pattern of new cases. Therefore, the Arrival rate of cases use case is not supported.

### Use case 17: Resource utilization rate

**Description:** Determine what are the utilization rates of the resource i.e, measure the fraction of time that a resource is busy.

**Event log used:** Event log E13 (Table E.13). The event log stores for each trace, besides the name of the activities, also the identity of the originators of the activities and the activities timestamps. The possible traces are: A-B-C-D-H-I-J, A-B-E-F-G-H-I-J, A-B-C-D-H-J, and A-B-E-F-G-H-J.

**Acceptance criteria:** The tools supporting this pattern should provide information about the utilization rate characteristic to each resource.
Figure F.60: ARIS PPM displays the EPC model corresponding to a set of process instances with throughput times in a certain interval.
Figure F.61: Futura Reflect displays in the Explore miner the process corresponding to the cases with the throughput times in a specified interval.

Figure F.62: ProcessAnalyzer displays the process model corresponding only to the cases with throughput times higher than a specified value.

**ARIS PPM** The graph depicted in Figure F.64 shows the fractions of time in which the resources were working (compared to the period of time between the start of the first case and the end of the last case from the event log). The **Resource utilization rate** use case is therefore, **supported**.

**Futura Reflect** The software offers no mechanism to compute the resources utilization rates, therefore the **Resource utilization rate** use case is **not supported**.

**ProcessAnalyzer** The tool can give the average resource utilization for a selected organizational unit on hourly, daily, monthly on early basis in **Resource analysis** functionality (Figure F.65). The **Resource utilization rate** use case is **thus supported**.

**Use case 18: Time sequence of events**

**Description:** Get a deeper understanding on the organization of a process by looking at the time sequence of activities for a specific case. (e.g. Gantt-graph for activities).

**Event log used:** Event log E13 (Table E.13). The event log stores for each trace, besides the name of the activities, also the identity of the originators of the activities and the activities timestamps. The

Acceptance criteria: In order to support this use case, a tool should provide a mechanism of representing the chronological sequence of the activities of a specific process instance. The diagram depicting this sequence captures information on both the start and end time of each activity and
Figure F.65: ProcessAnalyzer displays a chart with average resource utilization for a selected resource over time

makes it possible to see the overlapping activities.

**ARIS PPM** In the “Flowchart” tab one can visualize the timely sequence of events of a specific process instance (Figure F.66). Thus, the Timely sequence of events use case is supported.

**Futura Reflect** There is no possibility of visualizing the timely sequence of events of a specific process instance. Thus, the Timely sequence of events use case is not supported.

**ProcessAnalyzer** The Timely sequence of events use case is supported by the Timeline analysis functionality, by selecting a particular case and the option giving the absolute times (Figure F.67).

### Use case 19: Business rules

**Description:** Discover what are the process attributes that influence the choice points and what are the conditions for following one branch or another.

**Event log used:** Event log E15 (Table E.13). The event log stores for each trace, besides the name of the activities, also the identity of the originators of the activities, the activities timestamps and two additional attributes (“Value damage” and “History”). The possible traces are A-B-C-F-G-H, A-B-C-F-G-I, A-B-C-D-E-G-I, and A-B-C-D-E-G-H. The “Value damage” attribute influences
APPENDIX F. USE CASE-BASED EVALUATION RESULTS

Figure F.67: ProcessAnalyzer displays the timely sequence of events for a specific case

the decision taken after activity D (values lower than 5000 follow D and values higher than 5000 follow F), while the “History” attribute influences the decision taken after activity G (values lower than 15000 follow H and values higher than 15000 follow I).

Acceptance criteria: The tools supporting this pattern should highlight for each choice point in the process what are the event log attributes that are involved in taking the decision and what values correspond to following the possible paths.

Note: The acceptance criterion was initially stating the need of having as a result of this use case a decision tree, built using data mining functionalities. However, none of the evaluated tools provides such a feature. Nevertheless, two of the tools (ARIS PPM and Futura Reflect) tackle this type of analysis using a simpler and less powerful approach. We adapted the acceptance criteria accordingly and we consider the use case supported.

ARIS PPM Using the activity sequence of a process model, one can display different types of information on the arcs. Figure F.68(a) shows the average values of the “Value damage” attribute, while Figure F.68(b) shows the average value of the “History” attribute. If we look at the choice points C, respectively G, and at the large difference between the values of the corresponding attribute (“Value damage” in case of C and “History” in case of G), we can tell that the attribute has an influence on following one path or the other, as well as the range of values specific to each path. Therefore, the Business rules use case is supported.

Futura Reflect In the Explore functionality, there is the possibility of displaying the average value of an attribute under each activity in the model discovered. Figure F.69(a) depicts the model with the average values of the “Value damage” attribute displayed under the activity boxes, while in Figure F.69(b) one can see the average values of the “History” attribute. By looking at the average values of the two attributes at the choice points activities (C and G) and then at the values at the following activities (F and D, respectively H and I), one can get an idea related to the attribute influencing
Figure F.68: ARIS PPM displays the average value of an attribute for all cases following an arc. A high difference between the values of the attribute on two arcs following a choice point suggest that the routing decision is based on the attribute.

the choice, as well as the average of the values specific to subsequent each path. The conclusion is that the **Business rules** use case is **supported**.

**ProcessAnalyzer** There is no possibility of visualizing what are the attributes and the attribute values that influence a choice point. Therefore, the use case **Business rules is not supported**.
Figure F.69: Futura Reflect displays the average value of an attribute for all cases following an arc in the Explore miner. A high difference between the values of the attribute on two arcs following a choice point suggest that the routing decision is based on the attribute.