Eindhoven University of Technology
Department of Mathematics and Computer Science
Architecture of Information Systems Research Group

Master's Thesis

A Comparative Study for Process Mining Approaches in a Real-life Environment

Graciëla E. Hoogendoorn

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Business Information Systems

Supervisors: prof. dr. ir. H.A. Reijers, ir. E. González López de Murillas

Examination Committee:

prof. dr. ir. H.A. Reijers
dr. ir. H. Eshuis
ir. E. González López de Murillas
ir. P. Vos (Experis Ciber)

Eindhoven, January 2017

Failure is an option here. If things are not failing, you are not innovating enough.

Elon Musk

Abstract

This study compares three process mining approaches to complete a process mining project in real-life. The motivation for this comparison is driven by the manifestation of a new technique to create event logs, namely through the extraction of redo logs of a Data Base Management System.

We successfully applied this new technique with PADAS (Process Aware Data Suite) for the first time in a real-life environment. The resulting event log of the 'redo log process mining approach' was also verified by comparing it to an event log extracted with traditional techniques (i.e. a query on several tables). We found that these two event logs are equivalent.

However, to fairly compare complete process mining approaches to address real-life process mining projects with each other, no existing methodology existed. As such, this work defined and proposes a comparison framework for process mining approaches in real-life environments. The comparison framework uses quantitative and qualitative metrics to assess the actual and perceived efficacy of the execution of a task with a particular method. Tasks are defined in a structured manner through the application of the Process Mining Project Methodology, i.e. PM^2.

Accordingly, we evaluated the comparison framework by applying it to a case study. In total, we compared three process mining approaches to complete a project in real-life. First, the traditional process mining approach, widely applied in academic environments. Second, the redo log process mining approach. Third, the approach using the off-the-shelf software suite for process mining from Celonis, which targets business oriented users. As such, we applied each of these process mining approaches on the same process mining project. Simultaneously we adapted the comparison framework.

Our findings do not clearly point to one overall best-performing process mining approach but indicate that the best process mining approach should be determined based on the context of the process mining project. The choice depends for example on, the process mining experience of the process mining team, the time that is available for the project, the structure of the database, and the desired output for the analysis.

Preface

This master thesis is the result of my graduation project, which completes my Master in Business Information Systems. The work in this thesis has been carried out under the auspices of the Technical University of Eindhoven in collaboration with Ciber Netherlands. The project was performed in the Architecture of Information Systems group of the Mathematics and Computer Science department.

For one, I would like to thank Hajo Reijers, for providing me this opportunity and making me exceed my expectations. Next, many thanks go out to Eduardo González López de Murillas for always having his door open and providing me with guidance throughout the project. And also I would like to thank Peter Vos for all the valuable feedback and fruitful discussions. To all three of my supervisors, I would like to say that I sincerely appreciate all their efforts for guiding me on this project.

Additionally I would like to thank Kris van Tienhoven for his continuous help and optimism. Plus I want to thank Edith Rook for her critical point of view on my English. I would also like to thank Bing Lin, who has supported and motivated me all the way. Altogether, I am grateful to all my fellow students and friends that provided me with feedback and for the nice coffee breaks with them.

And with extra emphasis, I would like to thank my father, Simon Hoogendoorn for his care, encouragements and always being there. Also, I would like to thank my mother. Finally, my acknowledgments, but also encouragements go out to my brother and sister.

Contents

Contents **v**

List of Figures **vii**

List of Tables **xi**

1 Introduction **1**
1.1 The Emergence of Process Mining . . . 2
1.2 Significance of this Study . . . 2
1.3 Goal of this Study . . . 4
1.4 Scoping . . . 5
1.4.1 Description of the Case Study . . . 5

2 Preliminaries **7**
2.1 Background . . . 7
2.2 Related Work . . . 10
2.2.1 Adoption in Practice & Pragmatic Success . . . 10
2.2.2 Methodologies for a Process Mining Project . . . 11

3 Methodology **13**
3.1 Comparison Framework for Process Mining Approaches . . . 13
3.2 Measure the Actual Efficacy for Stage 2 of PM^2 . . . 17
3.3 Measure the Actual Efficacy in Stage 3 of PM^2 . . . 19
3.4 Measure the Actual Efficacy in Stage 4 of PM^2 . . . 23
3.5 Adoption of a Process Mining Approach in Practice . . . 25

4 Verification **26**
4.1 Requirements for Deployment . . . 26
4.2 Dealing with Missing Archived Logs . . . 26
4.3 Verification of the Redo Log Process Mining approach . . . 27

5 Summary Case Study Analysis **31**

6 Results **35**
6.1 Actual Efficiency of Process Mining Approaches . . . 35
6.1.1 Comparison per Input Measure . . . 35
6.1.2 Comparing PM approaches across all Input Measures . . . 38
6.2 Actual Effectiveness of Process Mining Approaches . . . 40
6.2.1 Actual Effectiveness in Stage 2: Extraction . . . 40
6.2.2 Actual Effectiveness in Stage 3: Data Processing . . . 43
6.2.3 Actual Effectiveness in stage 4: Mining & Analysis . . . 48
6.3 Perceived Efficacy of Process Mining Approaches . . . 52
6.4 Implications of the Schema Model Style of the Database . . . 53

6.5 Comparing the PM approaches Across Dimensions 54

7 Conclusion **56**
7.1 Summarizing the Results 56
7.1.1 Evaluating the Comparison Framework for PM Approaches 57
7.2 Recommendations for Future work 58
7.3 Redo Log Process Mining, Lessons Learned 60

Bibliography **61**

Appendix **65**

A Redo Log Process Mining in real-life Environments **65**
A.1 Archiving Process of Redo Logs 65
A.2 Feasibility check for Redo Log Process Mining (PADAS) 66
A.3 Script for automatically replacing the archived logs 67

B Case Study Analysis **68**
B.1 Completing Stage 1: Planning 68
B.2 Completing Stage 2: Extraction 74
B.2.1 Stage 2: Extraction with TPM 74
B.2.2 Stage 2: Extraction with RLPM 77
B.2.3 Stage 2: Extraction with CPM 79
B.2.4 Resulting measurements for Stage 2: Extraction 81
B.3 Completing Stage 3: Data Processing 82
B.3.1 Stage 3: Data Processing with TPM 82
B.3.2 Stage 3: Data Processing with RLPM 83
B.3.3 Stage 3: Data Processing with CPM 84
B.3.4 Resulting measurements for Stage 3: Data Processing 87
B.4 Completing Stage 4: Mining & Analysis 88
B.4.1 Stage 4: Mining & Analysis with TPM (equivalent to RLPM) 88
B.4.2 Stage 4: Mining & Analysis with CPM 94
B.4.3 Resulting measurements for Stage 4: Mining & Analysis 95
B.4.4 Findings for the Case Study 96
B.5 Completing Stage 5: Evaluation 97
B.6 Queries 98
B.7 Stage 2: Extraction, RLPM 101
B.8 Stage 3: Data Processing, CPM 102
B.9 Stage 4: Mining & Analysis, TPM - BPQ 2 103
B.10 Dashboards created with CPM - BPQ 1 105
B.11 Dashboards created with CPM - BPQ 2 108
B.12 Dashboards created with CPM - BPQ 3 110

C Results Comparative Study **113**
C.1 Actual Efficiency metric: Required Domain Knowledge 113
C.2 Comparison of Scores on Input Measures 116
C.3 Comparison of Scores on Output Measures 117
C.4 Comparison of Scores on Perceived Efficacy 118
C.4.1 Normalized scores for Perceived Efficacy 121

List of Figures

1.1 Process mining as the bridge between data science & process science. 1

2.1 Positioning of three main types of process mining: *discovery*, *conformance*, and *enhancement*. 7
2.2 Cases and events can have attributes and each event can be associated to a single case. 8
2.3 Logo ProM. 9
2.4 XES meta model with UML 2.- class diagram 9
2.5 Logo PADAS. 10
2.6 Logo CPM4. 10
2.7 Technology Acceptance Model. 10
2.8 Efficiency vs. effectiveness 11
2.9 An overview of the PM^2 methodology. 12

3.1 The three basic concepts of the comparison framework. 13
3.2 Method Evaluation Model. 15
3.3 Simplified illustration of PM^2, stages for comparison are highlighted and annotated with the logos of the PM approaches applied to that stage. 16
3.4 Operationalizing Moody's model for method evaluation with PM^2 to compare PM approaches. 16
3.5 Comparison framework applied to the second stage of PM^2, for the extraction. . . 17
3.6 Comparison framework for stage 3 of PM^2. 19
3.7 Overview of the activities in mining & analysis stage in PM^2. 23
3.8 Comparison framework for PM approaches applied to the fourth stage of PM^2. . . 24

4.1 Missing archived logs over time, where the greenly shaded time periods indicate the availabilty of archived logs and the white space indicates the gaps. 27
4.2 Extraction: in a with a SQL query for TPM and in b the trace ID pattern with PADAS. 27
4.3 Venn diagram of two sets of event data. 28
4.4 Process models mined with the Inductive Miner for the TPM and RLPM approach. 29

5.1 Process models of the completion of Stage 2: Extraction of PM^2 for the case study, with the TPM approach, the RLPM approach, and the CPM approach. 32
5.2 Process models of the completion of Stage 3: Data Processing of PM^2 for the case study, with the TPM approach, the RLPM approach, and the CPM approach. . . 33
5.3 Calculating the solve time taking into account a restricted service window. 34

6.1 Time plotted per PM approach for stage 2 and 3 of PM^2. 35
6.2 Number of iterations plotted for each PM approach. 36
6.3 The domain knowledge of PM approaches per stage. 36
6.4 The period of time covered by each process mining approach. 40
6.5 The number of events recorded per day for different process mining approaches. . . 40

6.6 Number of active cases and number of events, per day, in the event data of each PM approach. . . . 41
6.7 $V(L)$ of the event data of each PM approach. . . . 42
6.8 Number of unique data attributes per event data collection. . . . 43
6.9 M and SP for the event log of each PM approach. . . . 44
6.10 AE and LoD for the event log of each PM approach. . . . 44
6.11 $V(L)$, V_{min} and V_{max} for the event log of each PM approach. . . . 45
6.12 $UT(L)$ per PM approach. . . . 45
6.13 $ST(L)$, ST_{min} and ST_{max} for each PM approach. . . . 45
6.14 $MA(L)$, A_{min} and A_{max} for each PM approach. . . . 46
6.15 Process Comparator, comparing workflows of two event logs of the same process. One event log with tickets concerning customer 1 as group L_1 and one event log concerning customer 2 as group L_2. . . . 49
6.16 Model created with Inductive Visual Miner (96,0% of the activities and 80,2% of the paths). . . . 50
6.17 Process model created with Celonis (96% of the activities and 80,3% of the connections). . . . 51
6.18 The perceived efficacy of the process mining approaces, measured by the perceived ease of use and perceived usefulness scored on a seven-point Likert scale. . . . 52
6.19 A fragment of a Third Normal Form schema in a and a fragment of a star schema in b. . . . 53

7.1 Refinement of definition of stage 2 of PM^2 . . . 58

A.1 An illustration of how online redo log files are archived. . . . 65
A.2 Flow Chart for feasibility RLPM with PADAS (multitenant architecture Oracle RDBMS). 66
A.3 Script to automatically replace the archived logs to a separate storage location. . . 67

B.1 Two options for the setup for extraction with PADAS for the redo log process mining approach: a local installation in a and an installation on a server in b. . . . 73
B.2 The merging behaviour of tickets is logged in a separate table called `LINK_RELATION`, this can be visualized in a tree structure. In this simplified example, ticket 1 is the root, the parent node. Tickets 2 and 3 are the childeren of 1. This means that both ticket 2 and 3 are merged to ticket 1. . . . 76
B.3 The RapidMiner model that was created to merge two tables, one with event-data and one with case-data, in one event log. . . . 76
B.4 Configuration of the connection manager in PADAS: the connection settings a and the redo log settings in b. . . . 77
B.5 Creating a data model in PADAS. . . . 78
B.6 Create perspective with the Log splitter of PADAS in a and the visualized in PADAS in b. . . . 78
B.7 Configuring the connection to the database with CPM4. . . . 79
B.8 Manually defined relation between cases and events in CPM4. . . . 80
B.9 Filtering the event data on a trace attribute in CPM4. . . . 84
B.10 Examples of variables defined in CPM4. . . . 84
B.11 Example of a dimension in CPM4, in order to split the log on type of priority. . . . 85
B.12 Creating a component filter with PQL statements in CPM4 (the names of the customers are disclosed). . . . 85
B.13 Histogram in CPM4 plotting the distribution of the solve time of all cases of type 'incident'. . . . 86
B.14 Function in CPM4 to calculate the median solve time in seconds of a set of traces, depending on value of the priority attribute of the last event in the trace. . . . 86
B.15 Function in RapidMiner to find the cases with a solve time above a certain threshold, with for each attribute value of the 'priority' a different threshold. . . . 89

B.16 Function in RapidMiner to compute percentages of cases above and below the certain thresholds. 89
B.17 *Dotted Chart* created in RapidMiner for **TPM**, visualizing the 'activity' of the resources. 92
B.18 Social Network, Handover of Work, created in RapidMiner with the TPM approach). 93
B.19 Example of function in Celonis to compute percentages of cases above and below the certain thresholds. 94
B.20 Query for selecting data regarding the events for the TPM and CPM approaches. . 98
B.21 Query for selecting data regarding the cases for the TPM and CPM approaches. . 99
B.22 Query for creating a new table, so the PL/SQL script can store the merged events. 99
B.23 PL/SQL script that is used to merge the cases for the TPM and CPM approaches. 100
B.24 Scope for selection data: extraction for the RLPM approach is performed for these three tables. 101
B.25 Function in CPM4 to calculate the median response time in minutes of a set of traces depending on the priority. 102
B.26 Customer 2 and 3 with the Process Comparator. 103
B.27 Customer 1 and 3 with the Process Comparator. 104
B.28 Analysis of the solve times, BPQ 1, for the Case Study with Celonis Process Mining. 105
B.29 Analysis of response times, BPQ 1, for the Case Study with Celonis Process Mining. 106
B.30 Arbitrary set of Visual Analysis, for the Case Study in Celonis Process Mining. . . 107
B.31 Analysis of customers, BPQ 2, for the Case Study with Celonis Process Mining - 1. 108
B.32 Analysis of customers, BPQ 2, for the Case Study with Celonis Process Mining - 2. 109
B.33 Analysis of resources, BPQ 3, for the Case Study with Celonis Process Mining - 1. 110
B.34 Analysis of resources, BPQ 3, for the Case Study with Celonis Process Mining - 2. 111

C.1 Required domain knowledge visualized in ontology graph for stage 2, extraction, with TPM and CPM. 113
C.2 Required domain knowledge visualized in ontology graph for stage 2, extraction, with RLPM. 113
C.3 Required domain knowledge visualized in ontology graph for stage 3, data processing, with the TPM and CPM approach (resources and customers have been anonymized). 114
C.4 Domain knowledge visualized as an ontology graph for stage 3: data processing with the *RLPM approach* (resources and customers are anonymized). 115
C.5 Perceived ease of use for the traditional process mining approach. 118
C.6 Perceived ease of use for the redo log process mining approach. 118
C.7 Perceived ease of use for the Celonis process mining approach. 119
C.8 Perceived usefullness for the traditional process mining approach. 120
C.9 Perceived usefullness for the redo log process mining approach. 120
C.10 Average Perceived Ease of Use for the three PM approaches: a for the TPM approach, b for the RLPM approach and c for the CPM approach. 121
C.11 Average Perceived Usefulness for the three PM approaches: a for the TPM approach, b for the RLPM approach and c for the CPM approach. 121
C.12 Overview and averages of the scores on the perceived ease of use and perceived usefullness. 121

List of Tables

1.1 Software products with process mining functionality, academic and commercial. . . 3

3.1 Stages of PM^2 and their definitions. 14
3.2 Metrics for the actual efficacy of a process mining approach in stage 2: extraction. 17
3.3 Different entities of a log and four problem categories. 21
3.4 Maturity levels for event logs. 22
3.5 Metrics for the actual efficacy of a process mining approach in stage 3: data processing. 22
3.6 Metrics for the actual efficacy of a process mining approach in stage 4: mining & analysis. 23
3.7 Measurement Scales for Perceived Usefulness and Perceived Ease of Use 25

4.1 Characterisics of the event logs created with the TPM and RLPM approaches. . . 28
4.2 The distinct event classes for the event classifiers of the TPM and RLPM approach. 28

6.1 Visualization of the task complexity per stage of PM^2 and process mining approach. 37
6.2 Normalized results for the input measures for all PM approaches per stage. 38
6.3 Performances of PM approaches on efficiency input measures for different scenarios. 39
6.4 Evaluation of event log issues for the event logs of each PM approach. 47
6.5 Assessment of PM approaches across multiple dimensions. Scores are normalized to percentages. 54

7.1 Suggested starting point for categorizing business process questions. 59

B.1 Service Level Agreements for solve and repsonse times for tickets of type incident. 70
B.2 Overview of the systems and software tools required per approach. 72
B.3 Metrics describing the efficiency and effectiveness of Stage 2 of PM^2. 81
B.4 Identified outputs of the third stage, based on the Business Process Questions. . . 82
B.5 Metrics describing the efficiency and effectiveness of Stage 3 of PM^2. 87
B.6 Process conformance for *solve* and *response times*, answering all questions related to BPQ 4. 89
B.7 Process conformance for *solve* and *response times*, answering all questions related to BPQ 4. 90
B.8 Process conformance for solve and response times, regarding BPQ 3. 93
B.9 Results for metrics describing the effectiveness of Stage 4: Mining & Analysis of PM^2. 95

C.1 Overview of the estimations for the input measures for all three approaches for all the stages per stage. 116
C.2 Normalized results for metrics to estimate the actual effectivenesss of PM approaches for stage 2 of PM^2, scores in percentages. 117
C.3 Normalized results for metrics to estimate the actual effectivenesss of PM approaches for stage 3 of PM^2, scores in percentages. 117
C.4 Normalized results for metrics to estimate the actual effectivenesss of PM approaches for stage 4 of PM^2, scores in percentages. 117

Chapter 1

Introduction

This work is devoted to finding the best approach to practice process mining within organizations. To do so, approaches need to be compared to each other in a fair manner. To that end, we propose a comparison framework for process mining approaches with each other. We apply three contrasting approaches in a real-life case study and compare them by the use of the proposed comparison framework. One of the three approaches is relatively new, and has not been applied in a real-life setting before. This chapter introduces the process mining, illustrates the growth of the process mining field, and explains the significance of this study. Also, the goal of this study is explicitly given. Lastly, the scope of the project and a description of the case study are provided.

Process Mining (PM), sometimes also referred to as Automated Process Discovery, or Process Intelligence, is a discipline that intersects with many fields. As is illustrated in fig. 1.1, process mining intersects with data science domains such as: statistics, databases, computational intelligence, data mining and visual analytics.

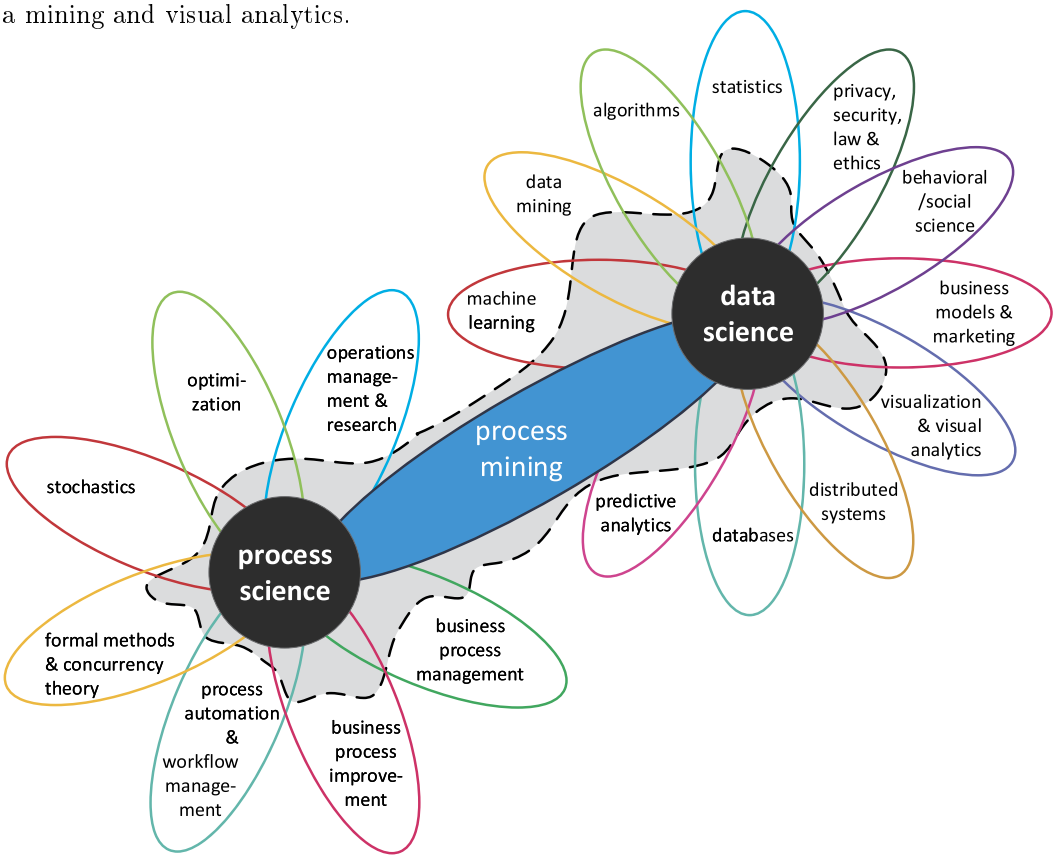


Figure 1.1: Process mining as the bridge between data science & process science, adopted from [1].

At the same time, PM intersects with domains in the field of process science: business process managment, optimization, stochastics, process automation & workflow management, and operations management & research. PM bridges the gap between data and process science [1].

With PM one can discover, monitor, and improve business processes. By exploiting historical data in the form of event logs, derived from information systems, valuable knowledge about real processes is harvested. The three bed stones of PM are: *process discovery* (extracting process models from a log); *conformance checking* (monitoring deviations by comparing model and log); *process enhancement* (extend or improve existing models using a log). PM techniques also entail social network/organizational mining, automated construction of simulation models, model extension, model repair, case prediction, and history-based recommendations. This makes PM an enabling technology for Business Intelligence related management tools, e.g. Corporate Performance Management, Business Process Improvement, Total Quality Management and Six Sigma.

1.1 The Emergence of Process Mining

Although process mining is a relatively new research discipline, it has been an evolving field of expertise. The growing interest for PM is fueled by two trends. These drivers are essentially that more and more data becomes available, and the need for information to improve processes [1].

Firstly, technological advancements enable an increasing growth of the digital universe. A digital universe that is well-aligned with processes in organizations enables *recording and analysis of events*. Events are activities, performed at a moment in time, on or with a particular instance of a process (e.g. a customer paying for an online purchase, a person making an appointment for a check-up on his car, or a doctor adjusting an X-ray machine).

Secondly, there is a growing interest from industry in process mining. The role of process mining becomes increasingly important in Business Process Modeling. Business is in need of more than just a static, marginal snapshot of performance by considering their processes. Process mining helps organizations to "mine their own business". They are empowered to discover, monitor and improve real processes by extracting knowledge from event logs.

At the same time, many new educational studies have been set up in the past few years. Data science, and with data science, also process mining, is the profession of the future. For example, the recent set up of the Jheronimus Academy of Data Science in the Brabant region, Masters and Post-graduate programs in Data Science all over the world and dozens of Massive Open Online Courses on numerous platforms. Addtionally, the establishment of the Process Mining Manifesto strengthens the awareness of process mining [2]. The publication of the manifesto, written by the IEEE Task Force on Process Mining, has the goal to increase the maturity of process mining as a new tool to improve the (re)design, control, and support of operational business processes.

As a consequence, the development of process mining techniques is emergent. Process mining algorithms have been implemented in various academic and commercial systems already. At this point in time more than 500 plugins have been developed for ProM[1] (the scientific cradle for new PM technologies), with also a new version called ProM Lite 1.1[2] recently released specific for end users. More and more software vendors are adding process mining functionality to their tools. For an overview of examples of software products with process mining capabilities, see table 1.1.

1.2 Significance of this Study

As was illustrated in section 1.1, PM is becoming increasingly important—growth is observed both on the supply and demand side. As a result, more and more organizations are launching PM projects. However, the availability of preferably mature event logs is a hard prerequisite to perform any process mining activities. In turn, the extraction of event logs, relies on the existence of event data. Although data is omnipresent, not every process is logged in a systematic manner,

[1] http://www.promtools.org/
[2] http://www.promtools.org/doku.php?id=promlite11

i.e. with explicitly defined case and activity notions in a reliable manner. Often in real-life cases, it is not easy to obtain such event data.

Table 1.1: Software products with process mining functionality, academic and commercial.

Name	**Organization**	**Country**
ProM	The Process Mining Group, TU/e	The Netherlands
ProM Lite	The Process Mining Group, TU/e	The Netherlands
RapidProM	RapidMiner	United States
Disco	Fluxicon	The Netherlands
Celonis Process Mining	Celonis GmbH	Germany
Process Aware Data Suite	Eduardo Gonzalez, TU/e	The Netherlands
Perceptive Process Mining	Perceptive Software, Lexmark International	United States
QPR ProcessAnalyzer	QPR software	Finland
Process Performance Manager	Software AG	Germany
Process Analytics Software	Fujitsu	Japan
StereoLOGIC Discovery Analyst	StereoLOGIC	United States
Comprehend	OpenConnect	United States
minit	GRADIENT	Slovakia
myInvenio	Cognitive Technology	Malta
Rialto	Euxera	Italy
SNP Business Process Analysis	SNP AG	Germany
Desktop and Process Analytics	Verint	United States
XMAnalyzer	XMPro	United States
Process Gold	Process Gold	The Netherlands

Using the redo log of a database could be an 'escape' in situations where event data is non-existent or not logged in an appropriate manner, and/or it is complex to set up an Process Aware Information System to record event data.

Definition 1. *The* ***redo log*** *can be used for recovery operations on a database. In the redo logs, all changes—insertions, updates, and deletions—made to a database are stored as they occur.*

A newly developed approach to create event logs from these redo logs creates a new opportunity in situations where the availability of event data is an issue [3]. The tool for this approach, named PADAS, has—as of now—only been tested with a dummy-database. Hence, it is interesting to apply and verify this new approach in a real-life business environment. Because process mining with redo logs does not require the availability of an information system logging the historical data a priori, mining with redo logs seems a promising solution for organizations that have no (or immature) logging of events.

Additionally, it is interesting to compare the new approach with existing approaches. However, there is no standardized methodology available yet to make an objective comparison between different process mining approaches. Also, there is no standard method in the industry to deploy a process mining project. Meanwhile, more and more organizations acknowledge the value of process mining, but struggle on the *how-to*. This includes questions like, "What data is required?", "How to retrieve this data?", "What tools to choose?", "Which business process questions can be answered with process mining?".

As the interest for PM in the industry grows, the need for clear directions and best practices grows. With this growth of supply and demand for PM techniques, the validation and comparison

of these techniques in real-life environments becomes more relevant. And in the mean time, an establishment of a more standardized process mining methodology is finding its way.

Concluding we can state that there is a need to enable the comparison of PM approaches, to assess them in a fair manner. And the verification of the redo log process mining in a real-life case study is still a job undone. The next section explains how this thesis addresses these tasks.

1.3 Goal of this Study

This study focuses on *comparing three process mining approaches* for executing a *Process Mining Project* in a *real-life* setting, by the means of a *case study* using a Data Base Management System.

The first one, the *traditional process mining* approach, covers a wide variety of tools, mainly used in academic environments. It is the baseline for the comparison; to create a frame of reference.

The second one, is the *redo log process mining* approach. Today's process mining techniques rely on the existence of event data, and often it is far from trivial to obtain suitable event data. Substantial efforts may need to be spent to make IT systems record historic data. Even if such records are available, it may not be possible to derive an event log for the case notion one is interested in, i.e., it may be challenging to correlate events into process instances [3]. Having stated this, a newly developed technique, namely to create event logs through the extraction of the redo log of a Data Base Management System (DBMS), could be a solution. Thus, this study also aims at applying and verifying the new technique in real-life context for the first time.

The last one, uses the off-the-shelf software suite for process mining developed by a company called Celonis. This company intends to deploy process mining in real-life scenarios; i.e. the business. In this work, this is reffered to as the *Celonis process mining* approach. It is interesting to find out if this approach is easier, or faster, because it is sometimes perceived so by the industry. Detailed information about each of the process mining approaches that is compared in this this study, is given in section 2.1 till 2.1.

For a fair comparison, this study proposes a *comparison framework for process mining approaches in real-life process mining projects*. In short, the framework uses quantitative and qualitative metrics to assess the effectiveness and efficiency of the execution of tasks using a particular method. The tasks are defined as the different stages in a process mining project in real-life.

Altogether, the following research question was formulated.

Research Question. *Which process mining approach is best suited for a real-life process mining project?*

We also defined research goals, these are shown below. By achieving these in an orderly fashion, we gradually gain more insights towards answering the main research question. Most likely, our research will not result in an absolute winning PM approach. However, we seek to find patterns, dimensions and context characteristics that are met differently by the process mining approaches. Additionally, we want to uncover fundamental differences and expose these in a clear manner.

RG 1. *Propose a framework for comparing different process mining approaches.*

RG 2. *Apply and verify the Redo Log Process Mining approach in a real-life case study.*

RG 3. *Perform the comparative study using the proposed comparison framework on process mining approaches in a real-life case study.*

RG 4. *Discover and report the findings of the comparative study of research goal 3.*

RG 5. *Evaluate the proposed comparison framework.*

This section is concluded with the research goals that have been identified for this research. The following section defines the scope of the comparative study. The subsequent section describes the case study in more detail.

1.4 Scoping

This section describes the scope and delimitations of the study. These are the weaknesses of the study beyond the control of the researcher.

This work involves a real-life case study, which directly scopes the application of the process mining approaches and the comparative study. Case characteristics entail: the (Process Aware) Information System, the type of database schema model of the case study, the complexity of the process that is analyzed, the size of the set of historical data that is available and the like. Details about the case study are given in the next subsection 1.4.1. This scoping implies that in other case studies, results could differ.

Moreover, this work is subjected to a sample of one; $n = 1$. I.e., the verification of the redo log process mining approach is scoped in the sense that (only) one case study is carried out. However, a successful verification in any real-life case does imply that the method for creating event logs from the redo logs of DBMS actually works in practice. This is a valuable result on its own. For the comparison framework on the other hand, the application of the redo log process mining approach in more case studies would definitely reveal more results and other observations. Especially comparing the same three process mining approaches again, could lead to interesting findings. Nonetheless, this research can show the feasibility of the comparison framework for process mining approaches.

In addition, the research is executed by one person, the author of this work, thus subjected to the domain knowledge and experience of this person. This will influence for example the time that is required to execute a task. Also, the qualitative assessments of the approaches (e.g. based on task complexity) might be affected by the PM skills of the researcher.

This research, by means of one case study will not provide significant results to appoint a *best* process mining approach. It is important to note that this work merely seeks to identify explanatory patterns for phenomena. By finding patterns, dimensions and case study characteristics that are met differently by the process mining approaches, we can generate hypotheses for future research. Also, this work can point out elementary characteristics of different process mining approaches, and show these in clear-cut fashion.

The study is also scoped by the amount and the chosen types of process mining approaches for the comparison. A balanced mix was created by the choice for traditional process mining, redo log process mining, and Celonis process mining. Interchanging for example the Celonis process mining approach with a similar type of approach from the industry, (e.g. Fluxicon or Perceptive Software) could result in different observations.

Additionally, an important *assumption* is the definition that is established on for the traditional process mining approach. This study defines that the traditional approach covers all the tools and techniques used in earlier academic work, which mainly involves ProM and RapidProm. For this case study, only a subset of the many possibilities in ProM and RapidProm is applied. The traditional process mining approach could entail more or other analysis PM techniques that are not included in this research. However, the potential of the traditional approach does become clear in the comparison study.

1.4.1 Description of the Case Study

The application of the redo log process mining approach in real-life and the proposed comparison framework for process mining approaches are both practiced in a real-life case. For the selection of the case study, two specific requirements for the information system of the case study were taken into account. Firstly, the information system should already record historical event data about the process. And secondly, the information system should run on a DBMS. These two requirements enable the application of all the three process mining approaches that are selected for this research.

The organization that the case study is about, is called Experis Ciber[3], a full-service IT-

[3] http://www.ciber.nl/

provider that supports other organizations in digital transformations. Experis Ciber is part of the Manpower Group. The information system that is used in the selected case study is called OTRS Business Solution™ provided by the OTRS Group[4]. The OTRS software, is a Process Aware Information System (PAIS) and it is open source.

The major goal of the incident management process is restoring a customer's normal service operation as quickly as possible when incidents arise, to ensure the highest possible levels of service quality, and maintain the availability. Activities involved in the process are diagnosis of the root cause(s) of incidents and to solve those problems to enhance the quality of IT services delivered by the organization. The ticket elements of the service management system can be considered as cases. Tickets are running reports of a particular problem or issue, its status, type, priority and other relevant data. The primary goal of the OTRS system is to support this process and its activities by providing a means to administrate and communicate about the incidents.

What should be noted is that this research is scoped (and limited by) the OTRS system. The process is not too complex since the team of resources that are responsible for the process consists of no more than ten people. The OTRS system has been live since the end of 2014 to support this system. Not all the functionality of the OTRS system is used in this case study. For example, the queuing functionality (i.e. queues can be used to appoint tickets to different service departments) is not used. Also, the timing accounting module (i.e. to maintain a time card for tickets) of the OTRS system is not used. Besides, the OTRS systems has a feature to define workflow templates for common tasks. However, as far as the researcher was aware this functionality was also not used. Looking at the schema model of the database of the OTRS system[5], we could state that this is a rather small one, especially since a lot of tables are not used. Summarizing, we could state that the PAIS is not used in an optimal and mature manner. However, sufficient process steps are logged to start a process mining project. The case is considered to be not too simple, since at the start of the project, the researcher had no domain knowledge of the process at hand. Moreover, the process is not logged in a very structured or standardized manner. Also, the tickets are sometimes merged in the process, when two tickets concern the same issue: this created an extra challenge in the process mining project. Thus the case is neither too complex, nor too simple.

Thesis Outline

Preliminaries Chapter 2 provides domain specific information, relevant for this research, to assist readers that are not familiar with the process science field. Here, also the existing the state-of-the-art work related to this research is presented.

Methodology The comparison framework for process mining approaches is presented in the methodology in chapter chapter 3, thus addressing the first research goal.

Verification Accordingly, the second research goal is achieved in the chapter 4, to apply and verify the Redo Log Process Mining approach in the case study.

Summary Case Study Analysis Chapter 5 summarized the steps that were executed to actually bring the comparison framework in practice in the case study. As a result, this chapter addresses the third research goal.

Results In chapter 6, the outcomes of the execution of the comparative study are discussed. This chapter reports the findings from the comparison and therefore meets the fourth research goal. Finally, with the results we aim to answer the main research question.

Conclusion The conclusions in chapter 7 starts with a summary of the research and its results to address the aim of work. Accordingly, this chapter evaluates the comparison framework by reflecting on the case study, and thus tackles the last research goal. Ultimately, a suggestions for future work are listed.

[4] https://www.otrs.com/
[5] http://ftp.otrs.org/pub/otrs/doc/database-schema/otrs-3.3-database.png

Chapter 2

Preliminaries

We have defined our research question, research goals, and research scope in the previous chapter. Before going into the details of our comparison framework, we introduce the preliminary concepts within the scope of this thesis. E.g. process mining, the process mining life cycle, event logs, the three process mining approaches. These concepts, which are presented in the next section (2.1), will be used throughout this thesis. The succeeding section (2.2), presents the related work, i.e. the technology acceptance model, Moody's model for method evaluation, and methodologies for process mining projects.

2.1 Background

This section provides background knowledge to support the reader that is not familiar with the field of process mining. The comparative study that is presented and conducted in this thesis is based on the application of existing process mining techniques. In order to understand the implications of a comparison of process mining approaches, it is helpful to gain a basic understanding of the process mining domain. Chapter 1 already introduced and discussed the growth of the process mining domain. This section explains process mining concepts in more detail.

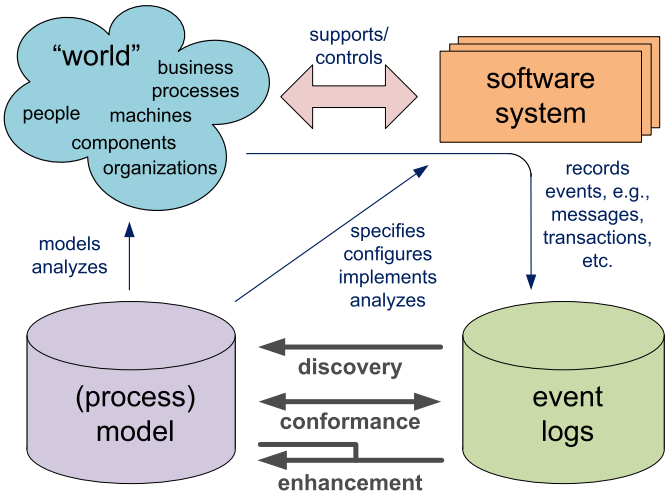


Figure 2.1: Positioning of three main types of process mining: *discovery*, *conformance*, and *enhancement*, adopted from [1].

Fig. 2.1 shows that process mining establishes links between the actual *processes* and their *data* on the one hand and *process models* on the other hand [1]. Processes are an integral part of today's world, driving services and internal functionalities in businesses, governmental bodies, and organizations everywhere. As was explained in the previous chapter, today's information systems log enormous amounts of *events* that occur within processes.

Logging is for example done in work flow management systems, Business Process Management systems, ERP systems, middleware, hospital information systems. This set of systems can be collected under the umbrella of Process Aware Information Systems (PAISs). PAISs include all software systems that support processes and not just isolated activities [4]. The data about the activities that are executed in the process that is produced by a PAIS, is referred to as *event logs* in fig. 2.1. However, information systems often store this information in an unstructured form. For example, event data that is scattered over many tables, or event data that resides in subsystems exchanging messages. In these situations, the event data exists but efforts are required to extract it. Therefore, data extraction is an integral part of any process mining effort [1].

For clarity reasons, a distinction needs to be made between event data and event logs. In short, event data records information of some timed event occurrences in a particular domain [5]. Event data is defined as a collection of data related to a selected process, without predefined case notion or event classes [6]. A *case notion* relates events such that together they form a process instance, and *event classes* identify activities within a process instance. Event logs are defined as specific views on event data, created by defining case notions and event classes. Event logs consist of events that are clearly defined, referring to precisely one case and one activity [7]. When creating event logs, it is possible to create different views, depending on the desired research goal of the analysis.

A direct consequence of logging information about activities of a particular process, is that the event data needs to be stored somehow. A commonly known way to store data is on a database. A database is essentially an organized collection of data. A Database management system (DBMS) is system software for creating and managing databases. In general, the DBMS controls the storage, organization, and retrieval of data. Among others, a DBMS provides users with a systematic way to retrieve (event) data.

Let us consider a situation in which we can sequentially record events, such that each *event* refers to an *activity*, (i.e. a well-defined step in the process). Additionally, each *activity* is related to a particular *case* (i.e., a process instance). Using the process described in section 1.4.1 as an example, the cases would be the tickets. An activity could be 'New ticket received'. Per case, we can record a *trace*, which is a set of events. Usually, we refer to these events with the activity names. Preferably, event logs store additional information about the events. With more (relevant) information, a more complete analysis can be performed. E.g. some process mining techniques use extra information such as the *resource* (i.e., person or device) executing or initiating the activity, the *timestamp* of the event, or *data elements* recorded with the event (e.g., the customer that issued the ticket). See fig. 2.2 for a depiction of the characteristics of events and cases and their relationship. Next, an event log can be used to conduct three types of process mining: discovery, conformance, and enhancement. Fig. 2.1 also presents the positioning of these three main types of process mining.

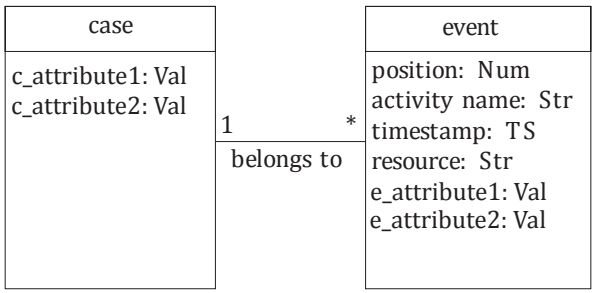


Figure 2.2: Cases and events can have attributes and each event can be associated to a single case, adopted from [8].

Discovery There is no a priori process schema available. Thus, based on an event log, some schema is constructed. E.g., using a process mining algorithm to discover a Petri Net model.

Conformance A process schema is a priori available. The schema (e.g. process model) is used to check if reality conforms to this schema. An example is an existing process model of some engineering process which indicates that all products should have a quality check. Conformance checking can also be used to detect deviations, locate, and explain these deviations, and measure the severity of these deviations [9].

Enhancement A process schema is a priori available. This schema is extended with a new aspect or perspective. Thus, the idea is to enrich the schema. E.g. the extension of a process model with performance data to spot the bottleneck(s).

Starting a process mining project, in real-life (e.g. in a business environment), enables an organization to improve the performance of a selected process or an organization's compliance to rules and regulations. As has been explained in the previous chapter, many tools are available to practice process mining. For such a process mining project, an organization will apply a certain process mining approach. An approach is defined as a way of considering or doing something. In the context of this research, an approach will in that sense refer to a way of executing a

task/performing an activity. We define a process mining approach as a distinguishable way to perform a process mining project, e.g. using a specific set of tools and techniques. As stated before, this work intends to compare three approaches to address a real-life PM project. This section elaborates on each of these three PM approaches: the traditional process mining approach, the redo log process mining approach, and the Celonis process mining approach. The remainder of this section explains each of these process mining approaches.

Traditional Process Mining Approach (TPM) In literature, the words "traditional process mining approaches" are sometimes used to address existing process mining techniques (e.g. algorithms) that have problems dealing with unstructured processes and thus often resulting in "spaghetti-like" models [10]. However, in this research *the traditional process mining approach* (the TPM approach), refers to the complete package of PM techniques that are applied throughout all the stages of a process mining project.

Figure 2.3: Logo ProM.

We advocate that theoretically, the TPM approach could cover every tool and technique published in academic environments for any activity in a process mining project. A substantial part of the foundation for traditional process mining approach, is written in the first book on process mining; *"Process mining: discovery, conformance and enhancement of business processes"*, by van der Aalst, W. (2011) [11]. Furtermore, parts from his most recent publication *"Process Mining: Data Science in Action"* have also been applied [1]. The lion's share of the existing literature on PM uses the ProM Tools[1] (fig. 2.3) for implementing process mining techniques [12].

Moreover, a lot of existing work has been devoted to defining a standard logging format. This has led to the standard *XES format* [13], which is extensible and supported by the OpenXES library[2] and by tools such as ProM, XESame, Nitro, etc. The meta model of the XES format is shown in fig. 2.4. An event log has traces, and every trace contains an arbitrary number of event objects. All information in an event log is stored in attributes. For a more detailed explanation, we refer to the complete standard definition[3]. The older MXML event log format, used to be the defacto standard format for storing event logs, before XES emerged. As this XES is the new standard for storing event logs, and well-supported, this thesis uses the XES event log.

For this research, a subset of these practices has been chosen by the researcher. This mostly entails the use of ProM, RapidMiner with the RapidProM extension and custom scripts (Python, Groovy, and PL/SQL). The reasoning behind these choices is based on the earlier academic work [12] and existing practical process mining experience of the researcher.

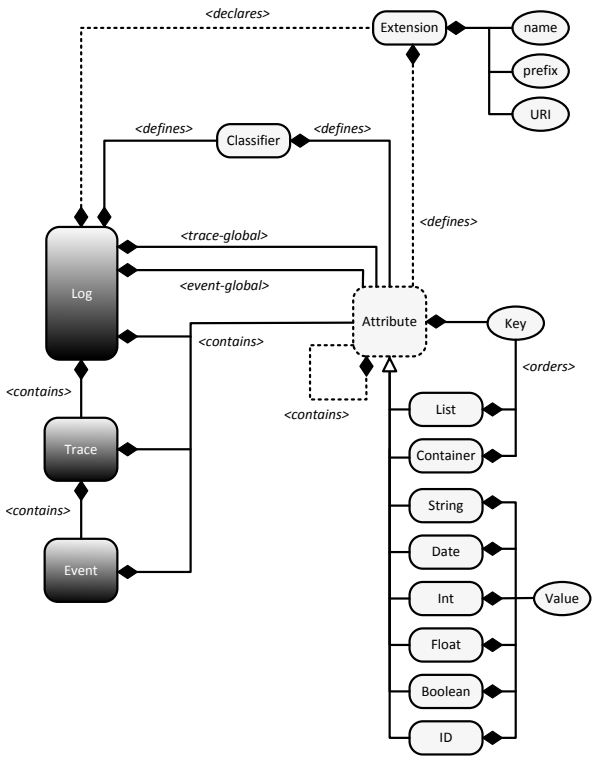


Figure 2.4: XES meta model with UML 2.- class diagram, adopted from [13].

[1] ProM: http://www.promtools.org
[2] The OpenXES library: www.openxes.org
[3] XES Standard Definition: http://www.xes-standard.org/xesstandarddefinition

Redo Log Process Mining Approach (RLPM) In this approach, a database view on event data is applied. The key idea is that events leave footprints by changing the underlying database, via the creation, deletion or modification of objects and relations [7]. Database technology often provides so called "redo logs", which are maintained for data integrity and recovery reasons by Database Management Systems (DBMSs) like Oracle RDBMS and MySQL [3]. In these redo logs, every operation (e.g. insertions, deletions and updates) on the database (DB) is logged. By correlating all these events to traces that one is interested in, it is possible to create an actual event log. These event logs can accordingly be used to perform mining and analysis. To implement this *redo log process mining approach*, the Process Aware Data Suite[4] (PADAS; fig. 2.5) is employed.

Figure 2.5: Logo PADAS.

Celonis Process Mining Approach (CPM) Celonis[5] is a company that offers a tool to perform activities like filtering, slicing and dicing, drilling down/up, discovering process models, creating dashboards and reporting. This is a commercial product, however, for research purposes, Celonis offers free academic licenses. Celonis focuses more on the business users, e.g. workers of commercial companies. As of 2015, Celonis is integrated in SAP software as SAP Process Mining, to distill real time data about business process execution into graphical representations, mostly with the purpose of visualization. The *Celonis process mining approach* tries to limit the use of tools to this software suite of Celonis (CPM4; fig. 2.6).

Figure 2.6: Logo CPM4.

This concludes the background section. The next section focuses on other related work that falls withing the scope of this research.

2.2 Related Work

The this section discusses all the relevant literature that is related to the comparative study and the research goals that have been presented in chapter 1. Section 2.2.1 discusses success in the form of adoption in practice, and explains how we can measure 'pragmatic success'.

2.2.1 Adoption in Practice & Pragmatic Success

For any information system in general, the user acceptance is a determining factor for the success of the information system. Irrespective of the technical superiority or potential advantages of any information system, if it is not *adopted in practice*, the benefits cannot be realized [14]. In this light, the Technology Acceptance Model (TAM) has been the most influential model for user technology acceptance, originated by Davis (1986) [15]. Substantial empirical evidence for TAM has been presented [16, 14, 17, 18, 19, 20, 21, 22]. Already, more elaborate models, TAM2 [23], TAM3 [24] have been proposed. These extended models mostly added additional constructs as determinants for TAM. The basis of knowledge accumulated over the years from TAM research remained the same, and is given in fig. 2.7.

TAM explains that an individual's behavioral intention to use a particular system is determined by two beliefs. The perceived ease of use (PEoU), defined as the degree to which a person believes that using an IS will be free of effort. And, the perceived usefulness (PU), defined as the extent to which a person believes that using a system will enhance his or her job performance.

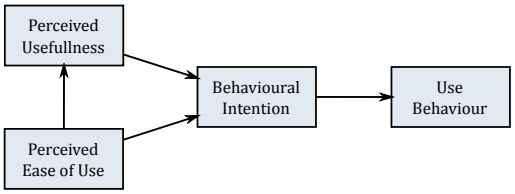


Figure 2.7: Technology Acceptance Model.

[4]PADAS: Process Aware Data Suite https://www.win.tue.nl/ egonzale/projects/padas/
[5]Celonis Process Mining: http://www.celonis.com/en/

It can be argued that TAM is also applicable on process mining approaches, since there are clear parallels between user acceptance of information systems and practitioners' adoption of methods [16]. Since these are all subject to individual choice (e.g. a researcher can make a decision on which process mining approach he/she applies). Any chosen process mining approach is therefore the result of reasoned action. Moody (2003) argued that theoretical models used to explain and predict user acceptance of methods [16].

With that in mind, Moody (2003) defined a generally applicable theoretical model for comparing and validating IS design methods. This model is based on two areas of theory: methodological pragmatism and the TAM [16]. TAM was explained previously. The remainder of next section elaborates on the concept of pragmatic success.

According to Moody (2003), the validity of a method can be established by applicative success in practice [16]. Only demonstrating that a method is correct, is not sufficient for validation. A validated method is a rational practice to adopt a method based on *pragmatic success*. The definition of pragmatic success is given as "the efficiency and effectiveness with which a method achieves its objective" [16]. Figure 2.8 illustrates that all methods are designed to improve performance of a task, and this can be achieved in two ways. Firstly, by improving the efficiency, i.e. reducing the effort required to complete a task. Secondly, by improving the effectiveness, i.e. improving the quality of the result. Actual efficiency and actual effectiveness are based on the concept of pragmatic success. Efficiency of a method is defined by the effort required to apply the method, and can be measured by input measures such as time, cost or effort. Effectiveness of a method is defined by how well it achieves its objectives, and can be measured by evaluating the quantity and/or quality of the results (output measures). Methodological pragmatism refers to the pragmatic value of a method.

So, by measuring the *actual* efficiency & effectiveness, and the *perceived* efficiency & effectiveness — the objective and subjective performance of a PM approach can be determined. Accordingly, with measurements of multiple approaches, performing the same task, one should theoretically be able to appoint the best performing method for a particular task. Moody's theoretical model for evaluating models, combining TAM and methodological pragmatism, is operationalized in chapter 3.

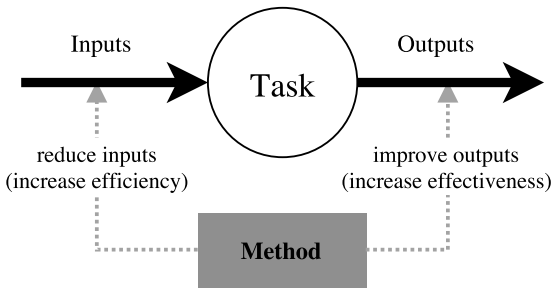


Figure 2.8: Efficiency vs. effectiveness, from [16].

In order to apply Moody's method evaluation model on a complete process mining project, it is necessary to identify tasks within such a project. Therefore, a methodology for a process mining project on it's own, is needed. The next section explores the state-of-the-art methodologies for PM projects found in literature.

2.2.2 Methodologies for a Process Mining Project

For a process mining project, an organization will most likely benefit of a methodology to guide the execution of process mining project. One can use the following metaphorical comparison; a methodology for a process mining project would be similar to a route description to for example the nearest library, and the choice of transport (e.g. car, bike, walk, or a combination) resembles the chosen process mining approach. This section discusses the earlier work from literature that is related to the aforementioned.

Next to the general methodologies to gain insights from datasets —like KDD [25], SEMMA [26] developed by a consortium led by SPSS, and CRIPS-DM [27]— several methodologies more tailored to process mining techniques are proposed in literature.

The life-cycle model ($\mathcal{L}^*$) [1, 11] and Process Diagnostics Methods (PDM) [28] are defined as two well-known process mining methodologies. $\mathcal{L}^*$ is a methodology to describe a process mining project. It is a more holistic view, that portrays the basic steps to improve a structured (Lasagna) process. $\mathcal{L}^*$ could, for instance, be used as a starting point for configuring a WFM of BPM system.

PDM, is designed to perform process diagnostics, based on process mining. Given an event log of an information system within an organization, process diagnostics gives a broad overview of the processes of an organization within a short period of time [28].

Recently two other methodologies, specifically based on the PM analysis, have been proposed. One by Delias, Doumpos & Matsatsinis (2014), to guide the implementation of process analytics projects [29]. This work claims to have synthesized existing works and to have extracted common methodological steps, in order to accelerate project delivery. This methodology contains four stages; Business Understanding (1), Data Collection & Reviewing (2), Discovery(3) and Decision Aid (4). The other methodology, Process Mining Project Methodology (PM^2) is presented by van Eck, Lu, Leemans & van der Aalst (2015). It is tailored for PM in practice aims to guide the execution of process mining projects [6]. PM^2 has been designed to support projects aiming to improve performance or compliance to rules and regulations. A visual overview of this methodology is given in fig. 2.9. PM^2 has six stages: Planning (1), Extraction (2), Data Processing (3), Mining & Analysis (4), Evaluation (5), and Process Improvement & Support (6).

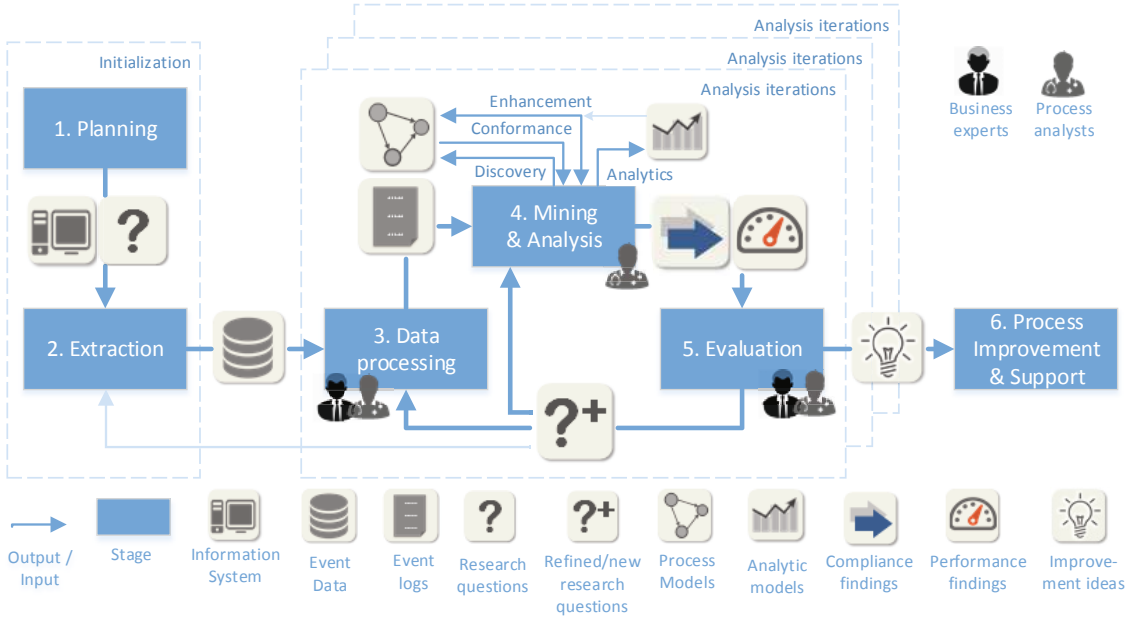


Figure 2.9: An overview of the PM^2 methodology, adopted from [6].

We observe that PM^2 is more detailed than the methodology proposed by Delias, Doumpos & Matsatsinis [29]. Firstly, it covers a wider range of process mining techniques (e.g. conformance and enhancement). Secondly, PM^2 provides more specific guidance through the PM process and the iteration steps. Whereas the methodology proposed by Delias, Doumpos & Matsatsinis [29] presents a simpler sequential process of four stages with an iteration back to the previous stage at the end of every stage. PM^2 is therefore considered to be the most complete methodology. Chapter 3 continues to explain PM^2 in more depth. Also, PM^2, has been evaluated in a real-life case study at IBM. Here, PM^2 led to valuable insights and concrete process improvement ideas. However, it has been stated that there is still need for more detailed and practical guidance to help process mining practitioners tackle a series of challenges [6].

As of now, we have not identified any other existing methodology in earlier work to compare different process mining approaches dedicated to the holistic execution of a complete process mining project. However, one of the studies that PM^2 is based on —proposing the PMPM (Process Mining Project Methodology)[30]— suggests to use Moody's model for method evaluation. In order to perform our comparative study, aiming to find the 'best' process mining approach, we define a comparison framework in this thesis. The next chapter provides a detailed description of this comparison framework for process mining approaches in real-life environments.

Chapter 3

Methodology

To fairly compare any process mining approaches, a comparison framework had to be defined. Therefore, a research goal was defined already in section 1.3, which is shown below.

RG 1. *Propose a framework for comparing different process mining approaches.*

In section 2.2 we explored the related work on comparing methods, and existing work to assess the quality of event logs. Moreover, in section 2.2.2 we identified the different existing methodologies for carrying out process mining projects. Based on this, we propose a comparison framework to compare process mining approaches and eventually to find the 'best' one.

This section explains how the comparison framework works. First, section 3.1 presents the fundamentals of the comparison framework. Next, in section 3.2 the details for the application of the framework in the second stage of PM^2 are given. Subsequently, section 3.3 and 3.4 elaborate in more depth on the different dimensions that were chosen to compare process mining approaches in the third and the fourth stage of PM^2.

3.1 Comparison Framework for Process Mining Approaches

The comparison framework proposed in this work consists of three building blocks as illustrated in fig. 3.1. The blocks represent concepts discussed in earlier work that has been presented in chapter 2. The left block is the Moody's model to evaluate methods, and the right block is PM^2. The top block represents the performance metrics that operationalize Moody's model with for each stage of PM^2. The next three paragraphs explain each block.

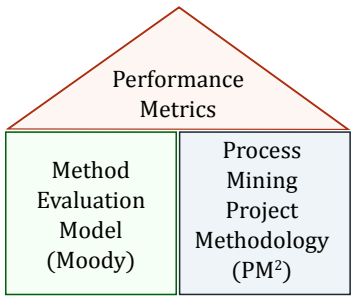


Figure 3.1: The three basic concepts of the comparison framework.

Moody's Method Evaluation Model In section 2.2.1 we introduced the work of Moody, presenting the *Method Evaluation Model* [16]. This model (see fig. 2.8) is the first cornerstone for the comparison framework. We apply the method evaluation model to compare the three process mining methods (i.e. PM approaches). Moody defined two "dependent variables" to determine whether an approach is successful. Firstly, the *actual efficacy* of the execution of a task with a particular PM approach. Moody used Reshers's notion of pragmatic success to estimate this. Secondly, the *adoption in practice*, which is the extent to which the method is used in practice (see fig. 3.2). Regardless of the superiority of a PM approach, if a process mining approach is not used in practice, its benefits cannot be reaped. The other way around, when a process mining approach is used but it reduces the performance of process mining tasks, this will have an adverse effect on the PM project. Thus both aspects of method success, i.e. of a process mining approach, are incorporated in the comparison framework.

Process Mining Project Methodology We combine Moody's model with the *Process Mining Project Methodolgy*, PM^2, which is the second cornerstone for the comparison framework. PM^2 provides directions for organizations performing PM projects aimed at improving process performance or compliance to rules and regulations [6]. PM^2 functions as a guideline to execute the three approaches in a standardized manner. The different stages of PM^2 define the tasks that are performed. A description of the stages, objectives, inputs, and outputs of each stage, are given in table 3.1. The stages consist of levels of progressively smaller subtasks (i.e. activities). However, the comparison framework compares on the stage-level, since the activities are very specific for a PM approach and a comparison of the specific activities is considered to be too low-level. With these two methods, PM^2 and Moody's model for evaluation methods, the foundations for the comparison framework are set.

Table 3.1: Stages of PM^2 and their definitions.

Stage	Definition
1. Planning	Activities: composing project team, selecting business processes and identifying business process questions*. **Objective:** To set up the project and to determine the business process questions derived from project goals. **Input:** Business Processes. **Output:** Business process questions and the set of information systems supporting the execution of the business processes.
2. Extraction	Activities: determining the scope, extracting event data, and transferring process knowledge. **Objective:** To extract event data and, optionally, process models. **Input:** Business process questions and the set of information systems supporting the execution of the business processes. **Output:** Event data and possibly process models.
3. Data Processing	Activities: creating views, aggregating events, and enriching and filtering logs. **Objective:** To create event logs as different views of the obtained event data and process event logs to optimize them for the next stage, mining and analysis. **Input**: Event data and possibly process models. **Output:** Event Logs.
4. Mining & Analysis	Activities: process discovery, conformance checking, enhancement and process analytics. **Objective:** To answer business process questions and gain insight into performance and compliance of the process. **Input:** Event logs. **Output:** Findings (and process models) that answer the business process questions related to performance and compliance goals.
5. Evaluation	Activities: diagnosis, and verification and validation. **Objective:** To relate findings to improvement ideas to achieve project goals. **Input:** Process models, performance and compliance findings. **Output:** Improvement ideas or new business process questions.
6. Process Improvement & Support	Activities: implementing improvements and supporting operations. **Objective:** To use insights to modify the actual process execution. **Input:** Improvement ideas. **Output:** Process modifications.

*In PM^2 these are referred to as research questions. However, this study refers to these questions with the term *business process questions*, to make a clear distinction between the research questions of this particular research and the questions about the process of the case study.

Performance Metrics In the previous sections, the two pillars of the comparison framework were established. The next step is to operationalize Moody's model for method evaluation. Therefore, we defined measurements to assess the objective and subjective efficacy of a PM approach.

Two concepts that determine the *adoption in practice* are the *Perceived Ease of Use*(PeoU) and *Perceived Usefulness*(PU) of a process mining approach. The PeoU is the degree to which one believes that using a particular process mining approach would be free of effort. The PU is the degree to which one believes that a particular process mining approach will be effective in achieving its intended objectives. These two subjective concepts together determine the *Intention To Use* a process mining approach. Which, then determines the actual behavior of using the PM approach. The aforementioned concepts are presented in a conceptual model in see fig. 3.2.

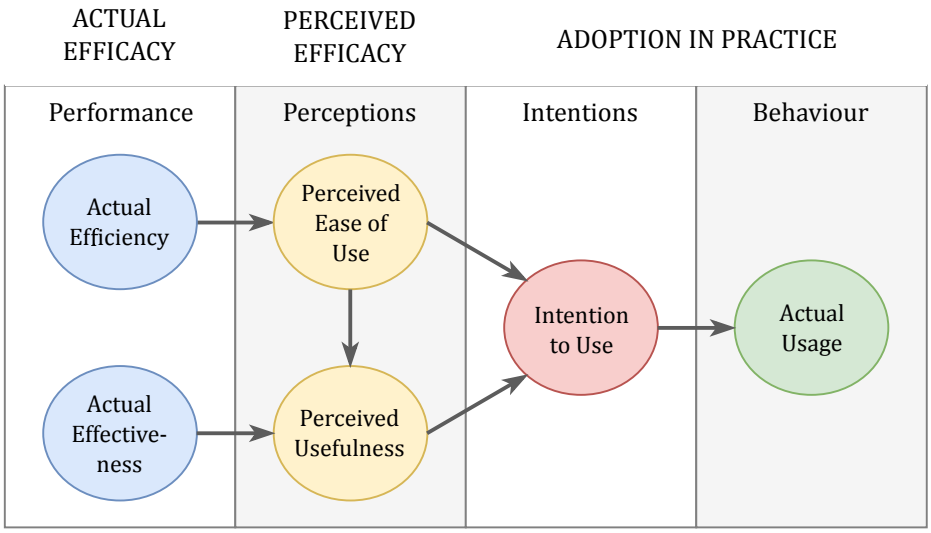


Figure 3.2: Method Evaluation Model adopted from [16].

The *actual efficacy* of a PM approach is measured with its *Actual Efficiency* and *Actual Effectiveness*, based on Rescher's concept of pragmatic success. The efficiency of a PM approach is the effort required to apply the PM approach. For our comparison framework, we propose the use of four different input measures: time, iterativeness, required domain knowledge, and task complexity. As in Moody's model for method evaluation, these are called *input measures*. The effectiveness of a PM approach is defined by how well it accomplishes its objectives. The effectiveness is measured by assessing the quantity and quality of the results. Aligned with Moody's model for method evaluation, these are called *output measures*. Each stage in PM^2 has a different output (again, check table 3.1). For each stage, different output measures have been defined, tailored to the outputs of a particular stage. The output measures are explained in more depth in section 3.2 for stage 2, section 3.3 for stage 3, and section 3.4 for stage 4. This set of performance metrics—the input measures, the output measures, the PeoU, and the PU—complete the comparison framework for PM approaches.

The three building blocks —Moody's model for method evaluation, PM^2, and the performance metrics— together, shape the comparison framework that we propose in this work. The comparison framework is also evaluated in a case study (see chapter 5 and 6). The comparison focuses on the execution of *stage 2: Extraction*, *stage 3: Data Processing*, and the comparison will focus in a minor degree on *stage 4: Mining & Analysis*. The other stages are considered to be performed in a similar manner regardless of the PM approach that is applied. For the second and third stage, the three process mining approaches are different and can thus be compared. In the fourth stage, we only compare the TPM and CPM approach. Remember, after the event logs are created with the RLPM approach, the same mining and analysis techniques can be applied as in TPM. For the sake of illustration, these stages are highlighted in fig. 3.3. This figure shows which process mining approaches are compared in which stages of PM^2.

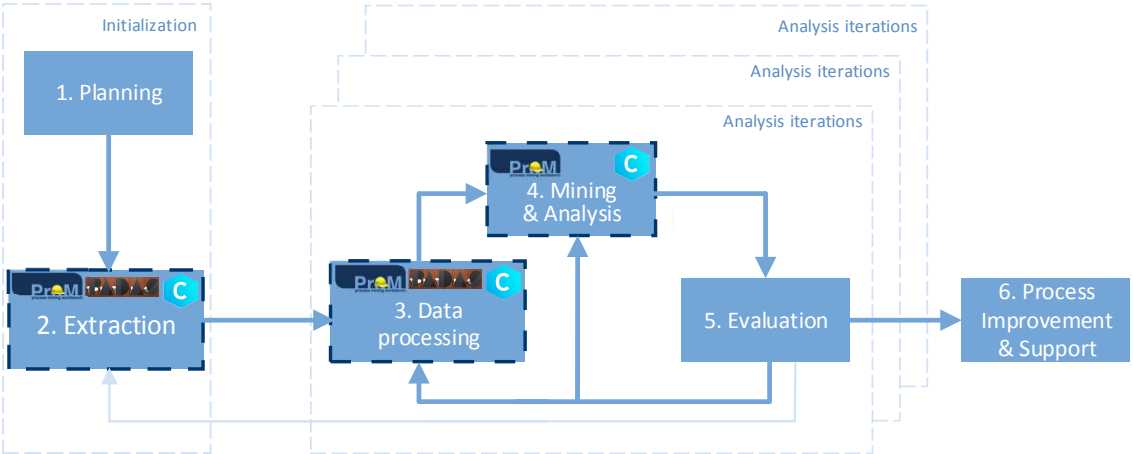


Figure 3.3: Simplified illustration of PM^2, stages for comparison are highlighted and annotated with the logos of the PM approaches applied to that stage.

Thus, each of these three stages highlighted in fig. 3.3 is executed with a set of process mining approaches. Each of these performances is assessed for its actual efficacy using the input measures and output measures. Moreover, each process mining approach, as a whole is assessed for its perceived ease of use and perceived usefulness. Together, these dimensions indicate whether the method improves the performance of the stage. Accordingly, comparing the results for the performance metrics, allows us to determine which process mining approach is best suited in real-life scenarios for optimal results. To support the reader in apprehending the contents of this section, an additional visualization of the comparison framework has been created, see fig. 3.4.

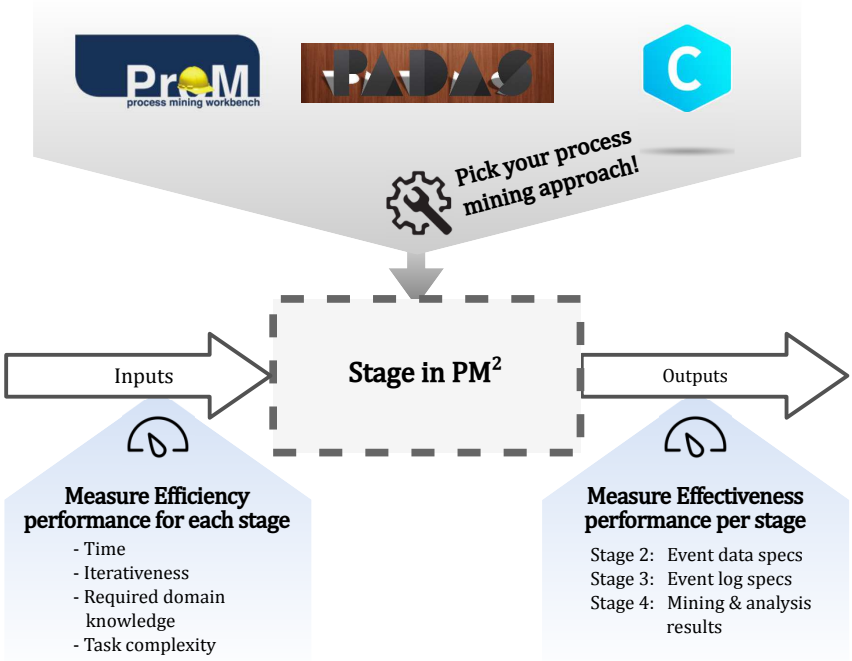


Figure 3.4: Operationalizing Moody's model for method evaluation with PM^2 to compare PM approaches.

The next sections describe the metrics that we defined as input and output measures to determine the efficiency and effectiveness of the execution of tasks. Again, these tasks are stage 2: Extraction, stage 3: Data Processing, and stage 4: Mining & Analysis. They are issued respectively in section 3.2 till section 3.4. Next, section 3.5 explains how the perceived ease of use and perceived usefulness of a PM approach are determined.

3.2 Measure the Actual Efficacy for Stage 2 of PM²

For an overview of the comparative framework, applied to stage 2, see fig. 3.5. We have specified metrics for measuring the effort that is made to execute a stage of PM² by looking at the time it takes to perform it, the number of iterations, the level of domain knowledge that is minimally required, and the task complexity. For an overview, see the upper half of table 3.2. Following the PM² Methodology, the output of the second stage is the event data. The actual effectiveness of the extraction stage are measured by the evaluating quantity and quality of the output, i.e. the event data. The bottom part in table 3.2 shows the metrics that are used to assess the quality and quantity of event data, that is measuring the actual effectiveness.

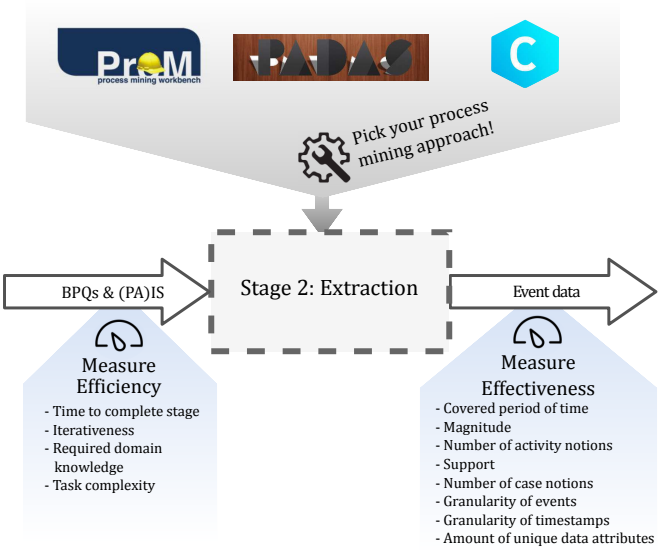


Figure 3.5: Comparison framework applied to the second stage of PM², for the extraction.

Table 3.2: Metrics for the actual efficacy of a process mining approach in stage 2: extraction.

Metrics to determine actual efficiency (input measures)

Index	Label	Brief description
2.1	TIME	Time required to complete stage
2.2	IT-S	Iterations within current stage
2.3	IT-OS	Iterations to previous stages
2.4	DK-O	Number of objects in required domain knowledge
2.5	DK-R	Number of relations in required domain knowledge
2.6	COMPL	The task complexity

Metrics to determine actual effectiveness (output measures)

Index	Label	Brief description
2.7	TIMECOV	Covered period of time
2.8	M	Magnitude (# events)
2.9	A-NOT	Number of activity notions
2.10	SP	Support (# of cases)
2.11	C-NOT	Number of case notions
2.12	$V(L)$	Granularity of events
2.13	GRAN-T	Granularity of timestamps
2.14	ATTR	Number of unique data attributes

Firstly, the *time* required to complete a task (or stage) measures the actual efficiency. Secondly, since PM² allows for iterations, we measure the *number of iterations* within a stage and the number of iterations to a different prior stage. The iterativeness provides a measure that captures the effort that is required to complete a task, and thus an input measure. Thirdly, the *required domain knowledge* to complete a task is an input measure. That is, the amount of knowledge that the performer of the task needs to acquire (or have) before the task can be completed. To make this concrete we propose to create ontology graphs of the required domain knowledge and then count the objects and relations that are present in these graphs. Fourthly, the complexity of a task is the final input measure. It determines the information needed and the cognitive effort to complete a stage of PM². We adopt the task categorization model from literature [31], [32], [33]. The task complexity is determined on the a priori determinability of five dimensions of a

task, namely: domain knowledge[1], process mining information[2], problem-solving information[3], the process itself, and the outcome of the process [34]. Note that the iterativeness and required domain knowledge are also influencing the task complexity. The remainder of this section discusses the output measures that are used to determine the actual effectiveness of a PM approach in the second stage of PM^2.

Covered period of time The period of time that is covered by a set of event data indicates the quantity and determines the quality of the data set. When the covered period of time is too short, not enough observations of the process might have been recorded to perform a reliable analysis. This performance metric is measured in days.

Magnitude This is the number of events in the event data, which also determines the quality of the event data as the quantity again, just like the previous output measure. Event logs with large magnitudes can present a difficulty to the infrastructure of the analysis environment (concerning both storage efficiency and access speed). A vast magnitude is not a problem for mining algorithms since most algorithms scale linearly with log magnitude.

Number of activity notions This is the number of attributes in the event data that can be used as a classifier for the activity label. The amount of activity notions that is available determines the quality of the event data—more options for choosing an activity notion allow for analysis from different points of view. At least one classifier for the activity notion should be available.

Support This is the number of independent observations, the number of observed process instances. Usually, a higher amount of cases in the event data set is better. The goal is to have sufficient independent observations to capture all the behavior present in the process at hand.

Number of case notions This the number of case notions that can be used as a classifier for the case notion present in the event data. This is determining the quality of the event data, just like the number of activity notions. More options allow for more freedom in perspectives for analysis. Again, at least one attribute should be available to be used for the case notion.

Granularity of events The granularity of events, also known as the variety of the event data is the number of unique activities in the event data. This performance metric is evaluated by counting the number of distinct event classes present in the data. A large number of distinct activities can cause process mining techniques to have difficulties since most algorithms scale exponentially with the variety. Then, discovered process models might be spaghetti-like and can be hard to comprehend. One major challenge in process mining is to bridge the gap between the higher level conceptual view of the process and the low-level event logs. As of now, no complete and single solution has been found yet to deal with very low-level events [8]. The granularity of events is, therefore, an important metric for the quality of the event data.

Granularity of timestamps The granularity of the timestamps is defined, in milliseconds, seconds, minutes, hours, days, months, or years, this can be detected by looking at the format of the timestamp attribute. The finer grained the timestamps, the more precise the logging, the better. The granularity should not be too coarse, for example in a situation of only day-wise precision and multiple events occurring on the same day, by which the order of the events becomes unclear. Even though sometimes an ordering attribute is available to sort all the events in the order of occurrence, information about the timing of the events is still missing.

[1] In literature [32], this is called problem information (i.e., the problem characteristics). In this research we call this domain knowledge (e.g., knowledge about the process at hand).

[2] In earlier work [32], this is referred to as domain information. Which is described as known facts, concepts, laws, and theories in the domain of the problem. This is typically tested scientific and technological information published in journal articles and textbooks. Which in this research context, is the knowledge and information available about the process mining field.

[3] We apply the definition of earlier work [33], problem-solving information covers the methods of problem treatment (i.e., what problem and domain information should be used and how).

Total number of unique data attributes This is a dimension to determine the quantity and quality of event data. More attributes usually result in more information about the process. However, too much information can be overwhelming, superfluous, and redundant, which is not beneficial for the researcher. The attributes that are more or less obligatory for event data are at least one case identifier, one activity identifier, and a timestamp attribute. Other attributes that are often contained in event data sets are life cycle information (e.g. start, complete), event importance (e.g. a function mapping each event class to a relative importance), additional case attributes (e.g. the department of the organization that issued the case), and additional event attributes (e.g. the resource that performed a task).

This section has listed all the metrics to determine the actual efficacy of a process mining approach in stage 2. More precisely, we identified a set of input measures to assess the actual efficiency and a set of output measures to qualify the actual effectiveness. The next section explains the performance metrics that have been identified to measure the actual efficiency and actual effectiveness of stage 3 in PM^2.

3.3 Measure the Actual Efficacy in Stage 3 of PM^2

To determine the actual efficiency of a process mining approach in the third stage, we reapply the same input measures of the previous stage. The only difference here is that, for the domain knowledge, we only count the "newly" required domain information. To determine the quantity and quality of the output of the third stage, the event log(s), we apply three concepts: structural log metrics, event log quality issues, and the maturity of an event log. For the visual representation, see fig. 3.6. Also, an overview of all the performance metrics is given in table 3.5. The remainder of this section discusses these in more detail.

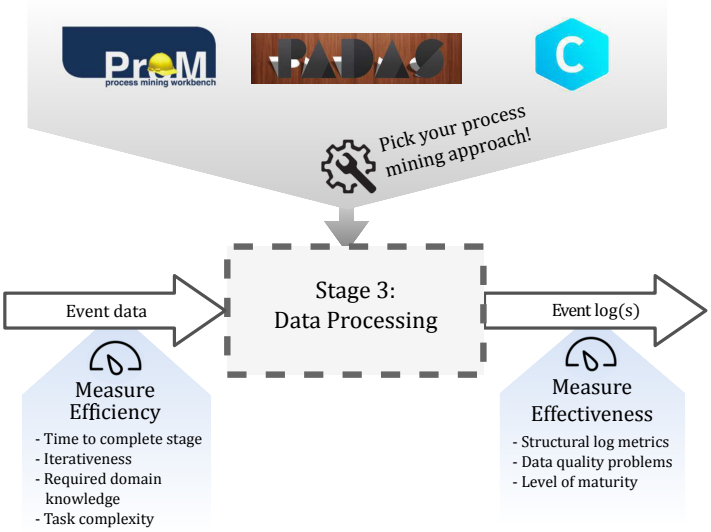


Figure 3.6: Comparison framework for stage 3 of PM^2.

Firstly, a set of *structural log metrics* was identified in earlier work to characterize and measure an event log from a structural perspective [35]. These metrics capture intuitive structural characteristics of an event log in an objective way. They can be used to describe an event log in an abstract manner. The comparison framework uses the structural log metrics to assess and compare event logs from a purely structural perspective. Additionally, these metrics can be beneficial to select suitable process mining algorithms for analyzing these logs. Alternatively, to estimate computational complexity (i.e., computation time) for applying a particular algorithm to a specific log. Secondly, several criteria have been identified to find the *quality of event logs* in earlier work [8]. Namely, by identifying the problems that affect the data quality of an event log [8]. Moreover, the *level of maturity* of an event log can be assessed [2]. The remainder of this section highlights how every metric is applied to assess the quantity and quality of an event log, to determine the actual effectiveness of PM approaches in the third stage of PM^2.

Magnitude & Support The magnitude and support are already discussed in section (3.2), they are also used to characterize an event log (L). The magnitude of an event log ($M(L)$) is the sum of all the events of all the traces, where l denotes a trace, see Eq. 3.1. The support of an event

log ($SP(L)$) is the absolute number of traces in the event, see Eq. 3.2.

$$M(L) = \sum_{l \in L} len(l) \tag{3.1}$$

$$SP(L) = |L| \tag{3.2}$$

Average number of events per case The average number of events per case of an event log ($AE(L)$) aims to identify the average size (or length) of a trace (see Eq. 3.3). When compared to the level of detail, the AE can be used to interpret the amount of repetition of the same type of events (i.e. activities) in traces.

$$AE(L) = \frac{\sum_{l \in L} len(l)}{|L|} \tag{3.3}$$

Level of Detail The level of detail of an event log (LoD) is the mean number of event classes per trace of an event log, see Eq. 3.4. The LoD can be applied to estimate the amount of repetition in an event log. Disproportionate repetition of (episodes of) events usually requires extra attention in the analysis stage; extra filtering can be required to remove distracting behavior.

$$LoD(L) = \frac{\sum_{l \in L} V(l)}{|L|} \tag{3.4}$$

Time Granularity The time granularity of a trace l describes the minimal time between two consecutive events that trace, see Eq. 3.5. What the timestamp of an event a is defined as $ts(a)$. The time granularity of the event log ($G(L)$) is the smallest (non-zero) difference between any two consecutive events in an event log, see Eq. 3.6. Because process mining techniques rely on timestamp information, the precision of timing information can make a crucial difference for the quality of the mining results [35].

$$G(l) = min\{ts(b) - ts(a) | a, b \in l \wedge ts(a) < ts(b)\} \tag{3.5}$$

$$G(L) = \frac{\sum_{l \in L} G(l)}{|L|} \tag{3.6}$$

Variety The variety of an event log, $V(L)$ is the number of distinct event classes as is stated in Eq. 3.7. Since many control flow-oriented process mining algorithms generate a process model in which every event class is mapped to a distinct node, this dimension directly affects the complexity of the mining result. It is also possible to determine the trace with the lowest and the highest variety (i.e., M_{min} and M_{max} respectively).

$$V(L) = |C(L)| \tag{3.7}$$

Unique Traces This metric was not listed in the structural log metric from [35], but it is added here since it describes the variety in traces that appear in the event log (UT, also referred to as the number of process variants). A high number of unique traces is not uncommon in event logs of real-life processes, which can be complex and ad-hoc. A high UT creates problems for traditional process mining techniques, which cannot deal with such unstructured processes. A technique that can be applied to deal with this issue could be trace clustering [36].

Structure The structure of a log trace, $ST(l)$, is the inverse relative amount of direct following relations in l (i.e. compared to the maximum amount of directly following relations possible), see Eq. 3.8. The structure of an event log, $ST(L)$ describes the chunk of observed behavior, in comparison to the theoretically possible behavior, see Eq. 3.9). A low structure of an event log can be problematic for some process mining algorithms. Especially algorithms that strive to fit all the observed behavior in the resulting process model; which will likely be an unstructured "spaghetti

model". These models are hard to comprehend because "everything" is linked to "everything".

$$ST(l) = 1 - \frac{|F(l)|}{C(l)^2} \tag{3.8}$$

$$ST(L) = 1 - \frac{|F(L)|}{C(L)^2} \tag{3.9}$$

Affinity The similarity of two traces, with respect to the observed direct following relations, is defined as affinity of two log traces l and m (see Eq. 3.10). The mean affinity of an event log, $MA(L)$ is the average affinity between all log traces contained in the event log, as shown in Eq. 3.11. This depicts the homogeneity of an event log. This directly affects the structuredness of a mined process model. Event logs with low mean affinity are difficult for process mining algorithms because it is more challenging to extract and bring out mainstream behavior. It is also possible to determine the trace with the lowest and the highest structure (i.e., M_{min} and M_{max} respectively).

$$A(l,m) = \frac{|F(l) \cap F(m)|}{|F(l) \cup F(m)|} \tag{3.10}$$

$$MA(L) = \frac{\sum_{l \in L} \sum_{m \in L \setminus \{l\}} A(l,m)}{|L|(|L|-1)} \tag{3.11}$$

Data Quality Problems Problems with respect to event logs in process mining have been classified into two categories, process characteristics and quality characteristics [8]. In earlier work, 27 classes of event log quality issues were identified, e.g. problems related to timestamps in event logs, imprecise activity names, and missing events [8]. In this comparison framework, the goal is to use these classes to identify the quality issues for the event logs that are created with the different process mining approaches. By doing so, we can structurally compare the quality of the event logs in a detailed manner. The list of the possible qulity problems for event logs [8] is provided in table 3.3.

Table 3.3: Different entities of a log and four problem categories, adopted from [8].

Problem category	Affected entity of the log
Missing Data	Missing Cases
	Missing Events
	Missing Relationships
	Missing Case Attributes
	Missing Position
	Missing Activity names
	Missing Timestamps
	Missing Resources
	Missing Event Attributes
Incorrect Data	Incorrect Cases
	Incorrect Events
	Incorrect Relationships
	Incorrect Case Attributes
	Incorrect Position
	Incorrect Activity names
	Incorrect Timestamps
	Incorrect Resources
	Incorrect Event Attributes
Imprecise Data	Imprecise Relationships
	Imprecise Case Attributes
	Imprecise Position
	Imprecise Activity names
	Imprecise Timestamps
	Imprecise Resources
	Imprecise Event Attributes
Irrelevant Data	Irrelevant Cases
	Irrelevant Events

Level of Maturity The maturity is another measure to define the quality of an event log. The aim is to have the highest level of maturity for an event log. Events should be *trustworthy*, i.e., it should be safe to assume that the recorded events truly happened and that the attributes of events are correct [2]. Event logs should be *complete*, i.e., given a particular scope, no events may be missing. Any recorded event should have well-defined *semantics*. Moreover, the event data should be *safe* in the sense that privacy and security concerns are addressed when recording the events. The five event log maturity levels that have been defined in [2] are ranging from excellent quality ($\star\star\star\star\star$) to poor quality ($\star$). See table 3.4 for an overview.

Having explained all the performance metrics for *stage 3, data processing*, the next section will elaborate on how the actual efficiency and actual effectiveness are measured in *stage 4, mining & analysis*.

Table 3.4: Maturity levels for event logs, adopted from [2].

Ranking	Characterization
⋆⋆⋆⋆⋆	Highest level: the event log is of excellent quality (i.e., trustworthy and complete) and events are well-defined. Events are recorded in an automatic, systematic, reliable, and safe manner. Privacy and security considerations are addressed adequately. Moreover, the events recorded (and all of their attributes) have clear semantics. This implies the existence of one or more ontologies. Events and their attributes point to this ontology.
⋆⋆⋆⋆	Events are recorded automatically and in a systematic and reliable manner, i.e., logs are trustworthy and complete. Notions such as process instance (case) and activity are supported in an explicit manner.
⋆⋆⋆	Events are recorded automatically, but no systematic approach is followed to record events. However, unlike logs at level ⋆⋆, there is some level of guarantee that the events recorded match reality (i.e., the event log is trustworthy but not necessarily complete). Consider, for example, the events recorded by an ERP system. Although events need to be extracted from a variety of tables, the information can be assumed to be correct (e.g., it is safe to assume that a payment recorded by the ERP actually exists and vice versa).
⋆⋆	Events are recorded automatically, i.e., as a by-product of some information system. Coverage varies, i.e., no systematic approach is followed to decide which events are recorded. Moreover, it is possible to bypass the information system. Hence, events may be missing or not recorded properly.
⋆	Lowest level: event logs are of poor quality. Recorded events may not correspond to reality and events may be missing. Event logs for which events are recorded by hand typically have such characteristics.

Table 3.5: Metrics for the actual efficacy of a process mining approach in stage 3: data processing.

Metrics to determine actual efficiency (input measures)

Index	Label	Brief description
3.1	TIME	Time required to complete stage
3.2	IT-S	Iterations within current stage
3.3	IT-OS	Iterations to previous stages
3.4	DK-O	Number of objects in required domain knowledge
3.5	DK-R	Number of relations in required domain knowledge
3.6	COMPL	The task complexity

Metrics to determine actual effectiveness (output measures)

Index	Label	Brief description
3.7	M	Magnitude (# number of events)
3.8	SP	Support (# number of cases)
3.9	AE	Average number of events per case
3.10	LoD	Level Of Detail
3.11	G	Time Granularity
3.12	V	Variance
3.13	V_{min}	Minimal Variance
3.14	V_{max}	Maximal Variance
3.15	UT	Unique traces in the event log
3.16	ST	Structure
3.17	ST_{min}	Minimal Structure
3.18	ST_{max}	Maximal Structure
3.19	MA	Mean Affinity
3.20	A_{min}	Minimal Affinity
3.21	A_{max}	Maximal Affinity
3.22	DQ-PR	Data Quality Problems
3.23	MAT	Level of Maturity

3.4 Measure the Actual Efficacy in Stage 4 of PM²

Stage four, as it is described in table 3.1, is about the actual application of process mining techniques on the event logs. Four types of activities that have been identified are: process discovery, conformance checking, enhancement and process analytics. Figure 3.7 shows an overview of these activities. *Process Discovery*, involves constructing a process schema based on an event log. In the article "Towards an Evaluation Framework for Process Mining Algorithms", published in 2007, the first effort was made towards the development of common means of assessing the quality of process models discovered with PM Algorithms. In this paper four dimensions for the evaluation of process models; *fitness*, *precision*, *generalization* and *structure*, also different measures for these dimensions have been listed. The four dimensions are still actively used in conformance checking today. In later work, more frameworks for comparing process mining algorithms were presented [37, 38, 39, 40]. The research on quality dimensions for process discovery has evolved to a domain on itself. *Conformance Checking* is a family of PM techniques to compare an existing process model with an event log of the same process. The goal is to check if the actually logged behavior of a (business) process conforms to the model and vice versa. *Enhancement* is defined as extending or improving an existing process model using information about the actual process recorded in an event log [1], e.g., by extending the process model with performance information related to time or cost, or repairing the process model according to current behavior from the corresponding event log. Resulting from this activity, one derives process models, enhanced on different aspects. *Process analytics* consist of other complementary analysis techniques, e.g. data mining and visual analytics.

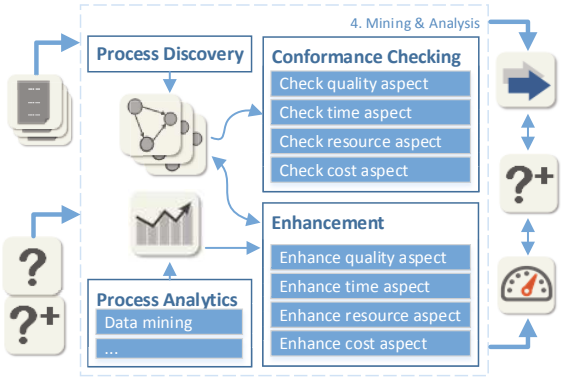


Figure 3.7: Overview of the activities in mining & analysis stage in PM².

Input measures Table 3.6 presents all the performance metrics for the Mining & Analysis stage. As was mentioned before, the comparison framework that we propose in this work focuses mainly on stage 2 and stage 3. For the fourth stage, less quantitative performance metrics are applied. For example, the time to complete this task is not recorded in this stage. Also, it is assumed that all the domain knowledge that is required for completing the two previous stages suffices to complete this stage. Therefore, measuring the domain knowledge required for this stage is irrelevant. Also, the iterativeness in the mining & analysis stage is not convenient for comparison either, since this is a highly iterative stage in PM². The task complexity is still a valid performance metric for the actual efficiency. This metric is applied just like in the previous two stages.

Table 3.6: Metrics for the actual efficacy of a process mining approach in stage 4: mining & analysis.

Metrics to determine actual efficiency (input measures)

Index	Label	Brief description
4.1	COMPL	The task complexity

Metrics to determine actual effectiveness (output measures)

Index	Label	Brief description
4.2	SEM	Executable semantics
4.3	PD	Process discovery techniques
4.4	CC	Conformance checking techniques
4.5	ENH	Performance enhancement techniques

Output measures It is important to keep in mind that this work does not focus on the comparison of different PM algorithms. Therefore, at least in this research, this specific comparison is out of scope. The comparison framework should be defined to compare process mining approaches from a higher point of view in this particular stage. Thus, for measuring the actual effectiveness of a process mining approach regarding in the mining & analysis, we defined a more high-level set of output measures. We compare the options and 'functionalities' that are offered by a PM approach. In this stage, the comparative study will assess a process mining approach based on the format of discovered process model(s), the process discovery technique(s), conformance checking, and process model enhancement. Fig. 3.8 visualizes the operationalization of the comparison framework in stage 4 of PM^2.

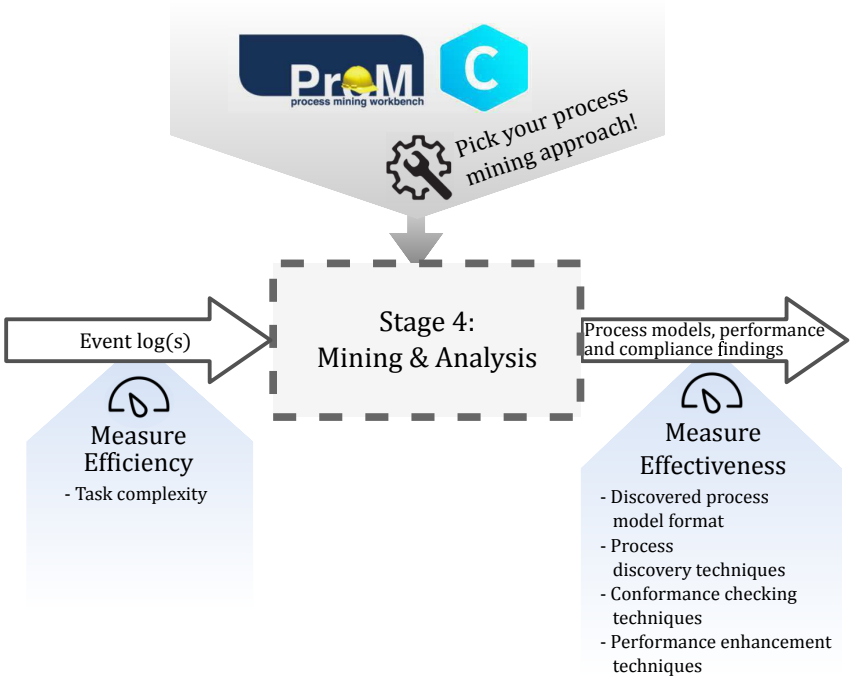


Figure 3.8: Comparison framework for PM approaches applied to the fourth stage of PM^2.

We defined four metrics to determine the actual effectiveness for a PM approach in the fourth stage of PM^2.

Firstly, to assess the quality of a discovered model, we check if a process model has *executable semantics*. A process model with executable semantics can be mapped onto a transition system (and therefore easily be translated to higher-level languages such as Petri Nets, BPMN, EPCs, and UML diagrams) [11]. Other tools for process discovery (or process exploration), such as the Fuzzy Miner, Fluxicon Disco, Celonis process mining are based on showing directly follows-graphs: nodes are process steps, and in general edges mean that an activity followed another [41]. One can inspect the process by considering the arrows between them. These process maps can be intuitive visualizations but have some limitations. E.g. plain directly-follows based maps do not show parallelism. Which would imply that in a map, the state of the system only depends on the last executed process step, which is not realistic. Some tools do support parallelism by sacrificing executable semantics. However their maps *cannot be used for automated analysis*; they do not show critical features (e.g., types of splits and joins) and it is impossible to reason over them (e.g., which traces are viable) [41].

Secondly, the number and quality of process discovery techniques that are offered withing a PM approach. The metric for this is defined as the diversity of process discovery techniques that are supported by a certain PM approach. This is quantified straightforward with a number, ranging from a small to a large amount of discovery techniques.

Thirdly, conformance checking is addressed. This study will not compare these techniques, as

they are 'comparison' techniques themselves. Also, conformance checking is a separate field of study, which is left out of scope in this thesis. For the comparison framework, we merely include a check whether conformance checking is possible or not with a PM approach.

Lastly, to address enhancement, i.e. taking an event log an existing process model to extend or improve the model using observed events, each PM approach is assessed. This assessment is based on how the enhancement is supported (or not) by a PM approach.

3.5 Adoption of a Process Mining Approach in Practice

As indicated in section 3.1, the adoption of a process mining approach in practice is a dimension that determines the success of a PM approach. The higher the adoption in practice, the more successful a process mining approach (assuming that the performance of the PM approach is not poor). In Moody's model for method evaluation, it is explained that (process mining) approaches will be adopted based on *perceptions* of their ease of use and usefulness [16]. Moreover, that the actual efficiency and effectiveness determine intentions to use a method only via perceptions of ease of use and usefulness. Moody emphasizes that in human behavior, *subjective reality* is more important than *objective reality*. This is illustrated in fig. 3.2.

As was stated in section 3.1, the proposed comparative framework for process mining approaches, both the subjective and the objective reality are considered to be important. However, specifically for the case study, we mainly focus on the objective reality. The reason is that the subjective reality can only be measured by one person, namely the author of this thesis, which results in unreliable outcomes. For future work, we suggest that this subjective reality would better be measured in a similar study, but then with multiple persons to assess the PEoU and PU.

Moody states that following from TAM, the intention to use a method is jointly determined by its perceived ease of use and perceived usefulness. And, actual usage of a method is determined by the intention to use it (see fig. 3.2 again). Thus, by estimating the PEoU and PU, we can determine the perceived efficacy of the different process mining approaches and compare them. This enables the researcher to indicate, to a minor and small extend, which process mining approach would possibly have the highest adoption in practice.

To measure the PEoU and PU, we adopted the measurement scales for perceived ease of use and perceived usefulness that have been validated in earlier work [17]. In this earlier work both perceived usefulness and ease of use were significantly correlated with self-reported indicators for using the system. The measurement scales are presented below in table 3.7. For each of them, a user is to determine the score on a seven-point scale, one with likely-unlikely end-point adjectives. The dots are a placeholder for any type of process mining approach that is to be assessed.

Table 3.7: Measurement Scales for Perceived Usefulness and Perceived Ease of Use

Perceived Ease of Use	
Q_1	Learning to operate _PM would be easy for me.
Q_2	I would find it easy to get _PM to do what I want it to do.
Q_3	My interaction with _PM would be clear and understandable.
Q_4	I would find _PM to be flexible to interact with.
Q_5	It would be easy for me to become skillful at using _PM.
Q_6	I would find _PM easy to use.
Perceived Usefulness	
Q_1	Using _PM in my PM project would enable me to accomplish tasks more quickly.
Q_2	Using _PM would improve my Process Mining project performance.
Q_3	Using _PM in my Process Mining project would increase my productivity.
Q_4	Using _PM would enhance my effectiveness on my Process Mining project.
Q_5	Using _PM would make it easier to do my Process Mining project.
Q_6	I would find _PM useful in my Process Mining project.

* For *_PM* any of the approaches to compare can be filled in.

Chapter 4

Verification

In order to be able to perform the actual comparative study, using the proposed comparison framework presented in chapter 3, first the redo log process mining approach is applied in the real-life environment (i.e. the case study). Up until now, this approach was only tested with a dummy database [3]. Therefore, a verification of the application of the RLPM approach is required. To this end, a Research Goal was defined already in section 1.3, see below.

RG 2. *Apply and verify the redo Log process mining approach in a real-life case study.*

This research goal is achieved by, firstly, extracting an event log in the real-life case study with the Process Aware Data Suite (PADAS). Secondly, the event log can be verified by comparing it with a second event log that is created with the traditional process mining approach. If the two event logs are equivalent, the redo log process mining approach is proven to be valid. Lastly, process models mined on each of the event logs can be compared. This chapter presents the results of the aforementioned.

The remainder of this chapter is structured in the following manner. Firstly, the *requirements* for the application of the RLPM approach are discussed in section 4.1. Secondly, section 4.2 explains how *missing data* in the archived log files is handled. Lastly, section 4.3 provides the *verification* of the redo log process mining approach.

4.1 Requirements for Deployment

This section presents all the requirements that had to be met to successfully implement the RLPM approach using the PADAS application. These requirements are the following: the `SUPPLEMENTAL LOGGING` should be enabled (1); archived logs should be available (2); the archived logs should cover a sufficient period of time (3); the archived logs should be complete (4); a snapshot or clone of the database should be available at the start date or end date of the archived logs (5); it is required to have access to the database as a `DBA` or as a `COMMON USER` (6); one should have the `SELECT ANY TRANSITION`(7) and `SET CONTAINER`(8) privileges; a correct dictionary of the database (in case of a multitenant architecture, the pluggable database)(9); PADAS should be installed locally (e.g. on the server) or connected remotely (e.g. with a SSH tunnel to the server) and connected to the database (10). We visualized all thesis requirements in flow chart, to check of redo log process mining with PADAS is feasible, is given in appendix A.2.

4.2 Dealing with Missing Archived Logs

In the case study, one of the requirements mentioned in section 4.1 was violated, namely that the archived logs should be complete. During the case study, once in a while, archived logs were deleted by Oracle. When limited storage space was available, Oracle removed archived logs that

were logged before the latest backup of the database. As a consequence, the event log that was extracted with PADAS contained traces with missing events. This would make any analysis on the event log unreliable. However, we present two manners to resolve this issue.

One way is to remove the incomplete traces in the event log. Firstly, the days with the missing archived logs were identified, i.e. the *gaps*. For a visual representation of the archived logs, missing and non-missing, see fig. 4.1. Next, with the aid of the `TICKET_HISTORY` table that stores the event data that is used for the traditional process mining approach, it was possible to the identify the traces with events with timestamps in the gaps. By removing these incomplete traces from event log of the RLPM approach, part of it was 'saved'. This partial event log was used for the comparative study, discussed in the analysis; appendix B.

Another way to preserve part(s) of the event log with the gaps, is to only select events that occur in a time-frame in which no gaps occurred. After having identified the gaps (again, see fig. 4.1), the most convenient way is to identify the longest set of consecutive archived logs where no missing archived logs are found. That would be the period of time between and including 17-06-2016 and 12-07-2016. By selecting this part of the event log through filtering, it is possible to derive a reliable subset of the event log. This method, for dealing the missing data, was chosen for the verification of the RLPM approach because the subset of the event log is 'unprocessed'. Moreover, we decided to compare the raw event logs (i.e., the cases are not merged). This allows for a verification of the most pure (i.e., original, without data processing) event log created with PADAS. The next section elaborates on the actual verification.

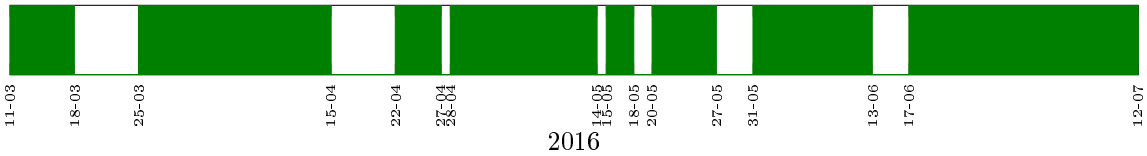


Figure 4.1: Missing archived logs over time, where the greenly shaded time periods indicate the availabilty of archived logs and the white space indicates the gaps.

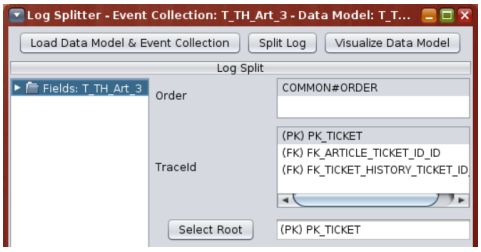
4.3 Verification of the Redo Log Process Mining approach

To verify the redo log process mining approach; an event log created with this approach is compared to an event log created with the traditional process mining approach. The chosen criteria for the equivalency of the event logs are: both event logs should include the same cases, for every event in the TPM event log it should be possible to create a mapping to at least one event of the RLPM approach, and the timestamps of mapped events should be equal.

For the TPM approach a query was executed; extracting only the case identifier, the activity identifier and timestamps, see fig. 4.2a. And for the RLPM approach three tables were extracted; the `TICKET`, `TICKET_HISTORY` and `ARTICLE` tables with the primary key of the `TICKET` table as a root, see fig. 4.2b. In this event log, the largest consecutive period of time without missing archived logs is 26 days. For a fair verification, the selection of event data both the TPM and RLPM approaches should be made for the same time period. Therefore, both event logs were filtered to the exact 26 days; all events before and after this period of time were removed from both event logs.

```
SELECT
  th.TICKET_ID          "T:concept:name",
  tht.NAME              "E:concept:name",
  concat(to_char(th.CREATE_TIME,'YYYY-MM-DD HH24:MI:SS'),'.' ||
  substr(to_char(th.ID), length(to_char(th.ID))-2, 3))
                        "E:time:timestamp"
FROM
  OTRS_OWN.TICKET_HISTORY th
LEFT OUTER JOIN
  OTRS_OWN.TICKET_HISTORY_TYPE tht
ON th.HISTORY_TYPE_ID = tht.ID
ORDER BY th.CREATE_TIME ASC;
```

(a)



(b)

Figure 4.2: Extraction: in a with a SQL query for TPM and in b the trace ID pattern with PADAS.

Table 4.1: Characterisics of the event logs created with the TPM and RLPM approaches.

	TPM	**RLPM**
Number of cases	907	907
Number of events	6342	7866
Number of distinct event classes	22	22
Timestamp first event in the log	Fri Jun 17 00:55:49 CEST 2016	Fri Jun 17 00:55:49 CEST 2016
Timestamp last event in the log	Tue Jul 12 21:34:19 CEST 2016	Tue Jul 12 21:34:19 CEST 2016

Table 4.1 presents a selection of event log metrics for the two event logs. The number of cases in both event logs is exactly 907 (it has also been established that these are the same cases). Also, the timestamps of the first events in the logs are similar, just like the timestamps of the last events in the logs. These analogies prove the validity of the redo log mining approach, since one can derive the same historical event data with both the TPM approach and the RLPM approach.

Unintentionally, the number of distinct event classes is equal (i.e., 22) for the event logs. Mind that this does not verify anything since the labels of the events are completely different. The TPM approach uses a readily available classifier present in the database, where the RLPM combines the table name, the type of operation and the change at the column-level. The different techniques used for defining the event classifier also justify the difference in the number of events (i.e. the magnitude). The classifier of the redo log process mining approach is very low-level, but also very generic. The difference in the magnitude (i.e., the number of events), characterizes, the difference between the TPM approach and the RLPM approach. This is visualized in fig. 4.3. At least in this case study, with the RLPM approach we can log more (fine-grained) events compared to the TPM approach.

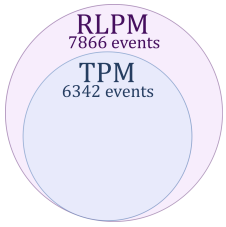


Figure 4.3: Venn diagram of two sets of event data.

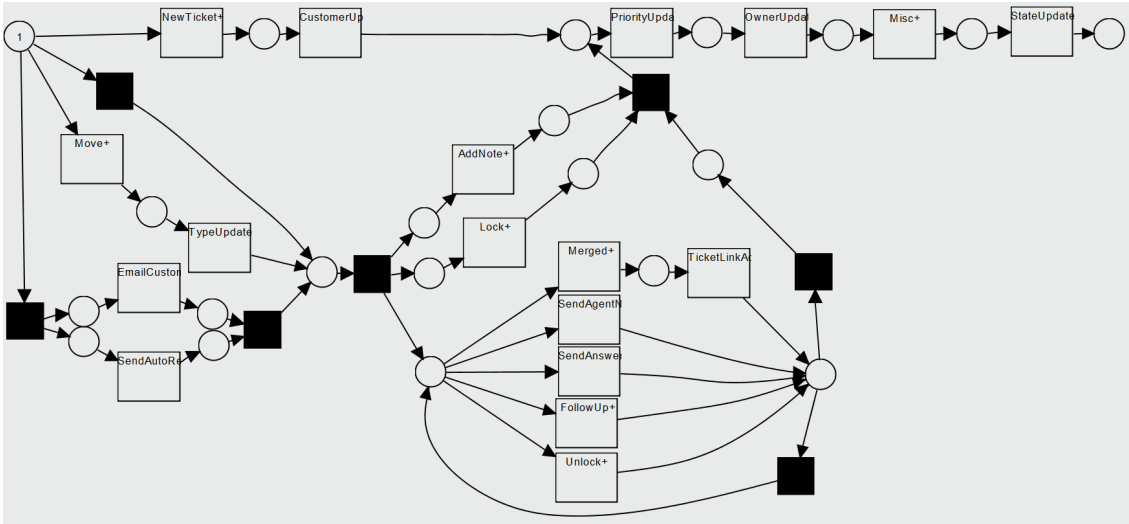
Accordingly, it is interesting to have a more detailed look at the activity labels (i.e., event classes). We wanted to link the activities of the RLPM approach to those of the TPM approach. This has been accomplished partially: it turned out that a complete mapping cannot be made. Table 4.2 presents all the activity labels (i.e. event classes) of both event logs.

Table 4.2: The distinct event classes for the event classifiers of the TPM and RLPM approach.

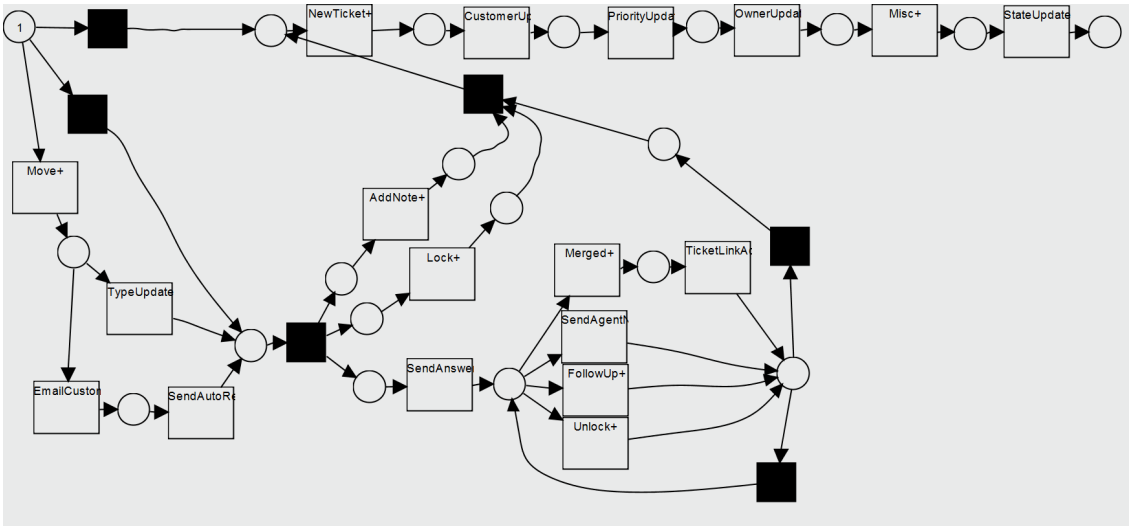
Traditional process mining			**Redo log process mining**		
Activity label	Frequency	Relative frequency	**Activity label***	Frequency	Relative frequency
`Misc`	1 182	18.64 %	`New event (without message)`	5 032	63.97 %
`OwnerUpdate`	976	15.39 %	`New event & message`	1 310	16.65 %
`StateUpdate`	919	14.49 %	Message by phone OR note internal	800	10.17 %
`CustomerUpdate`	883	13.92 %	Message, ticket merged	117	1.49 %
`NewTicket`	882	13.91 %	Auto-reply (ticket received)	74	0.94 %
`SendAgentNotification`	398	6.28 %	ARTICLE+INSERT+4444414144444414444444	70	0.89 %
`AddNote`	168	2.65 %	ARTICLE+INSERT+4444414444444414444444	63	0.8 %
`Lock`	165	2.60 %	ARTICLE+INSERT+4444111141111414444444	51	0.65 %
`Unlock`	164	2.59 %	Update message (ticket_id, time & user)	49	0.62 %
`Merged`	102	1.61 %	Update event (ticket_id, time & user)	49	0.62 %
`TicketLinkAdd`	102	1.61 %	New email or note (customer/agent)	40	0.51 %
`FollowUp`	94	1.48 %	ARTICLE+INSERT+4444411144411414444444	38	0.48 %
`EmailCustomer`	82	1.29 %	Update event (ticket_id & time)	34	0.43 %
`SendAutoReply`	72	1.14 %	Update message (ticket_id & time)	31	0.39 %
`SendAnswer`	63	0.99 %	Update message (ticket_ID & user)	29	0.37 %
`Move`	42	0.66 %	Update event (ticket_ID & user)	29	0.37 %
`PriorityUpdate`	26	0.41 %	New message (customer or agent)	19	0.24 %
`TypeUpdate`	11	0.17 %	New Email-external	10	0.13 %
`SendCustomerNotification`	4	0.06 %	Update message (only ticket_id)	9	0.11 %
`TimeAccounting`	3	0.05 %	Update event (ticket_id)	9	0.11 %
`SetPendingTime`	2	0.03 %	Email from customer without CC	2	0.03 %
`SendAutoFollowUp`	2	0.03 %	Email from customer with CC	1	0.01 %
Total	6342	100 %	Total	7866	100 %

*The activity labels of the RLPM event log are relabeled to a more convenient, understandable meaning when possible.

All the RLPM activities with the labels `New event (without message)` and `New event & message`, that is 6342 activities, are equal to the total number of events in the TPM event log.
To be more precise, `New event (without message)` can imply TPM activities like `Misc`, `OwnerUpdate`, `StateUpdate`, `CustomerUpdate`, `NewTicket`, `SendAgentNotification`, `Lock`, `Unlock`, `Merged`, `TicketLinkAdd`, `Move`, `PriorityUpdate`, `TypeUpdate` or `SetPendingTime`. And `New event & message` can indicate a TPM activity like `OwnerUpdate`, `StateUpdate`, `AddNote`, `FollowUp`, `EmailCustomer`, `SendAutoReply`, `SendAnswer`, `SendCustomerNotification`, `TimeAccounting` or `SendAutoFollowUp`.
The event log of the RLPM approach actually contains the activity labels of the TPM approach; this information is logged by PASAS as an event-attribute. Thus we can remap the activity label, using this attribute, to achieve the same activity notion of the TPM approach. Having established that we can create this mapping, we can state that the RLPM approach actually provides a superset of the events that were extracted with the TPM approach. This makes perfect sense since each TPM event is represented by an instance of the `TICKET_HISTORY` table, and each of the two aforementioned RLPM events is created through an insertion in this same `TICKET_HISTORY` table. Next to these two events, the event log produced with the RLPM approach contains more events, i.e considering operations on the `ARTICLE` and `TICKET` tables.



(a) TPM



(b) RLPM

Figure 4.4: Process models mined with the Inductive Miner, Infrequent (noise threshold = 0.2, RapidMiner 7.3.001 and RapidProM 4.0.0): a for the TPM approach and b for the RLPM approach.

Additionally, for an all-round verification, we compared two process models mined from the event logs of the two PM approaches. Firstly, an inverse mapping was created; from the two earlier discussed RLPM activities to the corresponding activity labels in the event log created with the TPM approach. Next, all the other (extra) activities in the RLPM event log were filtered out. Accordingly, process models were discovered using the event logs of both approaches, see fig. 4.4. These process models have roughly the same structure. Also, it has been established that each log contains the exact same set of activities with the same frequencies. Still, some differences can be spotted when comparing the two models, these are listed below.

- The activities `EmailCustomer` and `SendAutoReply` are executed subsequently in the model created with the TPM approach, whereas they are executed in parallel for the RLPM approach.
- Another difference is found in the XOR split with five choices in the event log of the TPM approach, where the `SendAnswer` activity will be executed in parallel with the other activities. However, it is executed before the big XOR split in event log of the RLPM approach.
- In the event log of the TPM approach, each trace will end with a series of four events (`PriorityUpdate`, `OwnerUpdate`, `Misc` and `StateUpdate`). However, each trace in the event log of the RLPM approach will end with two additional activities (`NewTicket`, `CustomerUpdate`) before these four activities.

The first two aforementioned points, show differences in parallelism/serial executions. We suspect that these differences are caused by differences in handling events that occur in the same second on the application level and on the redo log archiving level. These two might handle simulations events in different order. Moreover, it was discovered that when a certain ticket is merged to another ticket, the `TICKET_ID` for some instances in the `TICKET_HISTORY` table are updated. Because the events of the TPM approach are actually instances of this table, events belonging to merged tickets will belong to the *updated* version of the `TICKET_ID`. This is different for the RLPM approach. PADAS records such events, belonging to a merged ticket, with the `TICKET_ID` that the event has when it is inserted in the table, thus the *original* identifier. Thus with the RLPM approach, some events belong to other tickets (i.e. cases).

Having stated the above, this chapter illustrated that the redo log process mining approach is equivalent to the traditional process mining approach. We found that the event log of the RLPM approach is a superset of the event log of the TPM approach. Furthermore, we can create a mapping for the RLPM event log that results in the same activity labeling of the TPM approach. And, both event logs consider the same cases. Moreover, the timestamps of the events are equivalent. Moreover, the process models that are mined with these event logs are similar of structure. The differences between the models consider the ordering of the events, the causes for these differences have been explained in the previous paragraph. All told, it is confirmed that process mining with redo logs is a valid approach. This concludes the verification of the RLPM approach. Next, the actual comparative study was performed. Chapter 5 provides a summary of how the the three process mining approaches under study are applied in real-life. Chapter 6 then discusses the results with respect to the comparison framework that proposed previously in the methodology chapter.

Chapter 5

Summary Case Study Analysis

This chapter summarizes the completion of a process mining project in a real-life case study, applying the proposed *comparison framework* from chapter 3, on three process mining approaches (i.e. TPM, RLPM, and CPM). This chapter tries to fulfill the requirements of Research Goal 3, introduced in section 1.3 and shown below.

> **RG 3.** *Perform the comparative study using the proposed comparison framework on process mining approaches in a real-life case study.*

The comparison framework uses PM^2 (i.e., Process Mining Project Methodolody from [6]) as a guidance to define specific tasks to complete a process mining project. PM^2 identifies 6 stages (which are explained previously in section 2.2.2 and 3.1). This chapter provides a summary of the completion of these stages, in chronological order, per stage. More detailed documentation on the case study analysis is given in appendix B.

Stage 1: Planning This paragraph summarizes the activities that were performed to complete the planning stage. In this stage, the three PM approaches were executed similarly. Firstly, the project goal and the process mining team were defined. Next, the business process was selected. Also, the *business process questions* were defined, see below.

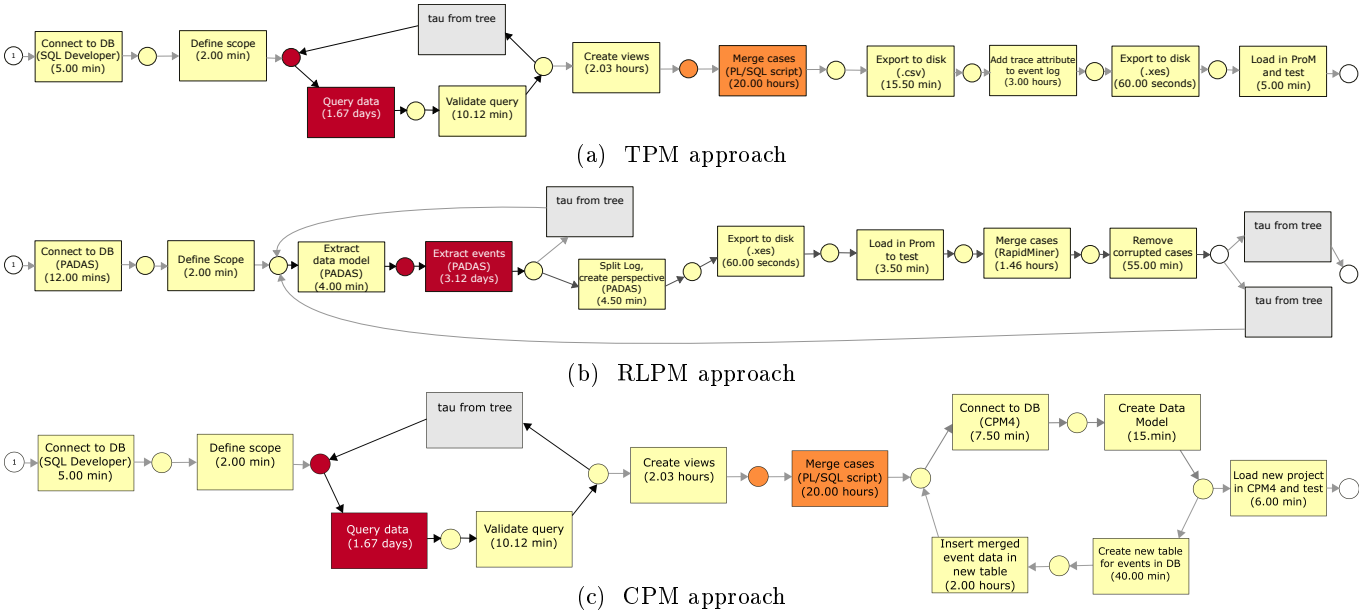
> **BPQ 1.** How many tickets (i.e. cases) conform to the SLAs with respect to their response time and solve time (i.e. throughput time)?
>
> **BPQ 2.** Are distinct customers treated differently in the process (e.g., with respect to the workflows and with respect to solve time and response time)?
>
> **BPQ 3.** What are the differences in the performance of the distinct resources?

Accordingly, the tools and techniques that are required in the next stages were selected, see table B.2 for an overview. The selected *IS* that stores the historical data of the selected process is similar for the TPM and CPM approaches. It is the OTRS application that logs the events in the `TICKET_HISTORY` table in the supporting database. The *IS* that logs the events for the RLPM approach is Log Writer Process (LGWR), which is a background process of an Oracle instance, responsible for redo log buffer management. An additional observation made during the completion of this stage was that *cases merge* during their lifetime, which means that some tickets are linked to another ticket.

Addtionally, the RLPM approach required a number of extra planning activities: correctly configuring the database to ensure the redo logs are archived correctly, ensuring the availability of storage facilities for these archived logs, extracting the LogMiner dictionary to a flat file, preparing the setup for the extraction (i.e. ensuring the PADAS application is installed on the server, etc.), and creating a snapshot of the database.

See appendix B.1 for a more detailed description about how this stage was completed. The next paragraph continues by discussing how the second stage of PM^2 was performed.

Stage 2: Extraction According to the comparison framework defined in chapter 3, data was collected on the process mining activities (e.g. the duration of tasks). This particular data was analyzed again, which could be considered some form of meta-mining. The steps (i.e. activities) that have been performed for each of the PM approaches completing the second stage of PM^2 are visualized with process models, see fig. 5.1.



(a) TPM approach

(b) RLPM approach

(c) CPM approach

Figure 5.1: Process models of the completion of *Stage 2: Extraction* of PM^2 for the case study, with the TPM approach in fig. 5.1a, the RLPM approach in fig. 5.1b, and the CPM approach in fig. 5.1c[1].

A major difference between the three process mining approaches became clear in the scoping step. The historical data used for the TPM and CPM approaches covered a vastly larger period of time than the redo logs that were archived during the project.

One can observe in fig. 5.1 that the first six activities of the TPM and CPM approaches overlap completely. In this case study, the TPM and CPM approaches require querying activities to collect all the data of interest in order to answer the research questions. SQL queries were executed out the RMDBS to obtain the required data. To be able to easily access the results, the selection was inserted in new *views* on the database. Creating the final query was a highly iterative activity, and also the most time consuming step in the process for the TPM and CPM approach. This activity is automated with PADAS, which is used for the RLPM approach. No query needs to be written since, with PADAS, one can extract the complete data model from the database with one click. Next, one needs to determine which tables on the database contain data of interest, select the names of these tables, and click the "Extract Redo Log" button. PADAS automatically collects all the events (i.e., insertions, updates and deletions) that have occurred on these tables. This is a straightforward step, but also very time consuming. Afterwards, one defines the trace ID pattern, which is essentially the case identifier.

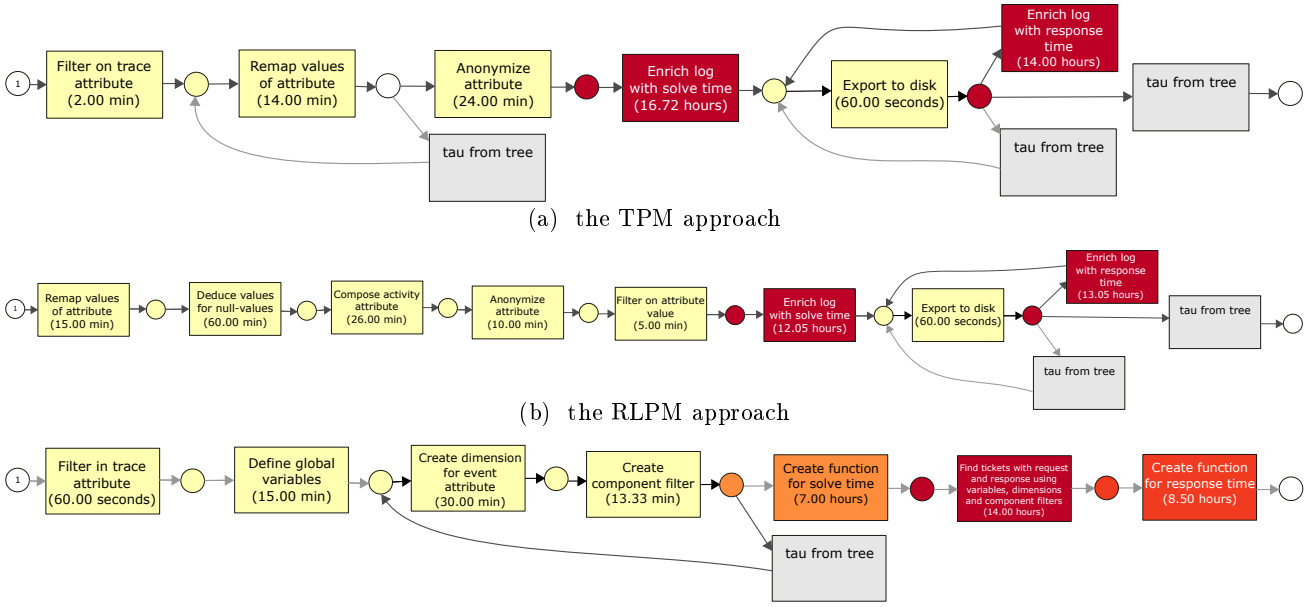
In this case study, additional steps were performed to anticipate the merging behavior. This behavior is visualized in detail with an example in appendix B.2.1. To anticipate this merging behaviour, the case IDs of the events needed to be updated. However, updating the earlier created views was not an options since that would ultimately update the base tables of these views. Therefore, with the TPM approach, we created a Global Temporary Table (GTT) and collected the data there. Accordingly we updated the GTT with a PL/SQL script to merge the traces. However, this GTT was session specific and is non existent in a second session (e.g. a session of

[1] All these process models are mined with the Inductive Miner, with 0.0 noise threshold, and accordingly analyzed for performance (manifest) with RapidMiner (7.3.001) and the RapidProM extension (4.0.1).

Celonis). Therefore, for the CPM approach, it was required to create a new table in the database. This table with the event and a view with case data is loaded in Celonis via the connection with the database. Next, the data model can be defined, which is, in this case, only one relation between the case view and the event table. For the RLPM approach, a groovy script was written with RapidMiner, which updates the event log that is created with PADAS. Thus, no views or tables needed to be created in the database for the RLPM approach.

For a detailed description of the completions of this stage of PM^2 see appendix B.2.1, B.2.2, and B.2.3 for the TPM, RLPM, and CPM approaches respectively. The results for all the metrics of the comparison framework are presented in appendix B.2.4. The next paragraph continues by explaining the third stage of PM^2.

Stage 3: Data Processing This paragraph is a summary of appendix B.3 in which the completion data processing stage is explained in detail. How each of the PM approaches completed the third stage of PM^2 is visualized in process models in fig. 5.2. These activities mainly focus on finding the solve and response times. These are the major performance indicators for the process at hand, since almost all SLAs are based on them. See fig. 5.3 for an example of the service window that needs to be taking into account when calculating the solvetime of a ticket with a low, medium, or high priority.



(a) the TPM approach

(b) the RLPM approach

(c) the CPM approach

Figure 5.2: Process models of the completion of *Stage 3: Data Processing* of PM^2 for the case study, with the TPM approach in fig. 5.2a, the RLPM approach in fig. 5.2b, and the CPM approach in fig. 5.2c.

For the TPM and RLPM approach, we mainly used RapidMiner. Additionally, we applied custom groovy and python scripts to calculate the solve times and response times.

One characteristic of the CPM approach is that it works completely in the single suite called CPM4. In which one starts with the import of the data, and continues the complete process mining project within CPM4. Celonis offers a set of standard process KPIs that, for example, count the amount of cases or throughput time between different activities. Moreover, a wide range of functions are available to compose advanced dimensions and calculate custom performance indicators (e.g., $!=$, $>$, $||$, *and*, *or*, $PU_LAST()$, $ABS()$, $MEDIAN()$ and more).

More details about the execution of this second stage of PM^2 is given in appendix B.3.1 for the TPM approach, in appendix B.3.2 for the RLPM approach, and in appendix B.3.3 for the CPM approach. The resulting measurements of the comparison framework are given in appendix B.3.4.

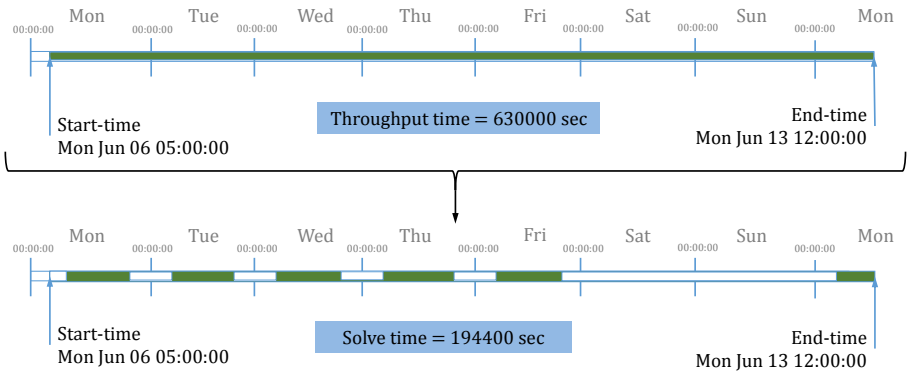


Figure 5.3: Calculating the solve time taking into account a restricted service window.

Stage 4: Mining & Analysis As explained in chapter 3, the comparison framework in this case study focuses less on the fourth stage. This stage was performed with the TPM approach and the CPM approach. Due to the gaps in the archived logs, the resulting event logs from the previous stage with the RLPM approach were too small to perform analysis with reliable results. However, the RLPM approach would be executed like the TPM approach, thus the TPM is representative for the RLPM approach.

For the TPM approach, the following techniques were applied: the Dotted Chart in RapidMiner, the Process Comparator, social network mining in ProM, and other custom functions in RapidMiner. For the CPM approach, a dashboard was created (see appendix B.10, B.11 and B.12) that uses different types of components (e.g., olap tables, text components, single KPIs, histograms, column charts, pie charts, process explorers, variant explorers and dropdowns).

The resulting performance of the process at hand, for the SLAs (i.e., BPQ 1), gave an exact match for the two different process mining approaches. However, since CPM4 does not offer the analysis techniques, some questions could not be answered by the CPM approach. These were mostly questions with respect to comparing workflows of two event logs of the same process and about social networks. Lastly, the answers to the BPQs were documented.

Detailed documentation on how this stage was performed is provided in appendix B.4.1, B.4.2 for the TPM and CPM approach respectively. Lastly, the resulting measurements of the comparison framework are given in appendix B.4.3.

Stage 5: Evaluation This stage is similarly executed for the process mining approaches under study. This is the current stage the process mining project of this case study is in. The evaluation will be performed by presenting the findings of the process analysis to the workers of Experis Ciber. After this presentation, the results will be discussed together, by which we intend to relate the results to improvement ideas. In initial evaluation of the case study findings can be found in appendix B.5. This section explains among others that we found that the logging of the events (for all PM approaches) is slightly ambiguous. Also, we identified a case variant, which mostly constitutes of automatically generated events, which might have skewed the analysis. Lastly, it was noted that the response time seems to be a less reliable performance indicator because of the low amount of cases that could be analyzed, and it was cumbersome to calculate it.

After the evaluation the final stage of PM^2 would be the *process improvement & support*, in which the improvements are to be implemented at the organization under study. These activities are irrelevant for our comparative study and are therefore outside the scope of this research.
Wrapping up, this chapter summarized all the activities that were executed to perform the first five stages of PM^2. Simultaneously, as was mentioned earlier, the comparison framework was applied. The comparison framework defines a set of metrics to analyze the the performance of a process mining approach with respect to the actual efficiency, actual effectiveness, perceived ease of use and perceived usefulness. To that end, we collected information about the execution of the stages while performing the case study analysis. The next chapter presents and discusses the results for the metrics of the comparative study for this real-life process mining project. Moreover, the next chapter aims to truly compare of the different process mining approaches.

Chapter 6

Results

From chapter 2, we know that an approach (i.e. method) seeks to increase the efficiency and effectiveness of a task performance. To quantify these two dimensions, we specified input and output measures for process mining tasks in the methodology (i.e. chapter 3). This chapter discusses the resulting measurements of the comparison framework using the TPM, CPM and RLPM approaches in the case study. As shown in chapter 5, all PM approaches are executed in stage 2 and 3 of PM^2. In stage 4, the TPM approach and the CPM approach are compared (recall that the RLPM and TPM approaches are equal for stage 4). In section 6.1 the differences in the actual efficiency of the PM approaches are analyzed. Next, section 6.2 discusses the findings with respect to the actual effectiveness of the PM approaches. And section 6.3 discusses the perceived efficacy of PM approaches. Next, in section 6.4, as a result of the comparative study, we explain the implications of the schema model style of (R)DBMS on the process mining approach. Lastly, section 6.5 provides a cross-dimensional comparison of the PM approaches.

6.1 Actual Efficiency of Process Mining Approaches

Different methods for performing a task vary in their performance on efficiency [16]. Efficiency of an approach is defined by the effort required to apply it, e.g. the *time* that is required for a stage in a process mining project. In chapter 3, time is defined as an *input measure* to estimate the efficiency of the approach that is applied to a task. Other input measures of the comparative framework are: *iterativeness*, *required domain knowledge* and *task complexity*. The results for the input measures, following from the application of the three PM approaches in the case study are discussed in this section.

6.1.1 Comparison per Input Measure

Time The time required to complete a stage is a direct input measure to quantify the efficiency. To compare the measurements for time, an analysis was performed by visualizing the time required for stage 2 and 3 of PM^2 for all the PM approaches in fig. 6.1.

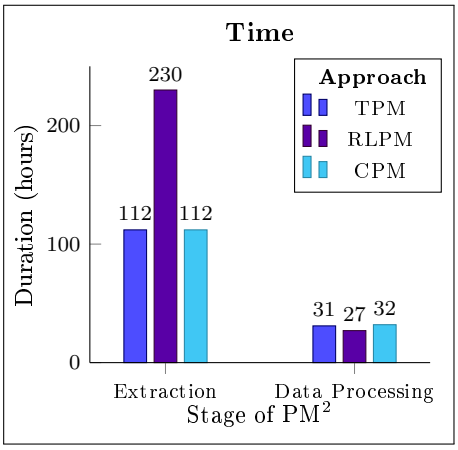


Figure 6.1: Time plotted per PM approach for stage 2 and 3 of PM^2.

We found that stage 2, extraction, takes more than twice as much time with the RLPM approach (230 hours), as with the TPM or CPM approach (112 hours). This is mainly due to one particular time-consuming step, which is the actual extraction of events (events being operations on the tables of the database) for a selection of tables of interest with PADAS. For the final extraction, 45 hours were required. Apart from that, dealing with cases that merge and the gaps in the archived logs required a lot of time. The TPM & CPM approaches require less, but still a fair amount of time; 14 workdays of 8 hours, spent on figuring out how to merge the cases properly. Also, time was needed to define, redefine and select the data of interest.

Accordingly in stage 3, event logs are created from the extracted event data. The time required for this stage was on the same order of magnitude for all three approaches (i.e. 31, 27, and 32 hours for the TPM, RLPM, and CPM approach respectively).

For stage 2 and 3 of PM^2 combined, the RLPM approach takes the longest to perform. The TPM and CPM approach both take nearly half of the time compared to the RLPM approach.

Iterativeness The more iterative a task is, the more effort it takes. Therefore, the iterativeness is an input measure to estimate the efficiency of an PM approach. Iterativeness or repetitiveness is a characteristic of task complexity as well. The metric is defined as the amount of iterations performed, again for each of the stages of PM^2. An overview of the performed iterations is given in fig. 6.2.

The second stage is highly iterative with the TPM and CPM approaches due to manually selecting different sets of data attributes to eventually query a complete set of event data, and to ensure that all the case identifiers were correct (due to the merging of the tickets). The CPM approach has one more iteration; to create a new table on the database. The extration with the RLPM approach is less iterative.

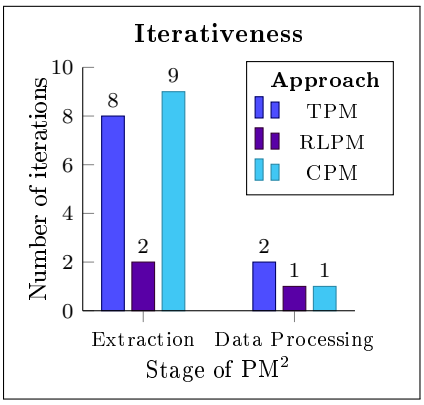


Figure 6.2: Number of iterations plotted for each PM approach.

In stage 3, no additional iterations were required for the TPM and CPM approaches. Only the RLPM approach required an iteration, namely back to the second stage of PM^2. The firstly created event data proved to be incomplete, so a complete new extraction needed to be performed.

For stage 2 and 3 combined, the TPM and CPM approach require the most iterations. This is mainly due to manual iterations to create a suitable query for the extraction of the event data.

Required Domain Knowledge This paragraph discusses the results for the required domain knowledge for stage 2 and 3 of PM^2. The domain knowledge is the information related to the case study at hand which one would require to perform a task (e.g. stage of PM^2. It determines also the complexity of a task. Domain knowledge can be visualized by the use of ontology graphs. By comparing the graphs of different PM approaches (i.e. the number of objects and relations in the graphs), we gained insights into differences in the domain information that was required for the PM approaches. Note that the business process questions also are part of the required domain information. However, since this information is equal for all three PM approaches it is omitted in the comparison. The resulting required domain knowledge is presented in fig. 6.3. More details can be found in appendix C.1, which also provides the ontology graphs.

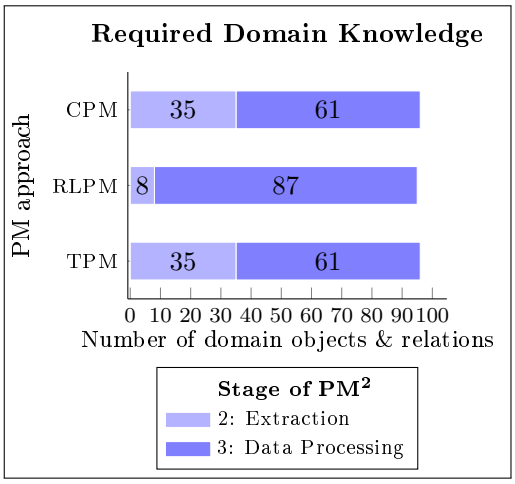


Figure 6.3: The domain knowledge of PM approaches per stage.

As is shown in fig. 6.3, the TPM and CPM approaches require the same domain knowledge for stage 2 and 3. For stage 2 that consists of 16 objects and 19 relations. These objects mostly represent tables, their connections and the attributes of interest, which are all required to build the query to extract the event data with all the information of interest. In contrast, the RLPM approach only required information about 4 objects and 4 relations for this stage. As, with PADAS only the dynamic tables (tables that are modified with inserts, updates, and deletes) of the database schema model are useful for extraction. PADAS automatically extracts the events on these tables from the archived logs, including the values of the attributes related to these events. In this case study there are (just) three of these tables. Additionally, there is the table to document the merging of tickets, which makes a total of four objects.

To assess the required domain knowledge in stage 3, only the new/additional domain knowledge is estimated. The the required domain knowledge for stage 3 is superset of the required domain knowledge for stage 2. In stage 3, the RLPM approach requires more new domain knowledge than the TPM and CPM approaches. The RLPM approach requires *additional* knowledge of 39 objects and 48 relations, where the TPM and the CPM approach require *additional* knowledge of 34 relations and 27 objects (see fig. 6.3). The RLPM approach requires more domain knowledge since a lot of mappings have to be created for it. Namely, for each attribute that represents a key, pointing to some value in an other table that is not extracted, the real actual value needs to be appointed (according to the key-instance). Moreover, the activity-label, which is actually a composition of tree other attributes (the name of the table for the event, the operation performed on the table, and the column that changed), needs to be remapped to a meaningful value. This requires domain knowledge about the process and the data model of the database.

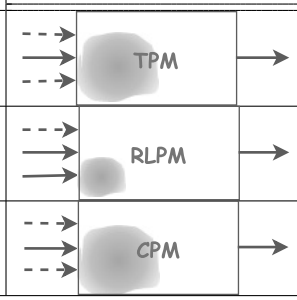
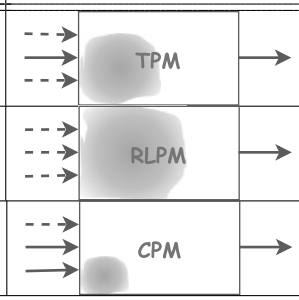
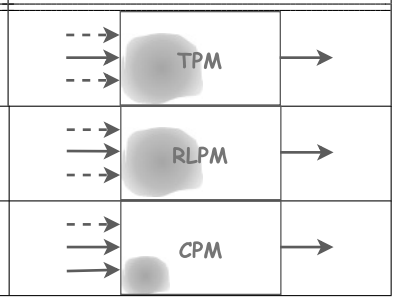
In total, all three PM approaches that are compared in this case study, require the same amount of domain knowledge to perform the extraction and data processing stages. In fact, it could be stated that, where in stage 2, the RLPM approach requires less detailed domain knowledge than TPM and CPM, this is compensated for in stage 3. Having stated this, we continue to analyze the task complexity in the next paragraph.

Task Complexity The complexity of a task determines the information needed and the cognitive effort to perform the task, which qualify as input measures for the efficiency of a task. Tasks can be categorized in five classes for complexity [31], [32]. Simple tasks are routine information-processing tasks, where the inputs, process, and outcomes can be a priori determined. Whereas complex tasks are new and genuine decision tasks, they cannot be a priori determined. The stages are categorized per approach in table 6.1.

Stage 2 was the simplest task with the RLPM approach (i.e. a normal information processing task). This is mainly because PADAS automates a large part of the extraction process; the need for (domain) information is a priori determined by knowing what tables are dynamic (fact-tables) and store information about the entity of interest (e.g. a ticket). With the other two approaches, more information is needed about relations to dimension tables.

On the contrary, the RLPM approach was the most complex in stage 3 (i.e. a known, genuine decision task). The difference in the required domain information in stage 2 is compensated for in stage 3, as the data processing (e.g. filtering) requires information of the dimension tables. Also, extra processing is needed to manipulate the event data to enable filtering on these attributes. With the TPM approach, stage 3 more simple (i.e. a normal decision task) because the event data is already equipped to be filtered. With the CPM approach stage 3 is simplest (i.e. a normal information processing task). Since Celonis offers an advanced query language which allows the user to easily slice and dice the data. The user can compose built-in functions (an a priori determined set of options) to create a view (event log) of interest.

Table 6.1: Visualization of the task complexity per stage of PM^2 and process mining approach.

Process Mining Approach	Stage 2: Extraction	Stage 3: Data Processing	Stage 4: Mining & Analysis
TPM			
RLPM			
CPM			

Additional information about how the task complexity is determined. A task has three inputs. From top to bottom these are: domain knowledge (1), process mining information (2), and problem-solving information (3). These inputs can be either determinable (visualized as a normal arrow) or indeterminable (visualized as a dashed arrow) before performing the task. Moreover, the task has a process, visualized as a black-box. If the process of the execution of the task is completely known upfront, the box will be shown as a simple rectangle. When part of the process is a priori indeterminable it is visualized with a blurry stain on it. The more blurred a task, the more indeterminable it is. Lastly, the output of a task is of importance for the categorization of a task. When the outcome (the result) of a task is not known upfront, it is visualized as a dashed arrow again. When the outcome is a priori determinable, it is again a normal arrow.

In stage 4, the TPM and RLPM approaches are considered to be equally complex (i.e. normal decision tasks). Which is logical since these approaches do not differ in this particular stage. Compared to the TPM and RLPM approach, the fourth stage is simpler with the CPM approach (i.e. normal information processing task). The reason for the latter is quite straightforward, namely that Celonis only offers a restricted set of process mining techniques; just one single mining algorithm. This makes the process, i.e. how the stage is executed, better determinable in advance. The TPM and the RLPM approaches on the other hand, offer a wide variety of discovery algorithms and many process mining techniques (e.g. conformance checking). Hence, the execution of the fourth stage has a high number of alternative paths to perform it. As a result, there are also more possible outcomes. Therefore, the execution of the fourth stage is more vague and less determinable in advance for the TPM and RLPM approaches.

6.1.2 Comparing PM approaches across all Input Measures

To summarize the observations from the previous subsection (6.1.1), we found that to perform extraction and data processing (i.e. stage 2 and 3 of PM^2) the *most time consuming* approach, but also the *least iterative*, is the redo log process mining. The Celonis process mining approach is overall (for stage 2 till 4 of PM^2) the *simplest* process mining approach. The traditional approach can be considered in between. On the one hand the TPM approach requires, on average, equal amounts of time and iterations as the Celonis process mining. And on the other hand, the TPM approach is overall equally complex as the RLPM approach.

For the *second stage*, the extraction, the RLPM approach is the simplest one; little domain knowledge is required. Compared to the RLPM approach in this stage, the other two approaches require more domain knowledge and manual iterations. The upside of the two other approaches is obviously that they require half of the time required for the RLPM approach in this stage.

In the *third stage*, data processing, all approaches require about the same amount of time. However, the TPM approach and the CPM approach are simpler to perform than the RLPM approach in this stage. Mainly because the RLPM approach requires more domain knowledge, extra steps for log manipulation, and one extra iteration in this stage.

Scenario analysis A complete overview of all the results for the input measures (per stage, per PM approach) is given in table C.1 in the appendices. The values in this table can be normalized to create a uniform scale for performance. An overview of the normalized performances is given in table 6.2; on a 0 (least performing) to 100 (best performing) scale. Accordingly, these normalized values can be used to find the approach that scores the best over all the input measures.

From the rightmost column in table 6.2, we cannot point out *one* best PM approach. Therefore, we defined three scenarios (see below), which we want to analyze in more detail. To this end, the input measures are assigned a weight-factor per scenario, and a weighted average is calculated over all the input measures per approach. This is illustrated in table 6.3.

Table 6.2: Normalized results for the input measures for all PM approaches per stage.

		Stage in PM^2				
		2	3	4		
Input measure	PM Approach	Extraction	Data Processing	Mining Analysis	Average	'Winning' approach
Time	TPM	51.3	86.5	-	**69**	TPM or CPM
	RLPM	0	88.3	-	44	
	CPM	51.3	86.1	-	**69**	
Required Domain Knowledge	TPM	59.8	29.9	-	**45**	None
	RLPM	90.8	0	-	**45**	
	CPM	59.8	29.9	-	**45**	
Number of Iterations	TPM	11.1	88.9	-	50	RLPM
	RLPM	77.8	77.8	-	**78**	
	CPM	0	88.9	-	44	
Task Complexity	TPM	60	60	60	60	CPM
	RLPM	80	40	60	60	
	CPM	60	80	80	**73**	

Table 6.3: Performances of PM approaches on efficiency input measures for different scenarios.

	Weights for input measures						
	Time	Required domain knowledge	Number of iterations	Task complexity	Process mining approach	Score	'Winning approach'
Scenario 1	1	1	1	1	TPM	56	None
					RLPM	57	
					CPM	58	
Scenario 2	3	1	1	1	TPM	**60**	TPM or CPM
					RLPM	53	
					CPM	**61**	
Scenario 3	1	1	1	3	TPM	57	CPM
					RLPM	58	
					CPM	**63**	

Scenario 1 Assigns equal weights to the performance of a PM approach on the input measures.
Scenario 2 Assesses faster approaches as *better*; the weight for the performance on time is higher.
Scenario 3 Assesses less complexity as *better*; the weight for score on task complexity is higher.

In scenario 2 in table 6.3, in which time is costly, the TPM or CPM approach would be more best suited. When, for a process mining project, little time is available, the RLPM approach is probably not the most convenient choice. Applying the TPM or CPM approach allows for a faster extraction of event data, especially when no archived logs are a priori available.

In scenario 3 in table 6.3, time is no issue for a potential process mining project, but a low task complexity is a desired dimension. An example would be a company that does not have much experience with process mining. In this scenario, the CPM approach might be preferred. Overall the stages for which the task complexity has been estimated, the simplest approach would be to use Celonis process mining. The other two are, on average, equally complex. This can be accounted to the extend to which the process to execute tasks with the CPM approach can be determined in advance in a higher degree. Also, Celonis offers less functionality with respect to process mining techniques. The other two approaches have less a priori determined processes for executing tasks, that also implies more freedom of choice for solving a problem.

An interesting observation is that overall, the same amount of domain knowledge is required for each of the PM approaches. Sooner or later, a certain level of domain knowledge is required. The only difference is that for the TPM and CPM approaches this is required from the extraction stage on, whereas for the RLPM approach, the most domain knowledge is stage 3.

Moreover, one could state that the TPM and CPM approaches versus the RLPM approach is analogous to many smaller iterations versus few bigger iterations. The TPM and CPM approaches are characterized by high iterativeness within the extraction stage. Whereas the RLPM approach requires less iterations (overall), but each iteration was much bigger (i.e. the time required for the RLPM approach was double the time required for the TPM and CPM approaches). This is justified by the observation that the actual extraction of the archived logs is a very time consuming step, where the result of a new query is just a matter of (milli)seconds.

An advantage of a different kind could be that for Celonis, no data exports are required since a connection with the database can me made. However, in the case study, we needed to create a table on the database to use Celonis properly. This requires additional privileges on the database. The TPM and RLPM approaches often involve exports of data. For the RLPM approach, additional requirements with respect to access to the database need to be taken into account (discussed in detail in section 4.1). The TPM approach requires the least privileges.

Having compared the results for the input measures of the comparison framework, we found differences in the effort required to apply a PM approach. Depending on the resources that are available for process mining project at hand, a PM approach can be determined. The next section discusses the results for the output measures, to compare the actual effectiveness of an approach.

6.2 Actual Effectiveness of Process Mining Approaches

Different approaches vary in the effectiveness with which the approach achieves the objectives of the task. The effectiveness of an approach is defined by the *quality of the result* [16]. The measurements for the output measures following from the actual application of the three PM approaches in the case study in appendix B, are discussed in this section. First, section 6.2.1 assesses the quality of the output of stage 2: Extraction. Next, section 6.2.2 discusses the differences in quality of the output of stage 3: Data Processing. Last, section 6.2.3 presents the differences found in the quality of the results of stage 4: Mining & Analysis.

6.2.1 Actual Effectiveness in Stage 2: Extraction

The *quality* of the output of stage 2, the *event data*, is assessed with eight quantitative metrics (previously defined in section 3.2). The result for each of them is discussed in this section.

Covered Period of Time The period of time for which events are logged is an important measure for the quality of event data. For example, a period that is too short could contain too little cases to perform analysis on and/or many cases could be incomplete. Fig. 6.4 shows the periods of time in which events are recorded for the PM approaches under study. For the TPM and CPM approaches this period is a consecutive 627 days, which is exactly time between the moment the PAIS went live and the moment of the extraction. The period of time covered by the RLPM approach is noticeably shorter. This is logically explained by the fact this PM project started from scratch for the RLPM approach; we started collecting archived logs only after the start of the process mining project for this research. Also the gaps in the archived logs become apparent again (see fig. 4.1 for a graph that visualizes the gaps in more detail).

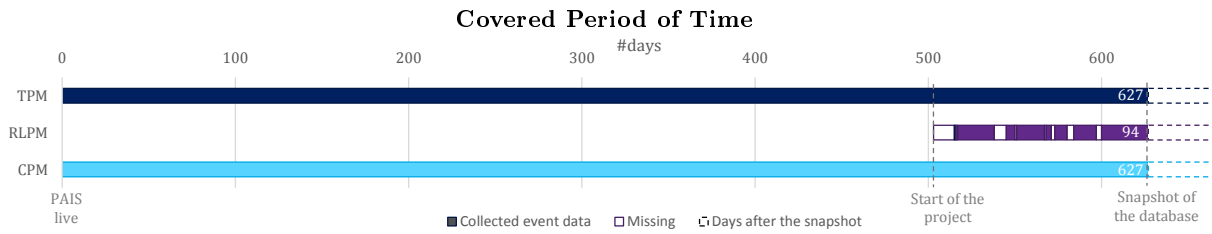


Figure 6.4: The period of time covered by each process mining approach.

Magnitude The number of events in the event data, magnitude, is an output measure to define the quality of event data. The extraction with the TPM and CPM approaches resulted in 281 906 events, and with the RLPM approach in 31 877 events. Since no archived logs were collected prior to the start of our PM project, the number of events harvested for the RLPM approach is considerably lower. Fig. 6.5 shows the magnitude per day (i.e. density of events) plotted for the TPM and CPM approaches (blue), and the RLPM approach (purple).

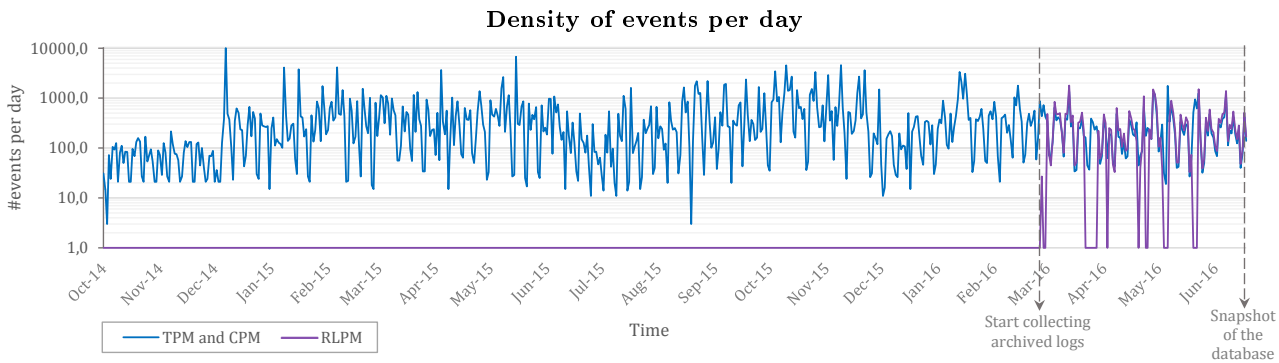


Figure 6.5: The number of events recorded per day for different process mining approaches.

The graph also shows the testing period for the PAIS from October till the end of December in 2014, where the number of events logged in this period is around 100 per day. The moment that the redo logs were archived and collected, the purple plot becomes active. As expected, the plot for the RLPM approach follows the same trend as the events collected for the TPM approach. Apart from the gaps in the archived logs, the density of the events logged for the RLPM approach is a bit higher than for the TPM and CPM approach.

Number of Activity Notions For the TPM and CPM approach, two activity notions were available: the `NAME` attribute of the `TICKET_HISTORY_TYPE` table (as is used in fig. 4.2a), and the `NAME` of the `TICKET_STATE` table. The latter was not chosen since the logic of the transitions between these states remained unclear, and the event granularity for this activity notion would be too coarse (namely, 7 distinct event classes). For the RLPM approach, two activity notions were available. The table name, type of operation, and change vector composed together in one attribute (e.g. `TICKET_HISTORY+INSERT+444444444444444`) or just the table name and the operation (e.g. `TICKET_HISTORY+INSERT`). The latter was not chosen since it resulted in a coarser event granularity.

Support The number of cases, i.e. support (SP), is the amount of independent observations of unique entities flowing trough the process. The TPM and RLPM approaches result in event data with more cases compared to the resulting event data of the RLPM approach. Again, this has to do with difference in the amount of historical data that was recorded. The total number of cases in the event data for the TPM and CPM approaches is 17 822, and the total number of cases in the event data for the RLPM approach is 3 019.

Fig. 6.6 presents the comparison of the PM approaches with respect to the number of active cases per day and the density of the events. A case is active for a particular day if one or more events take pace for that case on that day. First of all, one can point out the gaps in the graph where both dimensions for events and cases for the RLPM approach drop to zero. In some parts of the graph, the RLPM approach misses part of the events and cases, both graphs drop under the graphs of the traditional event logging. The outer-right part of the graph, after the last big gap around the 14th of July, is the 'healthy' part of the graph in which all the collected event data is complete. In this part of the graph it is shown that the density of events for the RLPM approach (i.e. RLPM #events) is higher than for the other two approaches, since the purple area is overshadowing the dark blue area behind it, except for the periods of time where there are gaps in the recording of the archived logs. It can also be observed that the number of active cases is equally or higher with the RLPM approach (in the healthy part of the graph). The higher density of events and number of active cases with the RLPM can both be explained by the fact that more events are extracted per day with the RLPM approach compared to traditional PM approach. These extra events are related to operations on other tables on the database than the `TICKET_HISTORY` table, these extra events are also listed in table 4.2.

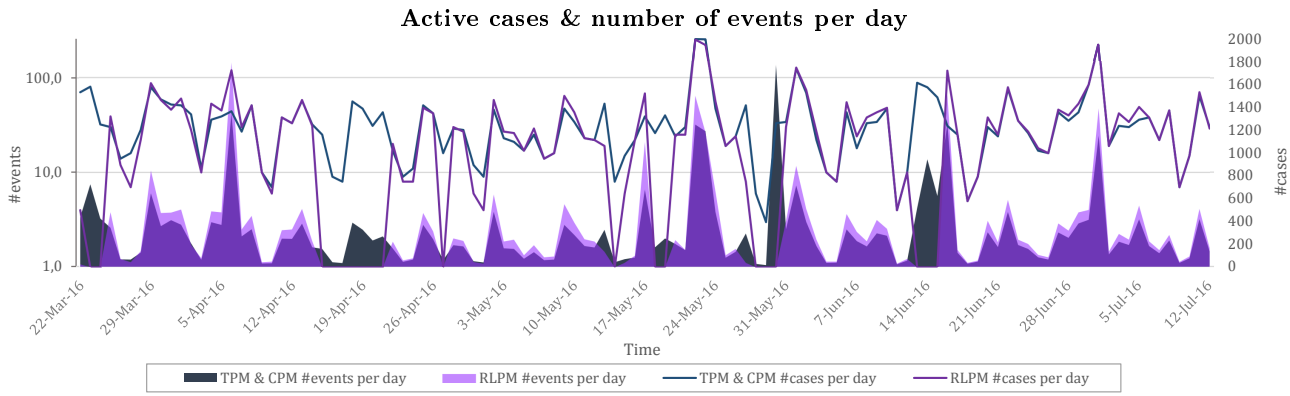


Figure 6.6: Number of active cases and number of events, per day, in the event data of each PM approach.

Number of Case Notions For the TPM and CPM approaches the chosen case notion is the `TICKET_ID`. For the RLPM approach another method was used to define the case notion. With the *trace id pattern* we defined that the individual traces should contain behavior for the same ticket instance (`PK_TICKET` of `TICKET` table as a root), the messages of this ticket (`FK_ARTICLE_TICKET_ID_ID` of `ARTICLE` table) and the `TICKET_HISTORY` instances of this ticket (`FK_TICKET_HISTORY_TICKET_ID_ID`). Thus, the number of case notions for each of the PM approaches is 1. However, if the schema model of the data base would have been more normalized, potentially more case notions could have been defined for the RLPM approach.

Granularity of Events The event granularity is defined by the number of distinct event classes (which can also be interpreted as the number of unique activities), i.e. the variety; $V(L)$. As was explained earlier in chapter 2, processes with higher amounts of activities generate more fine-grained event logs [8]. We compare the event granularity for the same process (i.e. the case study) with different approaches. The event granularity of each of the three event data sets, created with the TPM, RLPM and CPM approaches, are plotted in fig. 6.7. It can be observed that the event granularity is 31 for the event data extracted with the TPM and CPM approaches. This level of granularity is already considered to be high, since process models become quickly overwhelming and unsuitable for human perception and cognitive systems even if there are more than a few dozens of activities [42]. Fig. 6.7 indicates that event granularity of the event data of the RLPM approach is 23. However, section 4.3 showed that all the activities from the event data of the TPM approach (and CPM approach) events can be mapped to 2 activities of the event data of the RLPM approach. An inverse function could be created to map these 2 activities to all the activities in the event data of the TPM and CPM approach. Such a mapping would create a total event granularity of 52, this is indicated with the dashed bar in fig. 6.7. It can thus be stated that the RLPM approach generates finer grained event data than the TPM and CPM approaches.

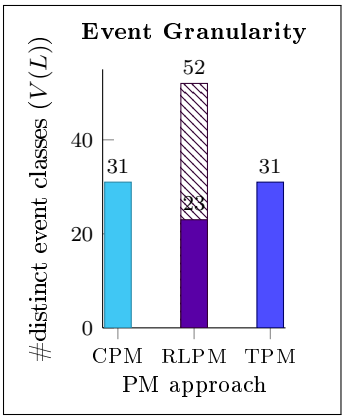


Figure 6.7: $V(L)$ of the event data of each PM approach.

Granularity of Timestamps The precision of timestamps determine the quality of the event data. The finer grained the timestamps, the more precise the logging. Fine grained timestamps are especially desired in logs where the time between two subsequent events is very small. This was also applicable to the case study as events often occurred in the same second. If timestamps are logged too coarse, the order of entries in the event data can become unreliable.

For the TPM approach, the extracted event data was logged in the order of the second. However, through a small trick, i.e. concatenating the timestamp and the last three digits of the `ID` attribute of the `TICKET_HISTORY` table, the milliseconds can be 'faked' (see also fig. 4.2a). The `ID` is the primary key of this `TICKET_HISTORY` table, thus always unique. From the configuration of this table in Oracle it is also confirmed that the primary key is always incremented by 1. This makes the granularity of the timestamp finer and thus the ordering of the events more precise.

The RLPM approach produces an XES file, which has events logged in the order of seconds. The order as events appear in the XES is used as the order of the events in the trace.

For the CPM approach the trick, performed with the TPM approach, was not necessary. Firstly, tables containing the event data with timestamps to the second were loaded in Celonis. Next, the `ID` attribute/column of the `TICKET_HISTORY` table was configured as the *sorting* table. Celonis automatically uses this sorting information to ensure the correct ordering of the events.

Total number of Unique Data Attributes The total amount of unique data attributes in an event log can indicate the 'richness' of the event data. When event data contains more information about the process, that is usually better. However, too much information can also be overwhelming for the researcher and complicate the data processing, and mining and analysis in later stages. The number of data attributes in the three event data sets created with the three PM approaches under study, are plotted in a graph in fig. 6.8. The Celonis process mining approach automatically generates two extra attributes: the number of activities and the duration for each case. Further more, it is clear that the RLPM approach produces event data with a lot more data attributes, namely 66 extra compared to the traditional process mining approach. Although this results in a lot of overhead in the event log, due to empty fields, this can also be beneficial in stage 3. Namely, when additional information is required for the data processing, the attributes containing it are all there. For the TPM and CPM approach one would need for another iteration back to the second stage, performing a new extraction with this additional information.

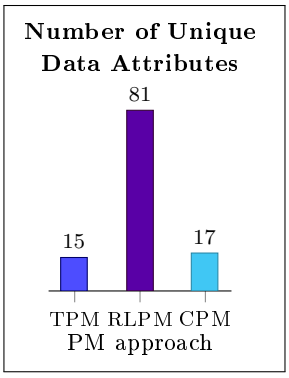


Figure 6.8: Number of unique data attributes per event data collection.

To summarize the paragraphs of this section, we can state that we have less event data for the RLPM approach (see fig. 6.4). The archived logs were not available for the same time period that the historical data was available for the TPM and CPM approaches. Logically, the number of cases and events are less in the event data created with PADAS (see fig. 6.6). Furthermore, we found for each approach, one feasible case notion, namely the ticket. Also, for each PM approach we had to choose between several options for an activity notion. Eventually we chose for each PM approach an activity notion that results in not too coarse event granularity. The event granularity and the density of events is higher with the RLPM approach compared to the other two PM approaches. The granularity of the timestamps are equally grained (i.e. seconds) for the all the PM approaches, and all three approaches have an ordering (or sorting information). Another difference can be found in the number of unique data attributes; the RLPM approach results in a very 'wide' event data set with more than 80 attributes.

6.2.2 Actual Effectiveness in Stage 3: Data Processing

To determine the effectiveness of the PM approaches for stage 3 of PM^2, we looked at the quality of the resulting *event logs*. Firstly, the performances for *structural event log metrics* will provide a quantitative assessment. Secondly, the issues for the process characteristics and the event log *quality* are illustrated. This section will then conclude with a ranking for the event logs of the different PM approaches concerning their *maturity*.

Perfomance on Structural Log Metrics

This subsection reviews the quality and quantity of each event log created with a different PM approach. Firstly, the *magnitude & support* of the event logs is compared. Secondly, the *average number of events per trace & level of detail* of the event logs is analyzed. Next, the *time granularity* for each of the PM approaches is discussed. After which the *variety*, *number unique traces*, *structure* and *affinity* of the three event logs are compared.

Magnitude & Support The magnitude (M), i.e the number of events of the event log, indicates the size of the event log. The support (SP), i.e. the number of cases of the event log, indicates the amount of unique instances that were observed. Fig. 6.9 compares the event logs created with the different PM approaches for these two output measures.

After the data processing many cases and thus events were filtered out. Firstly, the merging of tickets has an effect on all the event logs. Secondly, all the incident-type tickets were selected for all the PM approaches. And for the RLPM approach, the missing data had to be handled (see section 4.2 how part of the event log of the RLPM approach was retrieved). This is causing the resulting event log of the RLPM approach to remain with a mere 8,8% of the number of cases in the extracted event data. The event logs TPM and CPM approaches on the other hand, have around 59.1% of the support of the extracted event data.

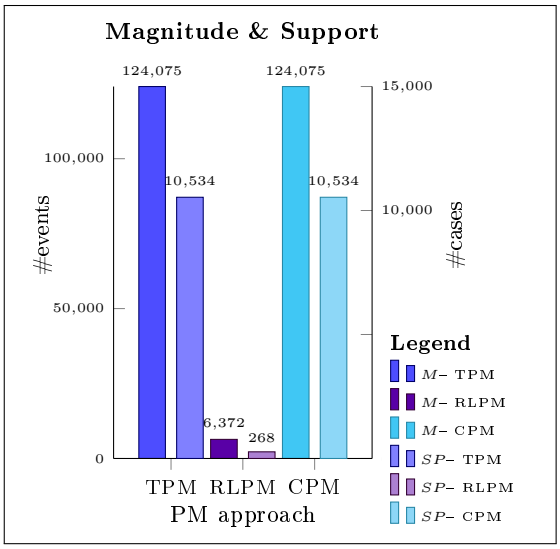


Figure 6.9: M and SP for the event log of each PM approach.

Average number of Events per Trace & Level of Detail The average number of events per trace (AE) in the event log gives an idea of the average length of the traces. The Level of Detail (LoD) of an event log represents the average number of distinct event classes per trace. The difference between these can indicate a form of repetition in the traces (without discovering any process models). Fig. 6.10 compares the event logs that have been created with the PM approaches for these two metrics. This comparison is based on the assumption that these metrics are performing equally over time, thus it is presumed that the AE and LoD are homogeneously distributed over time. The AE is more than twice as for the RLPM approach compared to the other two PM approaches. This is justified by the lower level of logging that has been witnessed in section 4.3, the higher event granularity described in section 6.2.1, and the higher event density illustrated in fig. 6.6. The LoD is of the same order, around 7 events, for all PM approaches. This implies that in the event log of the RLPM approach, one or more activities reoccur in a higher frequency than in the event logs of the other two approaches. However, if we would apply the mapping that was applied in chapter 4, the average number of events and level of detail would be higher for the RLPM approach.

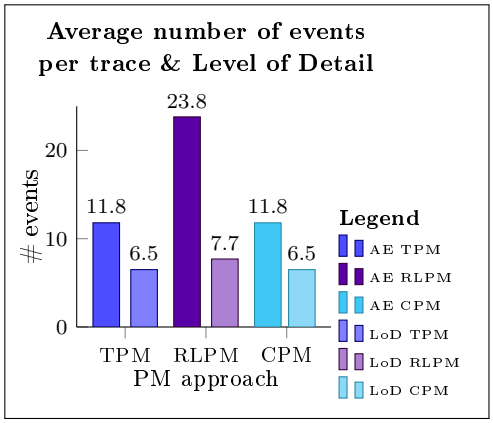


Figure 6.10: AE and LoD for the event log of each PM approach.

Time Granularity The time granularity of an event log, $G(L)$, describes the temporal raster for logging, i.e. the average minimal time between two consecutive events in a trace in an event log. This is 0 seconds for all the three event logs of the comparative study. For the TPM and CPM approaches, these events are logged on the `TICKET_HISTORY` table. Logically, for the RLPM approach, on the database level; insertions on this table take also place in the same second. That the $G(L)$ for all three approaches is equal and zero is explained by automatic logging of events by the OTRS system. For example, it was observed that the `NewTicket` activity is always followed by `CustomerUpdate` activity.

Variety The variety of an event log, $V(L)$, is the number of distinct event classes in the event log. Just like in section 6.2.1, this is an indicator for the event granularity. Fig. 6.11 compares the event logs of the different PM approaches with respect to this output measure. In the same figure, also, the minimal and the maximal variety are plotted, these are the lowest and highest amount of distinct event classes in traces of the event log. The variety of the event logs from the TPM and CPM approaches is 29, and the of the RLPM approach is 23. However, it is again important to note that the event log of the RLPM approach actually can be mapped to the activity labels of the traditional process mining approach (see also section 4.3). This would create a potential variety for the RLPM approach of 50. This again confirms the earlier observation that the events are logged on a lower level with the RLPM approach.

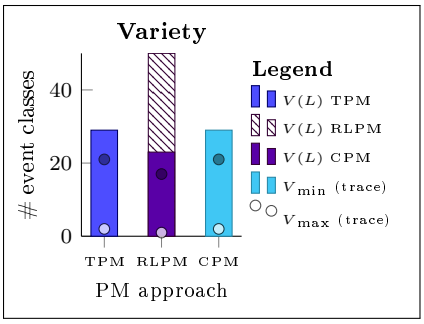


Figure 6.11: $V(L)$, V_{min} and V_{max} for the event log of each PM approach.

Unique Traces The number of unique traces in the event log (UT) is an output measure for this stage as well. E.g. an unstructured processes can produce a large set of unique traces [35]. Fig. 6.12 compares the event logs created with the different approaches based on the number of unique traces. We can observe that the number of unique traces with the RLPM approach is a lot lower than for the other two approaches. The cause for this might lay in the covered period of time that historical data is recorded for each PM approach. A longer period (for the TPM and CPM approaches) can result in a wider variety of traces. Regardless, the lower number of unique traces and the smaller variety of the RLPM event log could indicate that the RLPM approach would produce less complex process models. However, if the event log created with the RLPM approach would be mapped as stated in the previous section, the UT of the RLPM event log would increase.

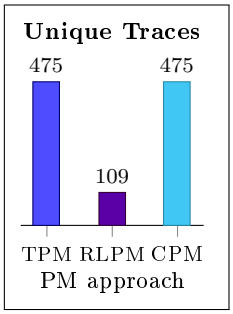


Figure 6.12: $UT(L)$ per PM approach.

Structure The structure (ST) of an event log describes the amount of observed behavior, as compared to the amount of theoretically possible behavior [35]. To be exact, the structure of an event log is the inverse relative amount of directly following relations present in the log (i.e. compared to the maximum amount of directly following relations possible). In fig. 6.13 we can compare the differences of the structure of the event logs originating from different PM approaches. None of the event logs have a very low structure, so none will be problematic for process mining algorithms. The structure of the event log created with the RLPM approach is the highest. The structure of the other two approaches could be lower due to the fact that the maximum amount of directly following relations is much higher for the TPM and CPM approaches (namely 29^{29}, instead of 23^{23} for the RLPM approach). However, the RLPM approach also extracts traces that have zero structure, since $ST_{min}$for the RLPM approach is 0, which is caused by traces that are composed by two instances of one single type of activity. The TPM and CPM have a better performance for the ST_{min}. The lowest structure found among the traces of these two PM approaches is 0.66. This indicates that these traces only exhibit part of the behavior (w.r.t. directly following relations) that is theoretically possible; which makes it possibly easier for discovery algorithms to find patterns.

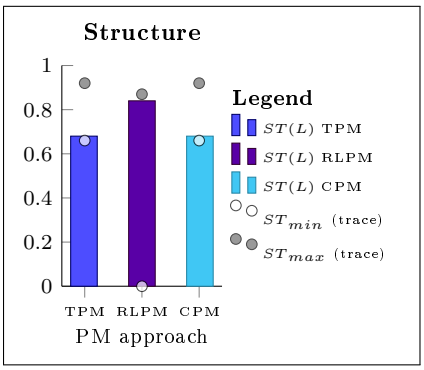


Figure 6.13: $ST(L)$, ST_{min} and ST_{max} for each PM approach.

Mean Affinity The mean affinity ($MA(L)$) of the event log represents the similarity between the traces with respect to the represented direct following relations; the homogeneity of the log. To be exact, $MA(L)$ is the relative overlap of directly following relations between all the traces in the log. Fig. 6.14 compares the different event logs for this output measure. The $MA(L)$ for the RLPM approach is clearly lower than the other two PM approaches. This means that for the RLPM less 'mainstream behavior' is can be detected, which might be caused by the smaller event log that we have for the RLPM approach. The support, being 268 cases for the RLPM approach, might not be enough to make a distinction between frequently occurring behavior (thus overlap), and incidental behavior.

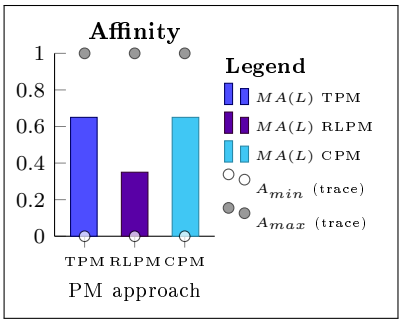


Figure 6.14: $MA(L)$, A_{min} and A_{max} for each PM approach.

The results from previously discussed metrics depend on the process at hand. Depending on how these metrics manifest themselves in the event log, we can also find potential challenges. These challenges, such as dealing with fine-granular events, case heterogeneity, voluminous data, and concept drift can be discovered without applying any further analysis with process mining algorithms. Apart from that, we have just illustrated that clear differences in these metrics can also arise, for the same process, depending on the approach that is chosen for the PM project.

It can be stated that, after the data processing stage, the event logs created with the TPM and CPM approaches are 'larger'. The direct reason for this has already been discussed in section 6.2.1; the period of time for which archived logs has been 20% of the period of time that events have been logged in the traditional manner. However, the number of cases that in the event log of the RLPM approach is only a mere 2,5% of the number of cases in the event logs of the CPM and TPM approaches. This unproportional difference is explained by the missing archived logs. Even if one event of a case was missing, we had to remove all the other events of this case.

Another observation is that the RLPM event log has, on average, twice as much events per trace than the other two event logs. At the same time, this event log created with the RLPM approach has on average not a lot more distinct event classes per trace (7,7 compared to 6,5 for the other two approaches). Thus the RLPM event log seems to have a higher amount of repetition of activities in the traces of the RLPM approach. Moreover, the RLPM event log has a lower variety and contains less unique traces; four times less. Altogether, this would imply that the RLPM event log is more homogeneous, which is confirmed by the higher score on structure. However, looking at the overlap in the directly following relations between traces, the mean affinity, for the RLPM approach is lower than for the other two approaches. This indicates that the procedures on the database vary more than the higher level procedures logged for the TPM approach.

Event Log Quality issues

Several classes of data quality issues can be manifested in a log [8]. These issues can frustrate the applicability of several process mining techniques. The PM approaches can be compared by identifying which classes of issues are found for the created event logs in the data processing stage. In table 6.4 the event logs created with the different PM approaches are evaluated for event log quality issues. Also, a description of each of the quality issues is given below.

Missing Events – R – The archived logs used in this process mining project had gaps; missing data for periods of days.

Missing Case Attributes – R – With PADAS all the data attributes from the tables of the database become event attributes in the event log. Different tables in the database have different sets of attributes, but all the events have the attributes of all the tables that are extracted. Thus each event has missing values for the attributes that belong to other tables. E.g., a ticket has a type; this attribute could be of value 'incident'. Now, messages belonging

to this ticket do not have this attribute, so the value of the type attribute is given as empty. This creates hurdles for filtering and data processing tasks.

Incorrect belongs_to – T,R,C – The data of the tickets that are linked is provided in the database. We observed several tickets linking to other tickets, and so on, where finally the last ticket again is linked to the first one. Based on the context information we know that the merging should behave like a directed acyclic graph, which is clearly not the case. This leads to incorrect data of which event actually belongs to which ticket.

Imprecise Case Attribute – T,C – For 6780 cases the trace attribute 'type' of the tickets are defined as `Unclassified`. For the RLPM approach, this was not an issue, since this issue does not hold for traces that were active in the period of archiving the redo logs.

Imprecise Activity Names – T,R,C – For the TPM and CPM approaches, some activity names are too much on the system level (e.g. `StateUpdate, Lock` and `Unlock`), and other activities are too coarse (e.g. `Misc`, which could mean anything). The activity labels from PADAS were also coarse; a distinction was made between `New event & message` and `New event (without message)` for all the insertions on the `TICKET_HISTORY` table. These activity labels require a remapping as was illustrated in section 4.3.

Imprecise Timestamps – T,R,C – The timestamps mined with all PM approaches were too coarse for the time granularity of the process under study. Timestamps with a precision of seconds are undesired if the event log has a time granularity of zero seconds or finer. The issue was resolved for each of the approaches by using the ordering (see section 6.2.1).

Irrelevant Cases – T,R,C – Many tickets in the event logs are created automatically by a monitoring tool. This gets triggered as soon as, for example, disc space is running out. This is an enterprise-class open source distributed monitoring solution called Zabbix. The tickets that are generated this way are considered irrelevant for the process that we want to analyze.

Table 6.4: Evaluation of event log issues for the event logs of each PM approach.

	case	event	belongs_to	c_attribute	ordering	activity name	timestamp	resource	e_attribute
missing		{R}		{R}					
incorrect			{T,R,C}						
imprecise	–	–		{T,C}		{T,R,C}	{T,R,C}		
irrelevant	{T,R,C}			–	–	–	–	–	–

The cells with "–" indicate that such a data quality issue is not possible. T, R and C respectively indicate the event logs created with the TPM approach, RLPM approach and CPM approach.

Having stated the above, we can observe that only the RLPM approach is affected by issues concerning missing data. The issue with respect to the imprecise case attribute in table 6.4 is only an issue for the TPM and CPM approaches, but this could also have been an issue for the RLPM approach if we had more historical data available for this approach (i.e. older redo logs). All the other issues are affecting all the process mining approaches. Thus not one approach can be appointed as a best PM approach with respect to the quality of the event logs that can be derived.

Maturity of the Event Logs

This section assesses the resulting output of the data processing stage for each PM approach in a more qualitative manner. This is done through ranking each of the event logs for their maturity according to the rankings of the process mining manifesto [2].

Traditional- and Celonis process mining: ⋆⋆⋆ Although the events need to be extracted from a variety of tables from the DBMS, events are recorded automatically, and mostly, in a reliable manner. There are some tables not filled in the DBMS, thus causing incomplete logging. The ticket-links for merging cases are sometimes unreliable, since cycles were discovered (e.g. one ticket linked to a second ticket, that is again linked to the first ticket again).

Also, logging is done per second, thus a lot of events happen at the same time. This problem is circumvented by using the primary key of this table (see section 6.2.1). For the TPM and CPM approach, also a notion for a process instance is supported in an explicit manner (i.e. a case is ticket). However, an activity notion is not supported in an explicit manner. For the extraction we chose the `ticket_history_type` attribute (see section 6.2.1), where we also could have chosen the `state` attribute of a ticket. Moreover, messages (i.e. instances of the `ARTICLE` table) could also be used to define activities. Altogether, this results in a three star ranking for maturity of the event logs created with the TPM and CPM approaches.

Redo log process mining: ⋆⋆⋆ The events in the log can be considered to be trustworthy, since the redo log writer of Oracle is a structured and automatic process that is very concise on a fine-grained low level. However, in this study, we had to encounter missing data (full days), thus the event log was incomplete. After the application of the Redo Log Process Mining approach in the case study, an automated process had been developed to secure the completeness of the data. So if a new process mining project would be executed within the same organization, this would most likely result in complete archived logs. For that scenario, the event log could be ranked with a higher maturity. The case notion and activity notion, however, are not pre-determined in any RLPM implementation. They always need to be configured manually in PADAS. Therefore, any RLPM event log will not rank in the four star maturity.

Thus, although not all the process mining approaches use the same source for their event data, the maturities of the event logs all rank ⋆⋆⋆. This concludes the comparison of the PM approaches on their actual effectiveness in stage 3 of PM^2. The next section elaborates on the actual effectiveness in stage 4.

6.2.3 Actual Effectiveness in stage 4: Mining & Analysis

Since the redo log process mining approach follows the same method as the traditional process mining approach for the fourth stage, only a comparison for TPM and CPM approach will be relevant here. To determine the effectiveness of the PM approaches for stage 4 of PM^2, we performed an assessment on a higher level than for the previous two stages. As was stated in section 3.4, the comparison is based on the type of models that result from the mining (i.e. semantically executable models or not), on the process discovery techniques that are supported, whether conformance checking is possible, the process enhancement techniques available, and visual analytics.

Executable semantics The TPM approach, using academically oriented tools, results often in process models with executable semantics. The Fuzzy miner is one of the few in the traditional process mining approach that results in a directly follows-graph. Other miners like the Inductive Visual Miner, Inductive Miner - infrequent and the ILP miner all result in executable process models. With Celonis process mining we derived directly follows-graphs. These process maps could be animated (at the speed of the users choice even), allowed for extensive log filtering, and were useful for global analysis. However, these process maps have no executable semantics. This is considered to be a fundamental difference between the TPM and CPM approach.

Supported Process Discovery Techniques With the traditional process mining approach we can choose from *a wide variety* of discovery algorithms. ProM has numerous plugins that support the discovery of transition systems, Petri Nets, or Event Process Chains (EPCs). A set of techniques that implemented in ProM are: α, α^+, tsinghua-α, α^{++}, $\alpha^{\#}$, α^*, the ILP miner, the Heuristics Miner, the Genetic Algorithm Miner, the Duplicates Genetic Algorithm Miner, Language-based Region algorithms, the Fuzzy Miner, Evolutionary Tree Miner, and State Discovery Algorithms. For the process mining project under study we applied the Inductive Visual Miner (fig. 6.16), Multi-phase miner, Social Network Miner (fig. B.18 in appendix appendix B.4.1), and the Process Comparator (e.g. fig. 6.15).

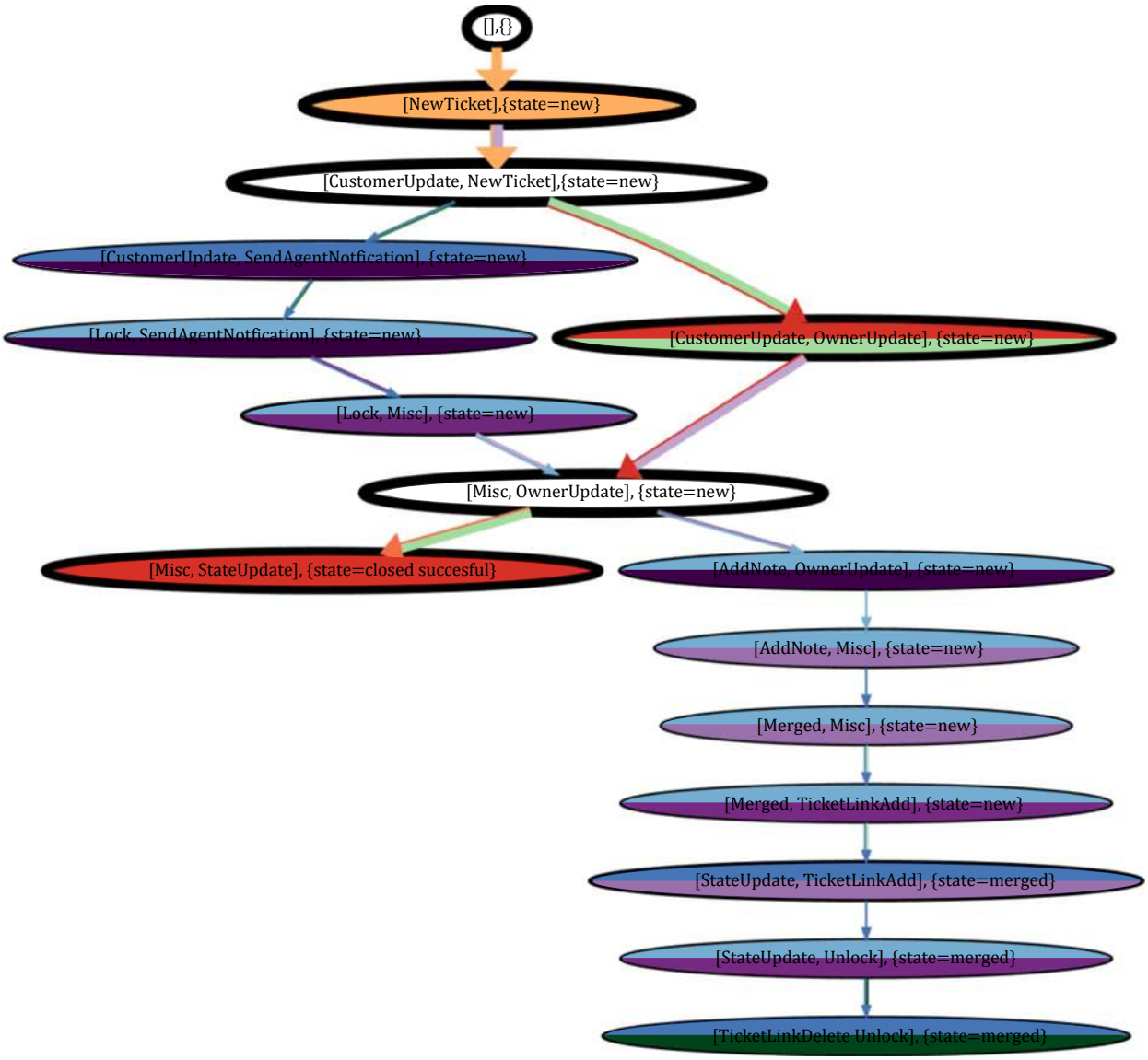


Figure 6.15: Process Comparator, comparing workflows of two event logs of the same process. One event log with tickets concerning customer 1 as group L_1 and one event log concerning customer 2 as group L_2.

For the Celonis process mining approach, there is exactly *one* algorithm that can be used. The *Process Explorer* (for an example, see fig. 6.17) and *Variant Explorer* are components that can be created in Celonis to create process maps, but they both rely on the same algorithm. The second one is an extension of the first one, that can be used to specifically look at one or more trace variants, thus actually this is again a slicing component to create very specific selections. These process maps turned out to be merely useful to give an impression of the workflow, using the sliders to change the amount of edges and nodes (based on their frequencies), allow the user to generally explore the order of activities in a process.

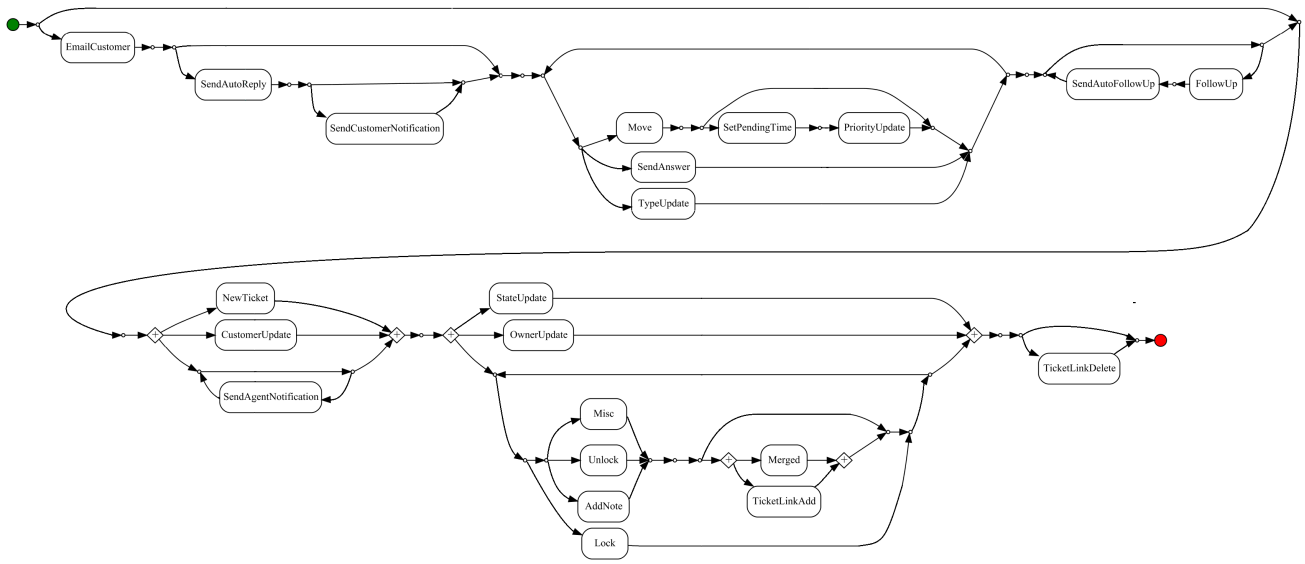


Figure 6.16: Model created with Inductive Visual Miner (96,0% of the activities and 80,2% of the paths).

Supported Conformance Checking Techniques Techniques that are used for conformance checking (finding deviations by replaying a log on a model) are completely supported by the traditional process mining approach. For the CPM approach, this is not the case. No log can be replayed on the discovered process maps (models) in Celonis. Therefore, no conformance checking can be performed explicitly with the Celonis process mining approach.

Performance Enhancement Techniques With traditional process mining, once a process model is created, all kinds of analysis techniques based on "replay" come in to reach. For example, flow times, waiting times, service times, synchronization times, etc. can be derived from the timestamps. Also, information concerning the resources, customers, organizational units, and cost rates could be analyzed. A resulting process model can be annotated with this information. Traditional process mining techniques entail a wide variety of these techniques. Bottleneck, frequencies, service levels, and other information that is contained in the log can be shown. Also, already mined models can be *repaired* by conformance checking.

Celonis has built in some functions that can be used to annotate the process maps. These are standard: Median throughput time (in days, minutes or seconds), Average throughput time (in days, minutes or seconds), the case frequency and activity frequency. More annotations can be created by the user as a specific KPI.

Visual Analytics It must be noted that specific visual analytics that are available, and very relevant for temporal data, are not possible (yet) in Celonis Process Mining. Creating a dotted chart is not possible, where this has been possible in ProM and RapidMiner for some years already. A scatter plot can be created to get close, however, no dimension can be mapped to the colors of the dots. This would have to be defined with a manually created mapping. An then still, the amount of data points that can be plotted is limited. This causes the plot to be restricted to only a certain time frame for plotting. On the other hand, a lot of other options for visual analytics are offered by Celonis. The options, only for creating graphs directly from the event data are; OLAP table, Column Chart, Pie Chart, Donut Chart, Line Chart, Area Chart, Scatter Plot, Bubble Plot, Histogram Chart, Single KPI. To illustrate this range of options, some additional analysis has been performed. Please refer to fig. B.30 to get an impression.

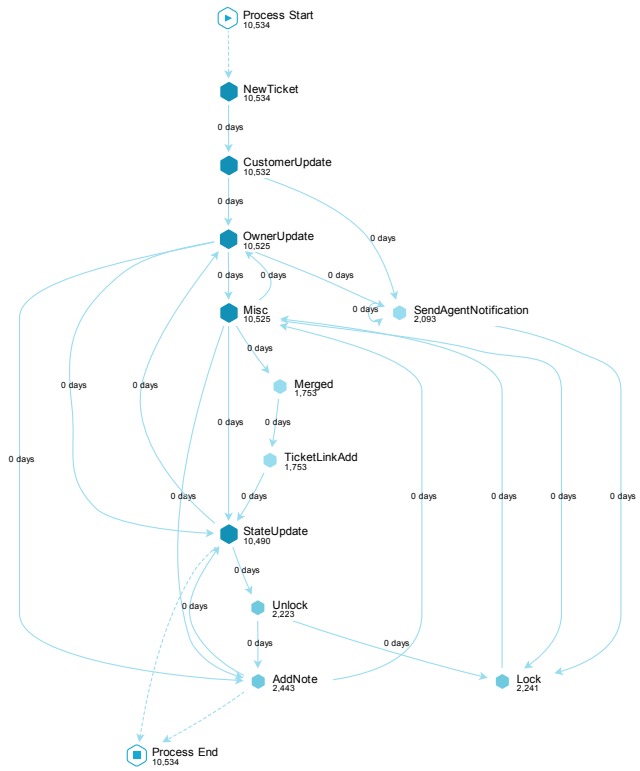


Figure 6.17: Process model created with Celonis (96% of the activities and 80,3% of the connections).

The *Process Explorer* and *Variant Explorer* need to be addressed with extra care; the user should be extra aware that these representations of the process are highly filtered. As soon as an activity is filtered from the process model with the slider for frequency, the model removes all the traces with this activity, which may lead to misinterpretations. The user should also be aware that the numbers on the edges in the process map, if configured for case frequency, are the number of directly following relations from the origin of that edge to the target activity.

To summarize, the quality of the resulting process models created in the mining & analysis stage are better with the traditional process mining approach (and thus also for the RLPM approach). For the TPM approach many mining algorithms have been developed and validated to discover process models, to assess the fitness, precision, generalization and simplicity. Contrastingly, the Celonis process mining approach is limited to óne technique. With this technique, the user will be able to get a general idea of the work-flow, an 'impression' of the complexity and ordering. For a beginning process miner, this might initially be enough, but for more advanced process analytics this might not be sufficient. Additionally, the quality of the discovered process maps is considered to be higher when they are semantically executable. The executable models we get with traditional process mining can be mapped onto a transition system, and thus easily be translated to higher-level languages such as Petri Nets, BPMN, EPCs, and UML diagrams. Moreover, the visual analytics with the traditional approach are better in the sense that one can create a dotted chart and the process models are less sensitive for misinterpretations. The CPM approach on the other hand offers an own type of query language in which functions can be applied to measure for example the average or median throughput times, solve times, response times, number of activities/cases per time-unit, creation dates, last dates, and more. This allows the user to make a dashboard in which the desired calculations are made for any selection of the event log.

This concludes the assessment of the PM approaches on their actual effectiveness in stage 4 of PM^2. The next section illustrates how the PM approaches performed on the perceived efficacy.

6.3 Perceived Efficacy of Process Mining Approaches

This section completes the final part of of the comparison framework. Namely, by comparing the perceived ease of use and the perceived usefulness, we determine the process mining approach with the best perceived efficacy. The *perceived ease of use* (PeoU), earlier discussed in section 3.1, refers to the degree to which a person believes that using a particular system would be free of effort [17]. And the *perceived usefulness* (PU) is defined as the degree to which a person believes that using a particular system would enhance his or her job performance [17]. As was stated earlier, jointly these concepts determine the *intention to use*, which in its turn determines the *actual usage*. Process mining approaches will be adopted based on perceptions of their ease of use and usefulness [16]. Since one of the dimensions that can be considered to evaluate the "success" of a method is *the adoption in practice*, which is equivalent to the actual usage, it is relevant to evaluate the indirect determinants (i.e PeoU and PU).

Following Moody's model for evaluating a method, the researcher assessed all the process mining approaches with respect to the PeoU and PU. In this case study, the assessment is carried out only once, and only by one person. Meaning that the findings from this assessment are subjective. However, the findings provide a taste of the perceived efficacy and they are part of the comparison framework. As stated in section 3.1, in order to do this, a validated set of measurement scales from earlier work [17] applied. All the scores for PeoU and PU, plus the average performances (calculated with an equal weight for each proposition) is presented in fig. 6.18. An detailed overview of all the propositions and how they are assessed is provided in appendix C.4.

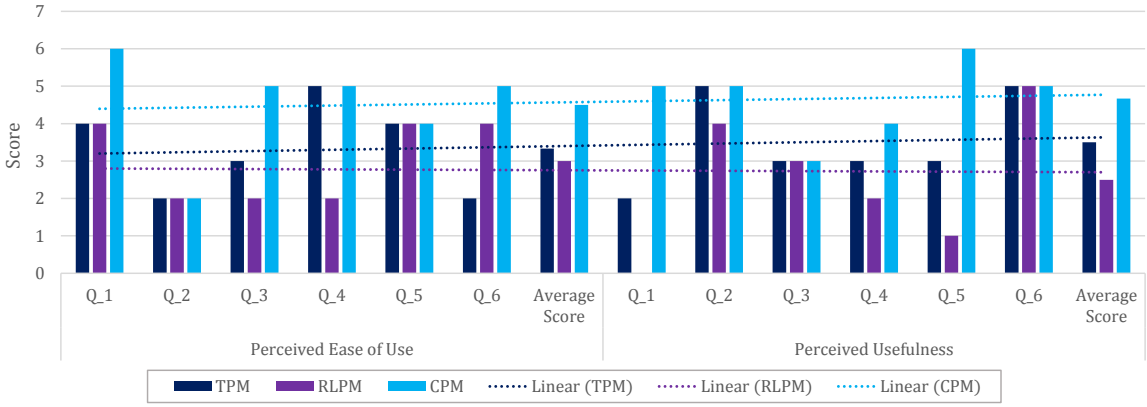


Figure 6.18: The perceived efficacy of the process mining approaces, measured by the perceived ease of use and perceived usefulness scored on a seven-point Likert scale.

Looking at the results in fig. 6.18, it can be stated that the perceived ease of use is scoring the best for the CPM approach, followed by the TPM approach, an right behind it the RLPM approach. CPM was perceived easier to learn, the interaction with Celonis was perceived more clear and flexible, and easier to use. Compared to the TPM approach, the RLPM approach was considered to be less flexible to interact with, and less clear and understandable. However, the RLPM approach was perceived slightly easy to use.

On the perceived usefulness the CPM approach scored best, followed by the TPM approach, and then by the RLPM approach. The CPM approach was perceived to accomplish tasks more quickly, and makes it easier to do a process mining project. Compared to the TPM approach, the RLPM approach was perceived not likely to accomplish tasks more quickly. Also, the RLPM approach would make the process mining project less easy to perform.

This section discussed the last dimension of the comparison framework for process mining approaches, namely the perceived efficacy. The next sections provide more high-level insights that are gained during the comparative study. Section 6.4 explains the implications of the style of the schema model of a database. Section 6.5 performs a cross comparison by taking into account all the low level metrics of all the stages, and aims to answer the main research question of this thesis.

6.4 Implications of the Schema Model Style of the Database

The comparative study has led to the insight that in real-life situations the schema model of the database should be studied before choosing a PM approach. The type of schema model affects the effort required in stage 2 (extraction) for the CPM and RLPM approaches, and for stage 3 (data processing) for the RLPM approach. This section first introduces the concept. After which two major styles for schema models, and how they differ, are described. Last, this section explains why and how the schema model style has an impact on the PM approaches.

The schema model, created in the logical design, of the DB is the collection of DB objects, including the tables and views. The schema objects are arranged in schema models that are designed for data warehousing. Schema models are designed for different purposes depending on the performance requirements, and required accuracy for inserting, updating and deleting data.

Two major schema model styles found in practice are the *Third Normal Form schema* (3NF) and the *star schema*, see figure 6.19. The 3NF design seeks to minimize data redundancy. This is achieved by *normalization*, a data design process that aims to keep each fact in just one place to avoid data redundancy while ensuring data integrity. This helps to prevent insertion, deletion and update errors, because the data is available in just one attribute of one table in the DB. The star schema (often this refers to a "dimensional modeling" approach), aims for structural simplicity and high performance data retrieval. Since many queries are generated by reporting tools and applications, making the query generation convenient and reliable for those tools and applications is vital. Star schemas divide data into facts and dimensions. Facts are the measurements of some event, such as a sale, a ticket, or ticket-event. Dimensions are the categories used to identify these facts, such as date, customer, or product. Where the 3NF schema tries to minimize it, the star schema accepts data redundancy (denormalization) in its dimension tables for the sake of easy user understanding and better data retrieval performance. This case study involves a star schema.

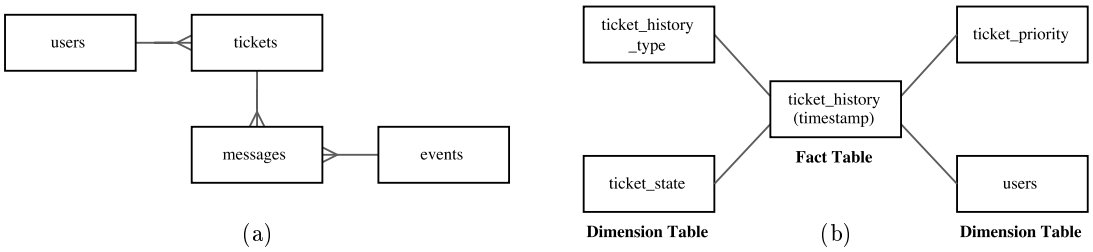


Figure 6.19: A fragment of a Third Normal Form schema in a and a fragment of a star schema in b.

The type of schema model might not necessarily influence the efficacy of the traditional process mining approach. A query will still have joins and involve multiple tables.

A normalized schema model makes life easier for the CPM approach, since no queries would have to be written to create new tables on the database. And thus, less permissions would be required and the extraction stage could be performed with less effort. The tables of interest can just be loaded into Celonis, the data model can be defined with two clicks per relation.

Also the RLPM approach is best suited for a normalized schema model. Because then tables are representing actual objects; where in each of the tables, operations will be executed on the database. This leads to an easier interpretation of the extracted activities (which are the insertions, updates and deletions on the DB). With a star schema, the extraction with PADAS is more straightforward (and thus less flexible), since only the fact-table(s) are convenient for extraction. Usually, the extraction will then involve less tables (only the fact-tables), and thus cost less time. This 'comfort' is accompanied with the disadvantage that more effort is required in the next stage, data processing. All the key-values that the fact-tables are composed of, need to be mapped to actual understandable values; which can become a quite labor intensive job as soon as the variety of values increases in complex cases (e.g. for a web-shop selling more than 1000 products, listed in a dimension table, related to a fact-table for sales with an identifier for the products).

To conclude, we can state that a normalized data model is beneficial to both the RLPM and CPM approaches. In spite of that the extraction with the RLPM approach will take probably take more time. Star schemas may increase the efforts for these two PM approaches.

6.5 Comparing the PM approaches Across Dimensions

The previous sections of this chapter discussed and compared the results on metric level and stage level. Next, this section aims to answer the main research question from section 1.3, namely, *"Which process mining approach is best suited for this real-life process mining project"*. To that end, this section provides a cross dimensional overview on a higher point of view.

Starting with table 6.5, we present the overall performance of each of the three PM approaches. In this table the scores (i.e. performances) are normalized from 0 (i.e., least performance), to 100 (i.e., best performance), measured from stage 2 to 4 of PM^2. The results in table 6.5 are based on the assumption that each metric is equally important, and each stage is equally important. Thus, the metrics and stages are equally weighted. As this thesis focuses more on the objective metrics for the comparison of process mining approaches, the remainder of this section first discusses the actual efficacy. Afterwards, the cross dimensional comparison is made, which also takes into account the subjective performance of PM approaches. Note that this perceived efficacy is a subjective dimension since it is based upon one point of view (i.e. of the researcher).

Table 6.5: Assessment of PM approaches across multiple dimensions. Scores are normalized to percentages.

	Actual Efficacy			Perceived Efficacy			
	Actual Efficiency	Actual Effectiveness	Average Actual Efficacy	PEoU	PU	Average Perceived Efficacy	**Cross Performance**
TPM	56.00	79.93	67.97	47.62	50.00	48.81	**58.39**
RLPM	56.75	61.91	59.33	42.86	35.71	39.29	**49.31**
CPM	57.75	49.10	53.42	64.29	66.67	65.48	**59.45**

Firstly, in the column *'Actual Efficiency'* we present the average performance of the PM approaches for the actual efficiency. These figures were presented earlier in table 6.3. In this scenario, where all the input measures are equally weighted and each stage is considered to be of equal importance, all the process mining approaches perform approximately equally. This means that, overall, all PM approaches require the same amount of effort to execute.

Secondly, the column *'Actual Effectiveness'* provides an average score on the resulting quality and quantity of the outputs (e.g., the event data, the event logs and the analysis outcomes) of the PM approaches. All the individual performances on the low level metrics were averaged firstly per stage to determine the actual effectiveness of the PM approaches on stage level. A more detailed overview of the normalized scores of the PM approaches on output measures at stage level is provided in appendix C.3. Accordingly, we calculated the mean of the average performances of over all the stages. These final scores indicate the following.

- The TPM approach is performing the best, with a score of almost 80. Which means that the output of this PM approach has the highest quality and quantity.
- Second-best would be the RLPM approach with 62. Which is explained by the delimitation that before the start of this research, the redo logs were not collected. Moreover, this is due to missing redo logs. The resulting event data lower in quantity. Also, the RLPM approach resulted in an inconveniently 'wide' event log with more than 80 attributes and empty fields.
- Worst performing, is the CPM approach with a score of 49. This is due to the poor process mining results (e.g. little freedom of choice in discovery techniques, no executable semantics, no explicit conformance checking possible).

Thirdly, in the column *'Average Actual Efficacy'*, is the average of the aforementioned two columns. Here, the same observation can be made as in the previous paragraph, namely that the TPM approach is the best performing and the CPM approach is the least performing. However, the difference between the scores has become lower. From an objective perspective, we can state that the efficacy of the TPM approach is notably higher than the CPM approach. The RLPM approach is performing in between, due to the lesser quantity of the results (i.e., the smaller event logs). However, this is explained by the fact that prior to the start of this project, no archived logs are available.

Finally, the last column *'Cross Performance'* takes into account all the aforementioned columns, plus the results for the perceived efficacy (which was discussed in section 6.3). More details about the calculation of the normalized scores for the perceived ease of use and perceived usefulness are given in appendix C.4.1. We observe that the CPM approach compensates its aforementioned lower scores by a high perceived ease of use and perceived usefulness. Furthermore, we found that the RLPM approach is a little more complex and time consuming to perform, thus less easy to use.

As we now have performed a cross dimensional comparison over multiple stages and metrics, this paragraph lists a number of additional, more detailed, findings about each PM approach.

The **traditional** process mining approach aims at getting sound results. Namely, one can mine process models that are semantically executable in a science-driven formal languages (e.g., Petri nets), perform conformance checking, and performance enhancement. Also, this approach allows the user to apply a wide variety of process discovery techniques, which makes the task a little more complex when comparing it to the CPM approach. E.g. a custom model in RapidMiner and a custom Python script were required to calculate the solve and response time. Perhaps, one could also label the TPM approach as "the manual" approach.

The **redo log** process mining approach resembles the goals of the TPM approach. Every event in the TPM event data is also extracted with the RLPM approach. With the RLPM approach, one can even extract more. The TPM event data set is a subset of the RLPM event data set. However, a considerable amount of archived logs were missing. Section 4.2 discussed this issue. The RLPM approach is the simplest to perform in the second stage of PM^2 since a lot of steps are automatically performed by PADAS. However, the extraction was time consuming. Contrarily, the RLPM approach was the most complex PM approach in the third stage due to extra data processing activities. E.g., creating mappings and log manipulation for filtering purposes since PADAS creates event logs solely with event-attributes and no trace attributes.

Celonis offers overall the simplest approach, and scores slightly higher on the perceived ease of use and perceived usefulness. The process maps resulting from the CPM approach are industry-driven and have no executable semantics. Celonis clearly targets business oriented users. Celonis offers *one* process discovery technique and no conformance checking. To calculate the solve and response times, functions were created using the formula builder provided by Celonis.

Lastly, below we list a few remarkable similarities between the PM approaches that have been found with the comparison framework.

- All PM approaches produced event logs with level $\star\star\star$ for maturity. Even though the TPM and CPM approaches use the data in the tables at the RDBMS, which are logged by a PAIS located at the application level. And the RLPM approach extracts events from the redo logs DBMS and a snapshot of the database (at the start or at the end of the set of redo logs).
- This research has also shown that regardless of the chosen PM approach, eventually the same amount of domain knowledge about the process and system at hand is required.

Concluding, we state the context of a process mining project, determines which PM approach is best applicable. Which depends on, the time available for the PM project, the structure of the database at hand (which was illustrated in section 6.4), the existing knowledge and experience with the process mining domain, and the desired output for the analysis. For example, in a process mining project where time would be very limited, the TPM and CPM approaches are better suited as they are faster to apply (as was explained in section 6.1.2). In another situation, where a process mining team is inexperienced in the process mining domain, the CPM approach would be a better advise as it is the simplest and easiest approach to use. But it also depends on the kind of analysis that one would like to perform. Since not all process mining techniques are offered by Celonis, the TPM approach could be a better solution in some situations.

Theoretically, assuming that redo logs are available, with the RLPM approach the same goals can be achieved as with the TPM approach, but not the other way around. Moreover, with the TPM approach, the same goals can be achieved as with the CPM approach, but not the other way around. Having stated the above, we stress that depending on the process mining project, one should decide which process mining approach is most suitable.

Chapter 7

Conclusion

7.1 Summarizing the Results

Within this master thesis, we have defined and applied a comparison framework to compare process mining approaches. With this comparison framework, we analyzed the traditional process mining approach (1), the redo log process mining approach (2), and the Celonis process mining approach (3). We recall the research question of this work and the research goals as defined in section 1.3.

Research Question. *Which process mining approach is best suited for a real-life process mining project?*

Before defining a comparison framework, we have explored earlier work related to method evaluation, methods for the process mining process itself and the three chosen process mining approaches. Findings are that Moody's method evaluation model is applicable to compare different methods for more concrete tasks only. And, PM^2 was recognized to clearly define stages and the thereto belonging activities in a real-life process mining project.

Our first contribution to existing work is the definition of the comparison framework for process mining approaches, which completes RG. 1. The comparison framework we propose aims for an objective comparison of complete process mining approaches in an structured manner. The two cornerstones for the framework are: PM^2 for scoping the tasks that need to be performed in a process mining project, and Moody's method evaluation model to determine how successful each task is executed with a particular approach. By having defined different sets of input and output measures, tailored to each stage of the process mining project, the user is enabled to make a comparison in every stage of the process mining project.

The second contribution of this master thesis is the application and verification of the redo log process mining approach for the first time in a real-life scenario, which completes RG. 2. This RLPM approach systematically uses database redo logs as a new source of event data [3]. To make sense of the events and obtain logs, the concept of the trace id pattern was applied using PADAS. The verification was performed through comparing the extracted event log with a second event log extracted with the traditional process mining approach, which confirmed that these two event logs were equivalent. We even found, for this case study, that the event data extracted with the RLPM approach is a superset of the event data extracted with the traditional approach. This establishes that one can extract event logs with a fine level of granularity, even finer than the traditional approach in this case study. Thus, it has been validated that transitive relations between data objects can be discovered, just like the causal dependencies of data modifications.

To verify the comparison framework, we used it to compare three process mining approaches in a real-life case study and thus completed RG. 3. The completion of every step was documented in appendix B and monitored to measure the input and output measures depicted in the comparison framework. Due to external causes, the archived logs for the RLPM approach were incomplete. Although this was no issue to verify the approach, it did not result in a sufficiently large event

log for the execution of mining & analysis activities. However, since the RLPM approach proved valid (i.e., can extract an event log equivalent to the TPM approach), and because the fourth mining & analysis stage of PM^2 with the RLPM approach is executed with the same tools and techniques as the TPM approach, we claim that the TPM approach also resembles the execution of the RLPM approach in the fourth stage. All the discovered findings have been reported in chapter 6, completing RG. 4.

Lastly, in section 6.5, we aimed to answer the main research question by performing a comparison across dimensions. This comparison takes into account all the input and output measures, over all the stages, and the perceived ease of use and perceived usefulness. This resulted in a high-level comparison of the average performance of each process mining approach on the actual and perceived efficacy. We found that, in this case study, the TPM approach scored best across all these dimensions. However, this is in a scenario in which each metric and each stage is weighted equally. We stress that the best suited process mining approach can only be determined depending on the process mining project at hand. See section 6.5 for the detailed conclusions resulting from this cross comparison.

The next section reflects upon the comparison framework. Accordingly, section 7.2 provides recommendations for future research. Last, in section 7.3 we provide an overview of the lessons learned from the application of PADAS. These are considered to be recommendations for future process mining projects applying RLPM.

7.1.1 Evaluating the Comparison Framework for PM Approaches

After the case study was performed, we evaluated the comparison. We believe that we have defined a concrete, complete comparison framework that is well tailored to process mining approaches in real life. Note that, as has been stated in section 1.4, the validation of the approach is limited to a single case study and *one* researcher. One could argue that the case is scoped by the author, since the efficiency and effectiveness of the execution of the three approaches depends on the prior knowledge and experience with the three approaches. In that sense, one could even argue that the researcher is the 'actual' case and the case study is 'just' input.

Overall, PM^2 provides detailed and concise directions to perform the process mining project [6]. PM^2 provided sound and practical guidance for each of the three process mining approaches that are compared in this thesis. Nevertheless, in lign with the authors that proposed PM^2, we agree that there is still a need for more detailed practical guidance. One of the challenges mentioned in the earlier work [6], namely for model-comparison, was tackled succesfully in this thesis by the application of the *Process Comparator* [43].

Reflecting on the input measures to determine the actual efficiency of a task To evaluate the use of Moody's model for method evaluation in the comparison framework we acknowledge that measuring the time (i.e. an input measure) was performed in a very inconvenient manner (e.g., manually by looking at he clock). It turned out to be a time consuming activity to do this manually. After the case study we identified a more automated approach; by using a tool to monitor the time a user is working with different software tools[1].

We also recognized and measured the number of steps as a possible input measure to determine the actual efficiency of a process mining approach. However after having executed and evaluated the comparison framework in the case study, this input measure was considered to be of a lesser interest for the comparison. The number of tasks is very subjective since it depends on how the researcher defines the tasks. Deviations the granularity of the definitions of the tasks make this measure less reliable.

Another challenge we recognize after applying PM^2 in practice considers the selection of PM tools and techniques. How should the analyst *determine the tools and techniques* to use? And where should this 'activity' be located in PM^2? This activity was already acknowledged as a separate actvity in previous work; PMPM [30], and it was placed in the planning stage. We

[1] On tools that could be applied are: https://www.rescuetime.com/, or https://toggl.com/

agree that this indeed should be a well-defined activity, just like "selecting business processes", "identifying research questions" (e.g. BPQs) and "composing the project team".

Additionally, perhaps the second stage of PM^2 should be redefined in a higher level of detail. The third and fourth stages were described on this level of detail already [6]. However, such an overview, displaying the order of activities and iterations is not defined for the extraction stage. During the execution of the research in this work, a refinement of the definition of *stage 2: Extraction* in PM^2 has arisen. Therefore, this thesis suggests an adjustment for the description of the second stage for extraction, as an extension to PM^2 for process mining projects in real-life environments. This refinement is presented in fig. 7.1. As this study has executed this stage with three different process mining approaches in a real-life environment, the proposed refinement is based on the practical experience.

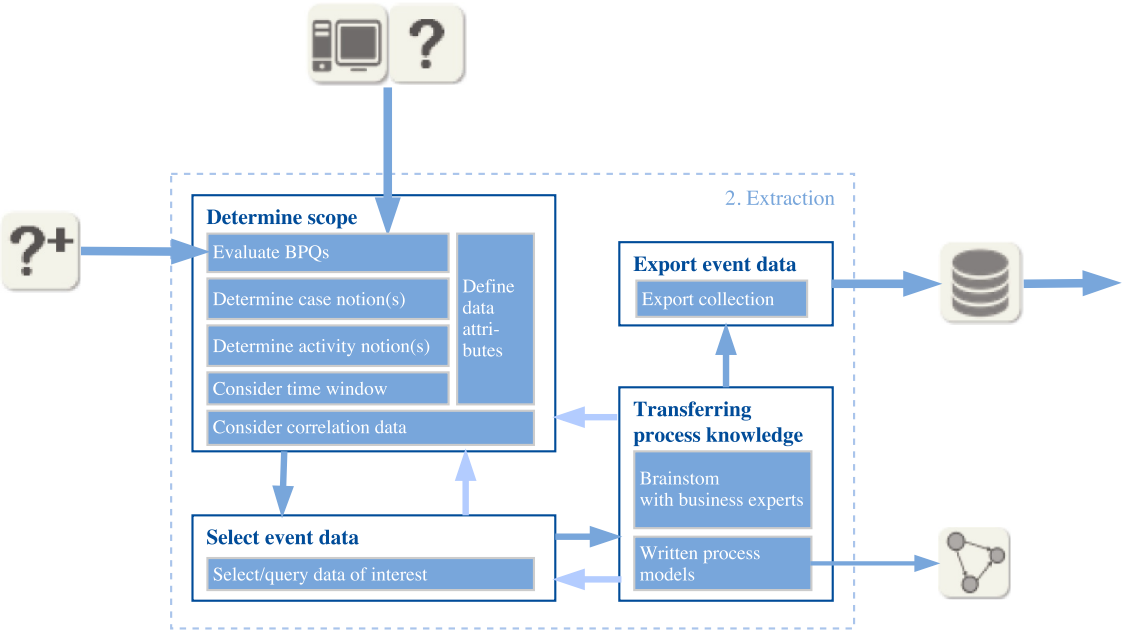


Figure 7.1: Refinement of definition of stage 2 of PM^2

7.2 Recommendations for Future work

Recommendations for future work entail the application of the comparison framework in more practical environment and with more process mining approaches. This will most definitely lead to more insights and more case characteristics (like the schema model type discussed in section 6.4) that lead to better performances of specific approaches.

Moreover, as explained in section 3.5, this case study mainly focused on the objective reality of the success of a process mining approach. The subjective reality was only measured by one person, namely the author of this thesis, thus the outcomes are subjective. For future work, we suggest that this subjective reality (by determining the PEoU and PU) would better be measured in a study with a group of persons that perform process mining tasks with process mining approaches. This will enlarge the sample size and give more reliable results.

Additionally, we found a recommendation for future work the RLPM approach. Moreover, it might be interesting to classify research questions in order to better decide on the best-suited process mining approach. The remainder of this section explains these ideas in more detail.

Create Trace attributes with the RLPM approach As has been discussed in section 6.2.2, all the attributes that were originally attributes in the columns of the tables on the data base become event attributes. This creates some complexity for filtering tasks. These hurdles can be

overcome by manually manipulating the event log, but this is not the ideal situation. Therefore, we would like to recommend future work to be devoted to tackling this issue.

Finding a best-suited Process Mining Approach based on the type of BPQs During the first stage of PM^2, the goals of the process mining project are translated into business process questions (i.e. BPQs). Business process questions can be related to different aspects of business processes, e.g. quality, time, resource, cost. In some real-life cases, the questions about the business process are not specific or concrete. However, these more abstract business process questions from the initialization phase can be refined through exploratory analysis and eventually result in concrete improvement ideas and valuable insights. These business process questions can possibly be used to determine which process mining approach would be the most suitable. We suggest that further research could dive into this topic. One could, potentially, make a 'better' decision for a process mining approach based on the type of business questions questions. This could for example be achieved by defining a good categorization framework for BPQs. Based on the categories that the BQPs would belong to, possibly, a set of PM approaches could be pointed out as useful/more suitable. As a starting point, this work suggests that business process questions could possibly be classified to be abstract or concrete, and to be work-flow related or more statistical measurements, see table 7.1 below.

Table 7.1: Suggested starting point for categorizing business process questions.

Abstract BPQs
– What does the process look like?
– What can be improved in the process?
– How can the throughput time of the process be reduced?
Concrete BPQs
– Is the process performing well, looking at the KPIs?
– Where is (possibly unnecessary rework) performed in the process?
– At which points in my process are products scrapped?
– Where are the bottlenecks in the system?

More concrete questions can be about the performance of the process with respect to the predefined Key Performance Indicators (KPIs). KPIs are instruments with the purpose of measuring the performance of the process. Typical KPIs with respect to operational performance are; delivery in full, on time rate (DIFOT), Order Fulfillment Cycle Time, Capacity Utilization Rate, Time to Market, Rework Level, Process or Machine Downtime Level, and many more. These KPIs usually have are statistical measurements to assess the performance of the process. For these type of questions, one could state that a simple and fast PM approach (e.g. Celonis process mining) would be preferred over a more complex PM approach that offers a wide variety of discovery algorithms. Concrete, statistical questions, can often also be solved with data mining techniques to analyze the process performance.

Other concrete questions can be related to the compliance of the business process. Is one particular activity always followed by a specific other activity? These can be quality checks, or process with a four eyes principle. The process mining techniques for conformance checking and model enhancement are excellent for solving these work-flow related questions. The more abstract questions and work-flow related questions could possibly be addressed more conveniently with exploratory process mining techniques. For example, by creating visual representations of the as-is process is excellent to get an overview of the actual process. Process mining methods like conformance checking can also be useful to detect loops, waiting times and undesired work-flow behavior. These type of business process questions would likely be better answered with a PM approach that offers the possibility of conformance checking, e.g traditional process mining.

7.3 Redo Log Process Mining, Lessons Learned

As demonstrated in chapter 4, the application and verification of PADAS in real-life was accomplished successfully in our case study. Provided that all the requirements (listed earlier in section 4.1) are met, the RLPM approach is considered to be a viable alternative for the TPM approach. This section presents the lessons learned from that application of redo log process mining. The purpose of this section is to give our recommendations for future process mining projects applying RLPM. The learned lessons, presented below, are generally applicable in real-life environments.

Archive the redo logs Usually, online redo logs are overwritten after two weeks or less, simply because of storage limitations on the database. By enabling the `ARCHIVE LOG` mode on the database, the online redo logs will be written to archived logs when there is no space left for online redo logs. This allows for gradually building a set of archived logs over any desired period of time. Also, to minimize the storage that is needed by the production environment for producing the redo logs, it is advised to not use online redo logs, but archived logs instead.

Configure the archiving to directories per day By having a directory for each day, containing 24 archived logs each, it becomes easier to discover and locate missing archived logs. This is required to manage the archived logs in a convenient manner.

Set up a storage facility In order to collect the archived logs to a desired location, a storage location should be created (if not present yet). Ensure that the size of the directory/disk is sufficiently large and preferably has a high scalability. The total size of the archived logs collected for our case study was in about 1,15 TB. However, this is heavily depending on how intensive an IS is used, e.g. the average duration of a case, the number of cases. The OTRS system, i.e. the PAIS of this case study, is considered a relatively 'calm' environment.

Relocate archived logs in a safe separate repository It is a strong recommendation to archive the redo logs, and relocate these redo logs to a separate repository in a timely fashion. In the case study, sometimes archived logs were removed, seemingly at random. It turned out that Oracle, when running out of space, removes old archived logs, if a newer backup of the database is already created. Oracle tends to evaluate these older archived logs as being obsolete. In case of storage shortage, Oracle removes these old archived logs. To prevent this from happening, relocating the archived logs, preferably every day, is a strong recommendation. The script that has been used in this case study to do so is provided in fig. A.3 in the appendices.

Together with the flow chart we created to check the feasibility RLPM with PADAS presented in appendix A.2, the lessons shown above should provide practical guidance for future deployments PADAS. We want to stress that, even though we managed to resolve the issue of missing data in the archived logs for our verification, it is vital to ensure that the set of archived logs is complete. The reason for this is that PADAS uses the values of the attributes of the objects provided in the redo logs to construct the events. If an event would be missing, the reconstructions of events after a gap are based on the wrong 'version' of the database. When using other process mining approaches, this is a risk with less impact. E.g., with the traditional process mining approach, a gap in the event data results in a few incomplete traces. Having a complete set of consecutive archived logs is especially crucial when a process mining project solely relies on the availability of redo logs. Because in that scenario, it is impossible to trace back how the database was changed (i.e., identify the affected traces).

Bibliography

[1] van der Aalst, W., *Process mining: Data Science in Action*, 2nd ed. Springer, 2016. 1, 2, 7, 9, 11, 23, 117

[2] van der Aalst, W., Adriansyah, A., De Medeiros, Ana Karla Alves, Arcieri, F., Baier, T., Blickle, T., . . . Wynn, M., "Process Mining Manifesto," in *Lecture Notes in Business Information Processing*, vol. 99, 2012, pp. 169–194. 2, 19, 21, 22, 47

[3] González López de Murillas, E., van der Aalst, W. & Reijers, H. A., "Process mining on databases: Unearthing historical data from redo logs," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9253, 2015, pp. 367–385. 3, 4, 10, 26, 56

[4] Dumas, M., van der Aalst, W.M.P. & Ter Hofstede, A.H., *Process-aware information systems: bridging people and software through process technology.* John Wiley & Sons, 2005. 7

[5] Chen W., "E-research event data quality," in *IEEE 29th International Conference on Data Engineering Workshops (ICDEW).* IEEE, 2013, pp. 314–317. 8

[6] van Eck, M. L., Lu, X., Leemans, S. J. J., & van der Aalst, W. M. P., "PM 2 : A process mining project methodology." in *Advanced Information Systems Engineering, CAiSE*, vol. 9097. Springer, 2015, pp. 297–313. 8, 12, 14, 31, 57, 58, 74, 82, 88

[7] van der Aalst, W.M.P., "Extracting event data from databases to unleash process mining," in *BPM-Driving innovation in a digital world.* Spinger International Publishing, 2015, pp. 105–128. 8, 10

[8] Bose, R. J. C., Mans, R. S., & van der Aalst, W. M. P., "Wanna improve process mining results?" in *IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, vol. 9694. IEEE, 2013, pp. 127–134. 8, 18, 19, 21, 42, 46, 117

[9] Günther, C.W., Rinderle-Ma, S., Reichert, M., van der Aalst, W.M.P. & Recker, J., "Using process mining to learn from process changes in evolutionary systems," *International Journal of Business Process Integration and Management*, vol. 3, no. 1, pp. 61–78, 2008. 8

[10] Günther, C.W. & van der Aalst, W.M.P., "Fuzzy mining–adaptive process simplification based on multi-perspective metrics," in *International Conference on Business Process Management.* Springer, 2007, pp. 328–343. 9

[11] van der Aalst, W., *Process mining : Discovery, Conformance and Enhancement of Business Processes*, 1st ed. Springer, 2011. 9, 11, 24

[12] van Dongen, B.F., de Medeiros, A.K.A.,Verbeek, H.M.W., Weijters, A.J.M.M. & van der Aalst, W.M.P., "The ProM framework: A new era in process mining tool support," in *International Conference on Application and Theory of Petri Nets.* Springer Berlin Heidelberg, 2005, pp. 444–454. 9

[13] Günther, CW and Verbeek, E, "Xes standard definition version 2.0," 2014. 9

[14] P. Chau, "An empirical assessment of a modified technology acceptance model," *Journal of management information systems*, vol. 13, no. 2, pp. 185–204, 1996. 10

[15] Davis, F. D., "A technology acceptance model for empirically testing new end-user information systems: Theory and results." Ph.D. dissertation, Sloan School of Management, Massachusetts Institute of Technology, 1986. 10

[16] Moody, D.L., "The Method Evaluation Model: A Theoretical Model for Validating Information Systems Methods," in *ECIS 2003 Proceedings.*, no. 79, 2003. [Online]. Available: http://aisel.aisnet.org/ecis2003/79 10, 11, 13, 15, 25, 35, 40, 52

[17] Davis, F. D., "Perceived usefulness, perceived ease of use, and user acceptance of information technology." *MIS Quarterly*, vol. 13, no. 3, pp. 319–340, 1989. 10, 25, 52

[18] Davis, F., Bagozzi, R., & Warshaw, P., "User acceptance of computer-technology - a comparison of two theoretical-models," *Management Science*, vol. 35, no. 8, pp. 982–1003, 1989. 10

[19] Hu, P.J. and Chau, P.Y,K., Sheng, O.R.L., & Tam, K.Y., "Examining the technology acceptance model using physician acceptance of telemedicine technology," *Journal of management information systems*, vol. 16, no. 2, pp. 91–112, 1999. 10

[20] Taylor, S. & Todd, P.A., "Understanding information technology usage: A test of competing models," *Information systems research*, vol. 6, no. 2, pp. 144–176, 1995. 10

[21] Adams, D.A., Nelson, R.R. & Todd, P.A., "Perceived usefulness, ease of use, and usage of information technology: A replication," *MIS quarterly*, pp. 227–247, 1992. 10

[22] B. Szajna, "Empirical evaluation of the revised technology acceptance model," *Management science*, vol. 42, no. 1, pp. 85–92, 1996. 10

[23] Venkatesh, V. & Davis, F.D., "A theoretical extension of the technology acceptance model: Four longitudinal field studies," *Management science*, vol. 46, no. 2, pp. 186–204, 2000. 10

[24] Venkatesh, V., & Bala, H., "Technology acceptance model 3 and a research agenda on interventions," *Decision Sciences*, vol. 39, no. 2, pp. 273–315, 2008. 10

[25] Fayyad, U., Piatetsky-Shapiro, G. & Smyth, P., "From data mining to knowledge discovery in databases," *AI magazine*, vol. 17, no. 3, p. 37, 1996. 11

[26] Matignon, R., *Data mining using SAS enterprise miner.* John Wiley & Sons, 2007, vol. 638. 11

[27] Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C. & Wirth, R., "CRISP-DM 1.0 Step-by-step data mining guide," 2000. [Online]. Available: https://www.the-modeling-agency.com/crisp-dm.pdf 11

[28] Bozkaya, M., Gabriels, J. & van der Werf, J.M., "Process diagnostics: a method based on process mining," in *International Conference on Information, Process, and Knowledge Management, eKNOW'09.* IEEE, 2009, pp. 22–27. 11, 12

[29] D. M. . M. N. Delias, P., "Business process analytics: a dedicated methodology through a case study," *EURO Journal on Decision Processes*, vol. 3, no. 3-4, pp. 357–374, 2015. 12

[30] Heijden, v. d., T.H.C, Reijers, H. A., Weijters, A. J. M. M., "Process mining project methodology: Developing a general approach to apply process mining in practice." Master's thesis, Eindhoven University of Technology, 2012. 12, 57

[31] Anon, "Tietosysteemin rakentaminen [Information system design]," *Tietojenkäsittelyliitto*, vol. 25, 1974. 17, 37

[32] Byström, K., & Järvelin, K., "Task complexity affects information seeking and use." *Information & Processing and Management*, vol. 31, no. 2, pp. 191–213, 1995. 17, 18, 37

[33] Barr, A. & Feigenbaum, E.A., "The Handbook of Artificial Intelligence, vol. 1," *Los Altos, CA: William Kaufmann*, 1981. 17, 18

[34] D. van de Ven, A.H. & Ferry, *Measuring and assessing organizations.* John Wiley & Sons, 1980. 18

[35] Günther C.W., van der Aalst, W.M.P. & Weijters, A. J. M. M., "Process mining in flexible environments," Ph.D. dissertation, 2009. 19, 20, 45

[36] Hompes, B.F.A., Buijs, J.C.A.M., van der Aalst, W.M.P., Dixit, P.M. & Buurman, J., "Discovering Deviating Cases and Process Variants Using Trace Clustering," in *Proceedings of the 27th Benelux Conference on Artificial Intelligence (BNAIC)*, pp. 5–6. 20

[37] Weber, P., Bordbar, B., Tiňo, P. & Majeed, B., "A framework for comparing process mining algorithms," in *GCC Conference and Exhibition (GCC), 2011 IEEE.* IEEE, 2011, pp. 625–628. 23

[38] Wang, J., Wong, R. K., Ding, J., Guo, Q. & Wen, L., "Efficient selection of process mining algorithms," *IEEE Transactions on Services Computing*, vol. 6, no. 4, pp. 484–496, 2013. 23

[39] Wang, J., Tan, S., Wen, L., Wong, R. K., & Guo, Q., "An empirical evaluation of process mining algorithms based on structural and behavioral similarities," in *Proceedings of the 27th Annual ACM Symposium on Applied Computing.* ACM, 2012, pp. 211–213. 23

[40] van Dongen, B. F., De Medeiros, A. A. & Wen, L., "Process mining: Overview and outlook of petri net discovery algorithms," in *Transactions on Petri Nets and Other Models of Concurrency II.* Springer, 2009, pp. 225–242. 23

[41] Leemans, S.J., Fahland, D., & van der Aalst, W.M.P., "Exploring processes and deviations," in *International Conference on Business Process Management.* Springer, 2014, pp. 304–316. 24

[42] Görg, C., Pohl, M., Qeli, E., & Xu, K. , "Visual representations." in *Human-Centered Visualization Environments.* Springer Berlin Heidelberg, 2007, vol. 4471, pp. 163–230. 42

[43] Bolt, A., De Leoni, M., & van der Aalst, W. M.P., "A visual approach to spot statistically-significant differences in event logs based on process metrics." in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9694, 2016, pp. 151–166. 57

[44] Fogel, S., & Lane, P., "Oracle Database Administrator's Guide, 10g Release 2 (10.2); Managing Archived Redo Logs." December 2016. [Online]. Available: https://docs.oracle.com/cd/B19306_01/server.102/b14231/archredo.htm 65

[45] Fruchterman, T. M., & Reingold, E. M., "Graph drawing by force-directed placement," *Software: Practice and experience*, vol. 21, no. 11, pp. 1129–1164, 1991. 93

Appendix A

Redo Log Process Mining in real-life Environments

A.1 Archiving Process of Redo Logs

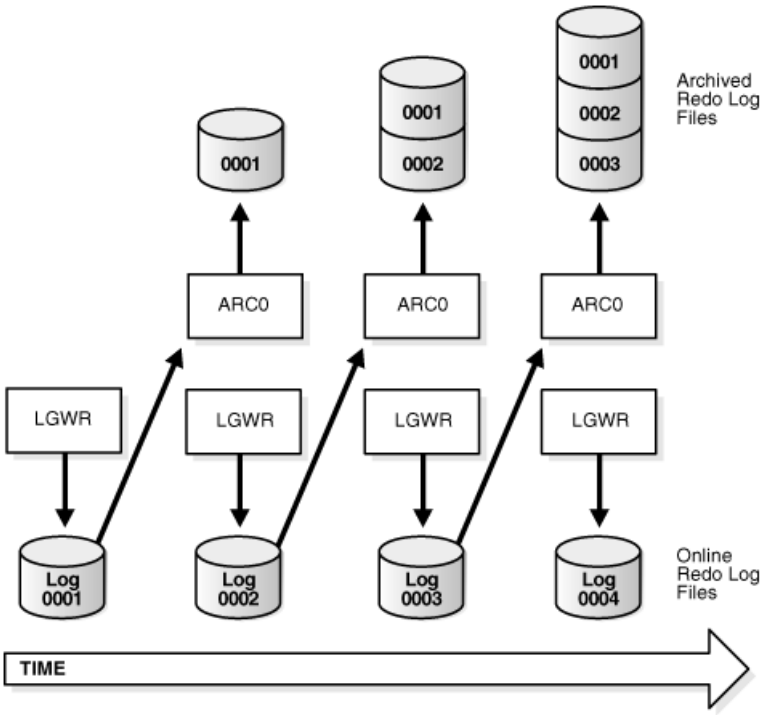


Figure A.1: An illustration of how online redo log files are archived, retrieved from [44].

In fig. A.1, the archiving of online redo logs is illustrated. In fig. A.1, there is only one archive process (ARC0) and only one archived redo log. Figure A.1 shows that LGWR writes to online redo log file 1 (Log0001). As soon as Log0001 is full, ARC0 writes it to the archived redo log. Next, LGWR writes to online redo log file 2 (Log0002). Again, when this is also full, ARC0 writes it to the archived redo log. The same patterns holds for the online redo log files 3 (Log0003) and 4 (Log0004).

A.2 Feasibility check for Redo Log Process Mining (PADAS)

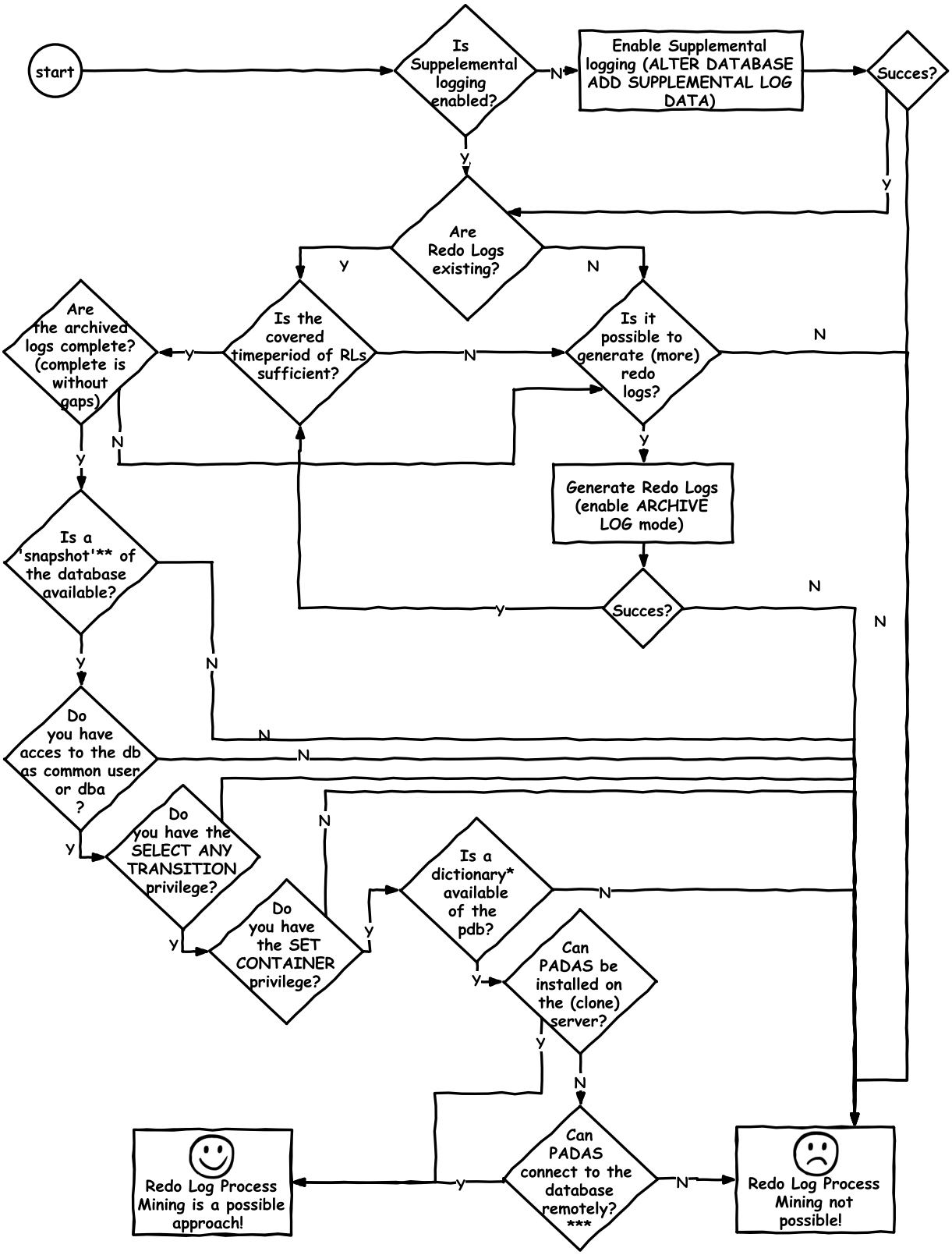


Figure A.2: Flow Chart for feasibility RLPM with PADAS (multitenant architecture Oracle RDBMS).

*To decide which type of dictionary should be used, check Oracle Documentation[a] **This can be a snapshot of the DB on the production environment, or a clone of the database. ***A local installation of PADAS can connect with a server remotely via an SSH tunnel and access the DB.

[a] Decision tree to select the correct dictionary: https://docs.oracle.com/cd/B28359_01/server.111/b28319/logminer.htm#i1014713

A.3 Script for automatically replacing the archived logs

```
#!/bin/bash

export ORACLE_SID=+ASM

DATUM=`date -d yesterday +"%Y_%m_%d"`

for i in $(asmcmd ls +RECO1/CPARCO2/ARCHIVELOG/''$DATUM'');
do
asmcmd cp +RECO1/CPARCO2/ARCHIVELOG/''$DATUM''/$i /u03/backup/archiveGra
done

ssh cbr-omf-002 'mkdir -p /data/archiveGra/''$DATUM'''

scp -r /u03/backup/archiveGra/  oracle@cbr-omf-002:/data/archiveGra/''$DATUM''

ssh cbr-omf-002 'sudo chown -R ghoogendoorn:parc /data/archiveGra/''$DATUM'''

rm /u03/backup/archiveGra/*
```

Figure A.3: Script to automatically replace the archived logs to a separate storage location.

Appendix B

Case Study Analysis

This chapter describes the complete execution of the process mining project for the three process mining approaches under study. The comparison framework that we defined in chapter 3 mentioned that to execute the PM project, we follow PM^2. It was already indicated that differences between the three approaches that are compared in this study mainly manifest themselves in stage 2 for extraction and stage 3 for data processing. In stage 4 for mining & analysis, the CPM approach is different from the TPM and RLPM approaches. Logically, for the stages in PM^2 that are executed in an equal manner for all the PM approaches, a comparison would be superfluous. The execution of these stages is performed only once, and thus described once.

Section B.1 describes how the *planning* stage is completed. Next, appendix B.2 describes three different completions of the *extraction* stage. Subsequently, this is done for the *data processing* stage in appendix B.3. Afterwards, appendix B.4 reveals how the *mining & analysis* stage is executed with the traditional process mining approach and the Celonis process mining approach. As is stated before, the redo log process mining approach would be executed similarly as the traditional process mining approach in this particular stage. Lastly, the completion of the *evaluation* stage is discussed in appendix B.5.

B.1 Completing Stage 1: Planning

Following PM^2, the *planning* stage forms the initialization of a process mining project. As was explained in section 3.1, the goal of this stage is: to compose a process mining team, select business processes, and identify business process questions. The outcomes of this stage serve as input for the succeeding *extraction* stage. The planning stage is completed similarly for all three approaches (TPM, RLPM, and CPM), thus the steps taken in this stage are discussed for all of them at once.

Firstly this section depicts the project goal for the case study and the process mining team. Next, this section discusses the selection of the business processes. Also, the identification of the business process questions is explained. Obviously, the right tools and techniques for this process mining project are are composed differently depending on the PM approach that is applied. An overview of which tool/technique is used in which PM^2 is given in table B.2. The RLPM approach required some extra activities for the planning of the process mining project. This section concludes with an explanation of these additional steps required for the RLPM approach.

Project Goal and Process Mining Team The project goal in this case study is different from regular process mining goals in real-life. As stated in section 1.3, the goal of this study is to compare PM approaches. To that end, to simulate a real-life PM project while conducting experiments, business process questions will be identified just like in any other PM project in a real-life setting. For this research, the composing of the project team was straightforward. The research was conducted by the author of this work, with the aid of three coaches. These are: the graduation supervisor, the graduation tutor and a business supervisor.

Selecting the Business Process For a PM Project one must identify the entities flowing through the process at hand, and how the activities are logged. Moreover, relevant questions are; how and where is logged data stored, in which (Process Aware) Information System, and what are the potential insights that can be derived from the available data? The selection of the business process for the case study was determined by the feasibility with respect to the goal of this research. The three earlier PM approaches that are compared in the case study should be applicable with the selected process. More importantly, the (PA)IS that supports the business process should be a feasible source of historical data for each PM approach under study. One requirement was that the PAIS was 'accessible' by the researcher. Moreover, the case study should be neither to complex, nor to simple. Therefore, at least one unique identifier for the object flowing through the system should be available. Moreover, the process should be fairly understandable to the researcher.

The *process* involved in this case is mainly consisting of support/service by IT Helpdesk for different customers of the company. The objects flowing through the process are tickets (i.e. running reports of the IT issues). The process is supported by the a PAIS named OTRS, which logs events for the tickets. These are logged on the application level and kept in a table on the supporting database of OTRS. The historical data in this table, the `TICKET_ HISTORY` table, is suitable for the traditional process mining approach and the Celonis process mining approach. The OTRS system relies on an Oracle DB, by which redo logs can be generated, thus this business process can also be analyzes with the redo log process mining approach.

Domain knowledge about the PAIS was gained from online documentation (OTRS is an open source system), along with a demo of 20 minutes from an expert user, and an in-depth interview with an expert user of OTRS. Other inputs for knowledge about the process were the data model, and access to the web-interface of the system via a read-only account. The PAIS is used in an inter-organizational setting; i.e organization and its customers, thus one to many. The system is human-oriented, from person-to-person (P2P), although some monitoring applications are also integrated in the system, person-to-application (P2A). This results in a mixture of tickets, namely initiated by humans and by systems. Both ticket-types are interesting for analysis of solve times, and for the 'P2P'-cases it is interesting to analyze the response times. The process has been identified as a loosly framed process, since some process has been defined a priori.

From the interview it became clear that the PAIS is not used optimally since many functions are not used. E.g, functions to support the work-flow and manage the tickets according to SLAs are not used. The resources of the process use the it merely for administration purposes. Another interesting observation that resulted from analyzing the business process is that the cases (i.e. tickets) sometimes merge. These merges occurred, for example, in situations where two tickets are created (by accident) for the same issue. Moreover, this merge behavior appeared if one ticket from a customer and another ticket from that same customer (but from two different employees of that customer), are caused by the same error or problem. When two tickets were merged, a new row in a in the database. This row has three important attributes: the ticket IDs of the merged tickets, and the timestamp of the merge-event. This knowledge is input for the extraction and data processing stages, in which this behavior is analyzed and handled correctly.

Selecting the Business Process Questions The in-depth interview was the most important input for identifying business process questions (BPQs). The three main BPQs that were defined are shown below.

BPQ 4. How many tickets (i.e. cases) conform to the SLAs with respect to their response time and solve time (i.e. throughput time)?

BPQ 5. Are distinct customers treated differently in the process (e.g., with respect to the workflows and with respect to solve time and response time)?

BPQ 6. What are the differences in the performance of the distinct resources?

The next part of this section discusses the concepts that are involved in these BPQs (e.g.

incident management, service windows, SLAs, customers, and resources). To answer the BPQs, more concrete sub-BPQs are defined, and presented in the remainder of this section.

The purpose of the *incident management* service is to restore normal operations as quickly as possible with a minimal impact on the business and/or the customer, in a cost-efficient manner. An instance of the process (i.e. a ticket) is initiated by the customer but could also originate from management and monitoring systems. A ticket is*solved* when it has the status 'Closed successfully', and no remaining activities are left for this ticket. The *response time* of a ticket is the time between the moment it is reported by the customer (via email/phone), and the moment the service provider responds to the customer (via email/phone).

A *service window* is defined as the time frame during which support services from the IT service provider are available. The service window timings are decided upon by the company and the customer together. These are placed on record in the service agreement documents. In the *Service Level Agreements* (SLAs) that the company of our case study has signed with its customers, two different service windows are mentioned. See table B.1 for an overview of the SLAs, where performance indicators for the solve and response times of incidents (i.e. tickets with type incident) are given. Tickets with type 'incident' with the highest priority, i.e. 'Critical', should have 24 × 7 hours support. The second service window, for tickets of type 'incident' and, the priority 'High', 'Medium' or 'Low', is set from 08:00 to 18:00 h on working days (i.e., Monday to Friday). Depending on the priority of a ticket, 90% of the incidents should be solved within a particular period of time. The remaining 10% of the incidents should be solved withing the another set period of time, again depending on the priority class. Even so, depending on the priority, for 90% of the incidents the response time of the tickets should not cross a particular boundary. These percentages are measured on a monthly base.

Table B.1: Service Level Agreements for solve and repsonse times for tickets of type incident.

	Priority	Description	Response time	Solve time
1	Critical	Incidents that directly disturb very important business processes, or the availability, of production (these need to be reported by phone).	90% < 30 minutes (as long as registered by telephone)	90% < 4 hours, 10% < 8 hours
2	High	Incidents that have a high impact on the operations of one or several user(s) with a,large impact on important business processes.	90% < 2 hours	90% < 8 hours, 10% < 4 days
3	Medium	Incidents with little impact on vital business processes that can be solved with,temporary solution.	90% < 1 day	90% < 4 days, 10% < 7 days
4	Low	Incidents with no impact on the operations of the end-user.	90% < 2 days	90% < 7 days, 10% < 14 days

In order to answer BPQ 4 in a precise manner, we defined a total of twelve more explicit business process questions, see the list B.1 below. These twelve BPQs are concrete since they address exactly each of the SLAs from table B.1.

1.1 What percentage of tickets of type incident, with critical priority, considering a service window of 24 hours on 7 days, are solved within 0 to 4 hours?

1.2 What percentage of tickets of type incident, with critical priority, considering a service window of 24 hours on 7 days, are solved within 4 to 8 hours?

1.3 What percentage of tickets of type incident, with critical priority, considering a service window of 24 hours on 7 days, have a response time within 0 to 0.5 hours?

1.4 What percentage of tickets of type incident, with high priority, considering a service window from 08:00 to 18:00 o'clock on Monday to Friday, are solved within 0 to 8 hours?

1.5 What percentage of tickets of type incident, with high priority, considering a service window from 08:00 to 18:00 o'clock on Monday to Friday, are solved within 8 hours to 4 days?

1.6 What percentage of tickets of type incident, with high priority, considering a service window from 08:00 to 18:00 o'clock on Monday to Friday, have a response time within 0 to 2 hours?

1.7 What percentage of tickets of type incident, with medium priority, considering a service window from 08:00 to 18:00 o'clock on Monday to Friday, are solved within 0 to 4 days?

1.8 What percentage of tickets of type incident, with medium priority, considering a service window from 08:00 to 18:00 o'clock on Monday to Friday, are solved within 4 to 7 days?

1.9 What percentage of tickets of type incident, with medium priority, considering a service window from 08:00 to 18:00 o'clock on Monday to Friday, have a response time within 0 to 1 day?

1.10 What percentage of tickets of type incident, with low priority, considering a service window from 08:00 to 18:00 o'clock on Monday to Friday, are solved within 0 to 7 days?

1.11 What percentage of tickets of type incident, with low priority, considering a service window from 08:00 to 18:00 o'clock on Monday to Friday, are solved within 7 to 14 days?

1.12 What percentage of tickets of type incident, with low priority, considering a service window from 08:00 to 18:00 o'clock on Monday to Friday, have a response time within 0 to 2 days?

The *customers* are the clients, those are the organizations that issue and pay for the IT services that are provided by the company of the case study. The company under study has contracts with several customers, these contracts include the SLAs of table B.1. The customers are anonymized in this work for confidentiality reasons. It is interesting to find out if whether all the customers are provided with the same services. Differences can be found by looking at the solve and response times, but also in the sequence of execution of the workflows. Therefore, the following set of additional BPQs were defined. We determined the following more concrete business process questions to make BPQ 5 regarding the customers more explicit.

2.1 Are the SLAs met for each customer?

2.2 Which customers are delivered the worst and which the best service, in terms of solve times?

2.3 Which customers are delivered the worst and which the best service, in terms of response times?

2.4 Do the workflows display significant differences in terms of the duration of the activities, and the frequencies of the activities?

Lastly, the differences between the resources of the case study are analyzed. The resources are the users of the PAIS that supports the incident management process, thus these are the employees of the company of the case study. They solve and respond to the incoming tickets. In order to analyze and compare the performance of the resources, the following additional BPQs were identified. To protect the resources from being judged on their performances, their names are anonymized in this work. The following more concrete business process questions were defined to make BPQ 6 regarding the resources more explicit.

3.1 What is the median solve time of each of the three most active resources of the incident management process?

3.2 What is the median response time of each of the three most active resources of the incident management process?

3.3 Which resource performs best, with respect to the performance on the SLAs?

3.4 Are the resources working as a team on solving the tickets, or are they working separately?

Selection of Tools In the planning stage, it is sensible pay attention to the tools and techniques that are required in the next stages. For example the system that stores the data of interest, but also the *software tools* to extract the event data and to create the event log(s). Also, the process mining techniques that must be applied by what software should be thought of. The tools that were selected for the different PM approaches, per stage, are presented in table B.2.

Table B.2: Overview of the systems and software tools required per approach.

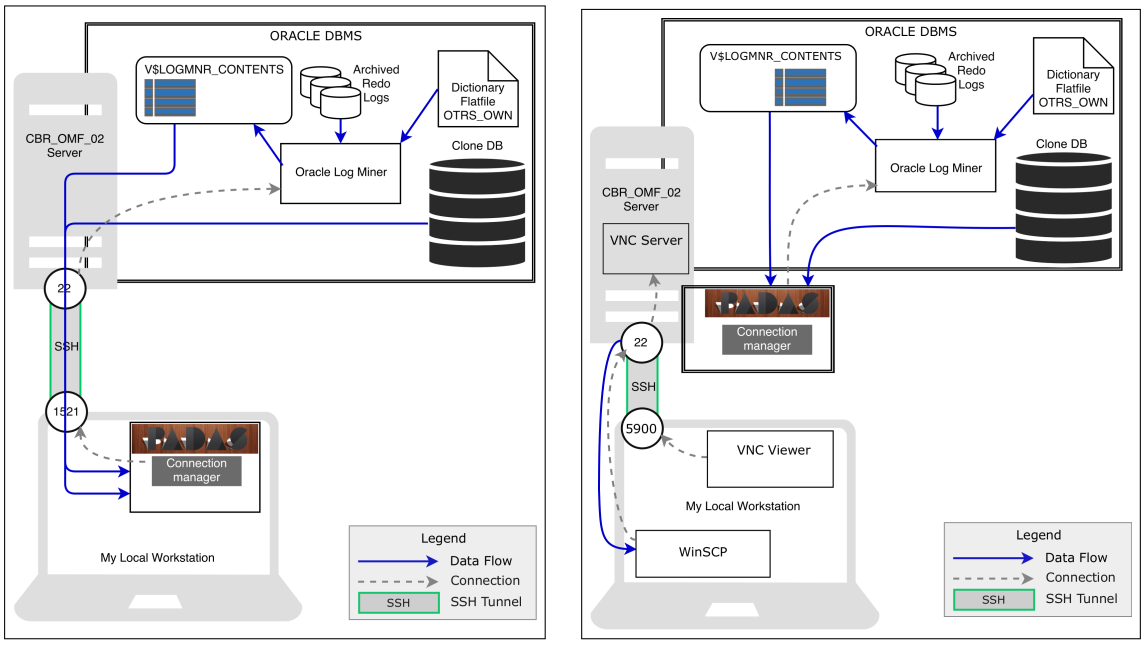
	Information System that stores the data →	**Software tools to extract event data** →	**Software tool(s) to create event log** →	**Process mining techniques to apply**
TPM	OTRS with Oracle Database on Linux Server* →	SQL Developer 4.1.3.20, PL/SQL →	ProM 6.6, RapidMiner 7.2.001 and RapidProm 4.0.0 →	ProM 6.6, RapidMiner 7.2.001 and RapidProm 4.0.0
RLPM	OTRS with Oracle Database, Log Writer & Archiver process on Linux Server* →	Process Aware Data Suite (PADAS) →	PADAS, RapidMiner 7.2.001 and RapidProm 4.0.0 →	ProM 6.6, RapidMiner 7.2.001 and RapidProm 4.0.0
CPM	OTRS with Oracle Database on Linux Server* →	SQL Developer 4.1.3.20, PL/SQL →	Celonis Process Mining 4 (CPM4) →	CPM4

*To be exact: OTRS 3.3 running on a Red Hat Enterprise Linux Server release 6.5 (Santiago), with the back-ed based on an Oracle Database 12c Standard Edition Release 12.1.0.2.0 64 bit,with PL/SQL Release 12.1.0.2.0 Production.

The source of historical data is equal for the TPM and CPM approaches. The data is stored in a table, with an instance for each event. Whereas for the RLPM approach, redo logs are used (archived by the Log Writer Process, i.e. an Oracle background process). For the extraction of event data, again the TPM and CPM approaches are applied with the same tool: Oracle's SQL Developer. This software can be installed locally and connected to the Oracle database (using a private key and SSH). This enables the user to direct execute queries on the database. This choice was based on convenience for the project, since it was proclaimed that the retrieval without a lot of domain knowledge about the process at hand, requires time and iterations. Moreover, in practice the SQL query language is commonly used and known. The RLPM approach uses its own approach for the extraction, namely PADAS. For the creation of an event log and data processing, the TPM approaches uses RapidMiner with the RapidProM Extension. Addtionally, for data processing purposes the TPM approach additionally uses ProM. The RLPM approach initially benefits from PADAS, as this produces an event log already. Moreover, the RLPM approach requires RapidMiner with the RapidProM Extension for data processing. The CPM uses the Celonis Process Mining 4 software suite (CPM4) to create an event log and apply any other filtering and/or analysis techniques. The PM techniques for the TPM and RLPM approaches are both performed in ProM and RapidMiner and RapidProM.

For the redo log mining approach, some additional planning activities were required. The main reason was that at the start of this project, no redo logs were archived. The redo logs had to be collected to a reasonable volume before any step to the next stage could be taken. The different steps that have been carried out, solely for the RLPM approach, are listed below.

- ✔ Configure the database in Linux;
- ✔ Arrange storage facilities for archived logs;
- ✔ Create setup for extraction;
- ✔ Make snapshot of database.



(a) (b)

Figure B.1: Two options for the setup for extraction with PADAS for the redo log process mining approach: a local installation in a and an installation on a server in b.

For a more detailed explanation of the requirements for the database configuration and the storage facilities, see section 4.1 and fig. A.2. To create a setup for the extraction, there are two options. In the first setup (fig. B.1a) PADAS is installed locally on some workstation, and a connection is made with the Oracle DB via een SSH tunnel (see fig. B.1a). In the second setup, PADAS is installed on the server and connects to the Oracle DB from there, where it is possible to connect a workstation to the server to work with PADAS (see fig. B.1b). The first setup was realized for testing, and the second setup was used to scale up and perform bigger extractions.

After having defined and fixed the setup for the extraction, the snapshot of the database needed to be created. There are two options for creating the snapshot: either *before* or *after* the set of archived logs. PADAS requires this snapshot to retrieve values that can not be extracted directly from the redo log, but only from the database content itself. For example, when a field is updated, the other fields in the same row remain the same and their values are not specified in the redo logs. In this case study the snapshot was made at the end of the redo logs.

In total, these preparations took more than 5 months. However, it was the first time that the redo log process mining approach was applied in practice, which explains this extra effort. Should the company of our case study start another process mining project in the future, the previously stated planning activities do not have to be completed again. Also, potential applications in other practical environments can more easily start up a process mining project with the redo log process mining approach by using this work for guidance. Therefore, the extra efforts listed above, are excluded from the comparative study.

By having selected the business process, and by having identified the BPQs, the planning stage of PM^2 is completed. In this first stage, we also analyzed the process characteristics and we defined the tools for the next steps in the process mining project. The resulting outputs of this stage are the BPQs and the set of information systems supporting the execution of the business processes to be analyzed. These are also the inputs for the extraction stage, which is discussed in the next section.

B.2 Completing Stage 2: Extraction

This section describes the steps and the results from deploying the traditional process mining approach, the redo log process mining approach, and the Celonis process mining approach in the second stage of PM^2. The process knowledge, the BPQs, an the knowledge about the PAIS that stores the historical data of the process, are derived from the previous planning stage. These are the inputs for this stage. As described in table 3.1, the three main activities in this stage are: determining the scope, extracting event data, and transferring process knowledge. The desired output of the extraction stage is event data and possibly process models.

This section describes the completion of *extraction* stage separately for each process mining approach under study. In appendix B.2.1 the TPM approach is explained, followed by the RLPM approach in appendix B.2.2 and the CPM approach in appendix B.2.3. finally, appendix B.2.4 presents the results for the input and output measures of the comparative framework proposed earlier in chapter 3 for stage 2.

B.2.1 Stage 2: Extraction with TPM

As is mentioned for the PM^2, event data can be created by selecting and collecting the process related data from the PAIS and joining them into a single collection of events [6]. For the extraction with traditional process mining, we created a query to do this. An overview of all the steps that were completed for this stage with TPM, see the list below.

- ✔ Connect to the database with SQL Developer;
- ✔ Define scope;
- ✔ Query the database;
- ✔ Validate the query for the number of cases and events;
- ✔ Create a new table and compose a view on the database (for the events and for the cases);
- ✔ Merge cases with PL/SQL script;
- ✔ Export to disk;
- ✔ Create Event log (add trace attributes to log with RapidProm);
- ✔ Load in ProM to test if extraction was successful.

The completion of the steps for the extraction were visualized previously in fig. 5.1a in chapter 5. The remainder of this section explains these steps in more detail.

The first step, is to *connect to the database* to explore the data schema in more detail. SQL Developer can be connected to the database remotely by using a setting up a connection using SSH with a private key. The *scope* of the project, the limitations that are created by the case, are to be considered next. For example, the start time of the period for the collection of data in this case study is bounded by the day that the system went live (from the 23rd of October 2014). Also, the process mining possibilities are limited by the information that is logged by the PAIS. Some tables in the database are empty, for example the SLAs table, this scopes the availability and richness of the historical data about the process. Nevertheless, sufficient information about the events were available for answering the BPQs that were identified in the previous stage.

Selecting the information of interest about the process with the TPM, a *query was created* with SQL Developer. This concerns information with respect to the events and the cases, in order to later on answer the BPQs. Accordingly, and creating views on the database were used to store these selections. One view was created to capture all the information about the events, and another view was created to capture all the information related to the cases. This results in the following set of attributes regarding information about the events and the cases (i.e. traces).

- The `ID` of the event from the `TICKET_HISTORY` table for sorting purposes;
- The timestamp (`CREATE_TIME`) of the event from the `TICKET_HISTORY` table;
- The `TICKET_ID` of the event from the `TICKET_HISTORY` table;
- The activity identifier `TICKET_HISTORY_TYPE` of the event table;
- The state of the event from the `TICKET_STATE` table;

- Both of the `TICKET_STATE` and the `TICKET_TYPE` of the event combined in one attribute to create an even more specific activity-label;
- The resource of the event from the `USERS` table;
- The priority of the event from the `TICKET_PRIORITY` table;
- The type of the message of the event (if applicable) from the `ARTICLE_TYPE` table;
- The sender-type of the event, again if applicable, from the `ARTICLE_SENDER_TYPE` table.

- The `ID` of the trace from the `TICKET` table;
- The `TN` of the trace from the `TICKET` table;
- The `CUSTOMER_ID` of the trace from the `TICKET` table;
- The `CUSTOMER_USER_ID` of the trace from the `TICKET` table;
- The priority of the trace from the `TICKET_PRIORITY` table;
- The type of the trace from the `TICKET_TYPE` table.

There was one clear straightforward choice for the case identifier for the event data, namely the `TICKET_ID`. The choice for the activity label could have been the `state` of a ticket or the `history_type`, or a combined version of them. However, the `state` was not chosen since granularity of events (i.e. the variety, V) was too low, which most likely results in too simple process models after mining. The combined (i.e. concatenated version), on the other hand, would have been too fine grained. Therefore the activity label was set to be the `history_type`. The other attributes that were selected to extract were decided upon the business process questions. Moreover, the foreign and primary keys linking these tables needed to be identified.

Accordingly, the selection of events and cases was counted, and confirmed to be correct. Finally, after 7 iterations the final queries were defined, the ones that satisfied all the needs for later stages (i.e. contain the information required to eventually answer the business process questions). These queries can be found in the appendices: fig. B.20 and fig. B.21. The selections of events and cases were saved as *views*. In SQL, a view is a virtual table based on the result-set of an SQL statement.

Next, to deal with the *merge behavior* in the data, we applied hierarchical queries. Oracle SQL allows for the use of hierarchical queries to deal with hierarchical data, which is fitting to the situation where tickets are linked to other tickets. However, it turned out that closed loops in these hierarchies are only detected by Oracle when they are self-loops (i.e. a link from a ticket, to itself). A second solution was applied to solve the merging problem, PL/SQL script was written. This script essentially walks through a link-tree, starting from every root node, i.e. parent (or 'source'), then walks down all the levels of leave nodes (i.e. children or 'targets'). See fig. B.2 for a clarifying figure about the how merges are stored on the database level. For all tickets that are linked to some parent, the original `TICKET_ID` is updated to the `TICKET_ID` of their root node. Mind that running the script would not work on a view, since changing case ids in a view, would also result in changing the actual data on the database. To that end, the event view that was created earlier with the queries was first loaded into a new table on the database before executing the PL/SQL script[1]. This allowed the PL/SQL script to change the case ids of the events without interfering in the original tables on the database in a safe manner. The full statements for creating a new table and the PL/SQL script are provided in the appendices, fig. B.22 and fig. B.23 respectively. The information about the cases (i.e. tickets) still resided in the case-view on the database.

The data was then *exported* to flatfiles (i.e. CSV files) to the local workstation. Accordingly, RapidMiner with the RapidProM extension was applied to create an event log. These CSV files were both imported in RapidMiner in order to combine them to *create an event log*, see fig. B.3. The data set containing the events was converted to an event log using 'Data Table to Event Log' operator. Accordingly, the 'Add Trace Attributes to Event Log' operator added the information from the case data to the just before created event log. Finally, this complete event log was *exported* to a directory on the workstation again.

This concludes the extraction with the traditional process mining approach. Next, in appendix B.2.2 we explain how the extraction stage is executed with the redo log process mining

[1] This could also have been a Global Temporary Table, but this one would not be accessible from another session, hence a session of Celonis, which also needs to be able to access this data

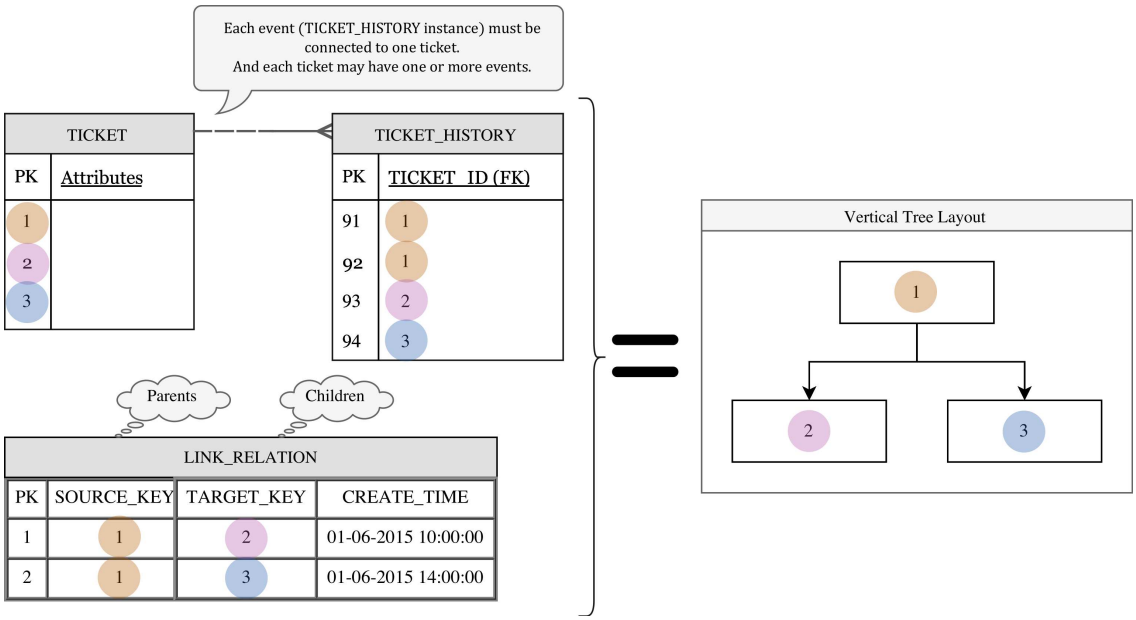


Figure B.2: The merging behaviour of tickets is logged in a separate table called `LINK_RELATION`, this can be visualized in a tree structure. In this simplified example, ticket 1 is the root, the parent node. Tickets 2 and 3 are the childeren of 1. This means that both ticket 2 and 3 are merged to ticket 1.

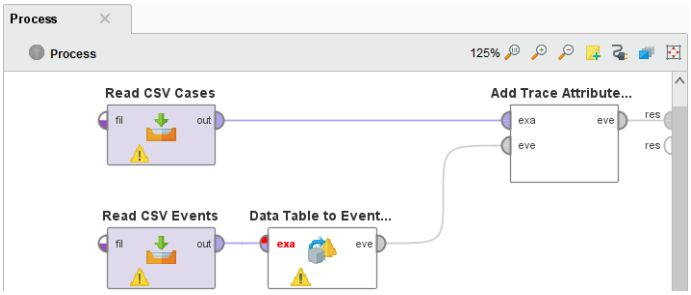


Figure B.3: The RapidMiner model that was created to merge two tables, one with event-data and one with case-data, in one event log.

approach. The same holds for the Celonis process mining approach in appendix B.2.3.

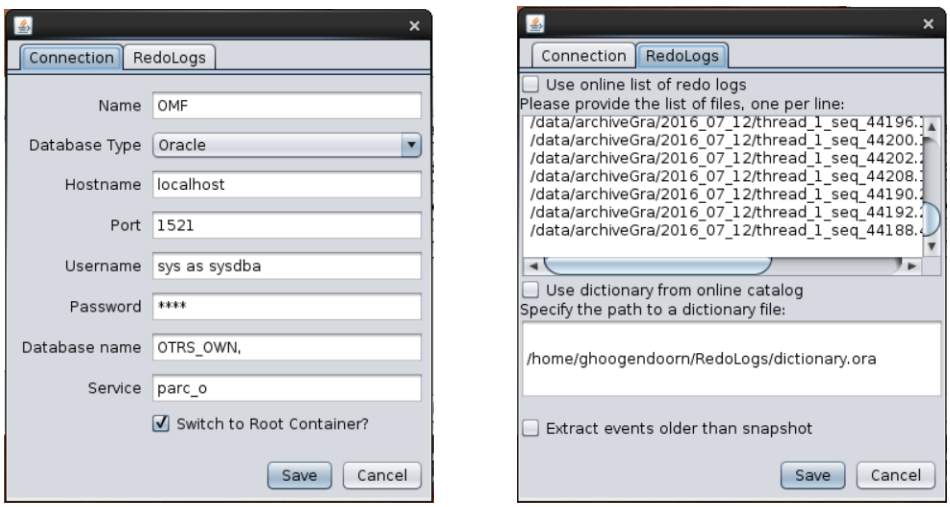
B.2.2 Stage 2: Extraction with RLPM

The steps taken for the redo log process mining approach in the extraction stage are listed below.

- ✔ Configure *Connection Manager* of PADAS;
- ✔ Define scope;
- ✔ Create data model in PADAS;
- ✔ Extract events with PADAS;
- ✔ Split log, create perspective with the *Log splitter* of PADAS;
- ✔ Export to `.xes` with PADAS;
- ✔ Load in ProM to test for a successful extraction;
- ✔ Merge the cases with RapidMiner;
- ✔ Remove corrupted cases with RapidMiner;

These steps in the extraction process with the RLPM approach were visualized previously in fig. 5.1b. The remainder of this section explains these steps in more detail.

For the RLPM approach, the *scope* was strongly defined by the period of time for which events are collected. This period of time is scoped to the period in which redo log files are logged, which in the case study was bounded from the 11th of March 2016 to the 12th of July 2016. This resulted in a set of archived logs of a period of 124 days including the end date. The rest of the scope definition was exactly performed as how it was performed in the traditional process mining approach; the data model was inspected to find out which tables contain the information of interest.



(a) (b)

Figure B.4: Configuration of the connection manager in PADAS: the connection settings a and the redo log settings in b.

The next step, *Configuring the connection manager* in PADAS, requires the setup as was indicated in fig. B.1b. To connection manager is used to setup the connection. One needs to provide an arbitrary name for the connection, set the database type, provide the hostname and port to connnect to, along with the username and password. Also, the name of the database and the service name are required, see fig. B.4. The thickbox 'Switch to Root Container?' should be ticked when working with a multitenant database.

To *create the datamodel* in PADAS, one first loads all the tables in from the selected database (in this case study named 'OMF') in the 'Tables Lists' window. Besides, seeing all the tables listed in this window is a check to verify if the connection to the database was successful. Next, one selects the tables of which events should be extracted and clicks the button 'Save Data Model'. Accordingly, a name for the data model (in our example 'My DataModel') can be provided and the extraction is started, see fig. B.5. PADAS first extracts all the events in the archived logs,

including their ordering. Next, PADAS selects all the events that are related to any of the tables that were selected.

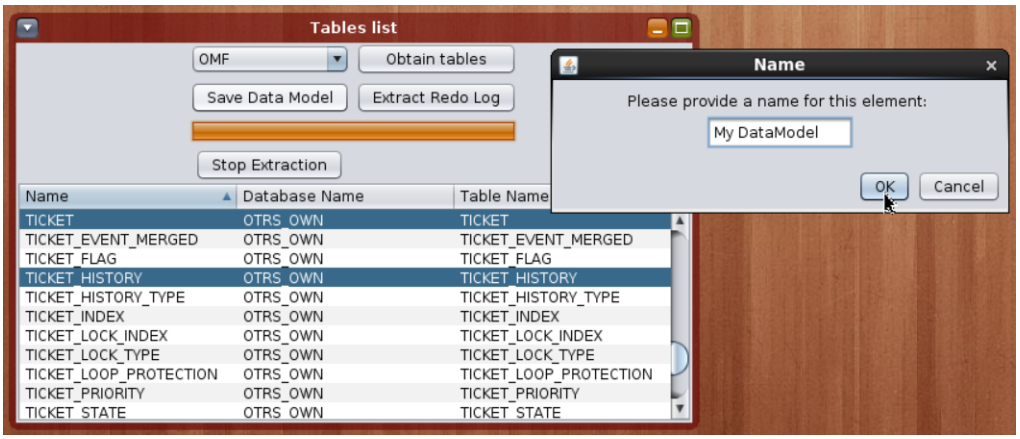
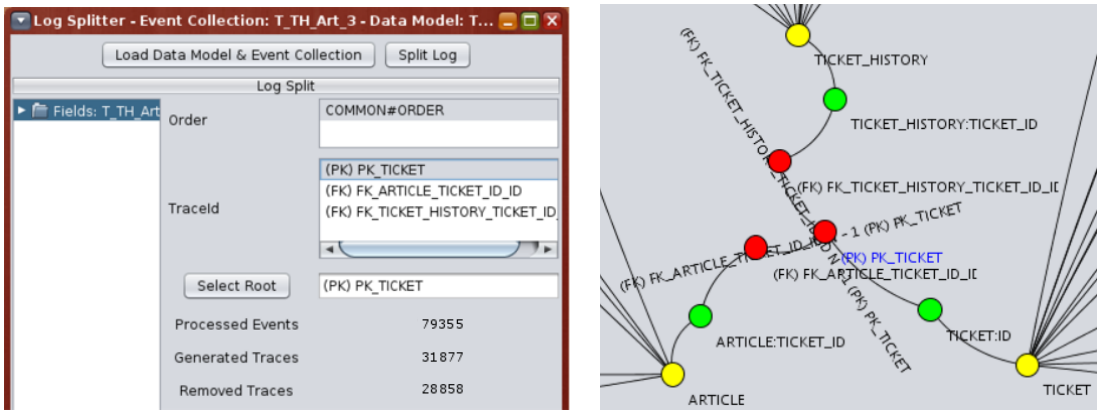


Figure B.5: Creating a data model in PADAS.

The selection of tables for the extraction really depends on the tables that represent the objects of interest. The data schema of the database of the PAIS of the case study contained three such objects. The table `TICKET`, `TICKET_HISTORY` and `ARTICLE` are selected. A visualization in an Entity Relationship Diagram of these tables is given in fig. B.24. Each `article` must have a `TICKET`, and one `TICKET` may or may not have one or more `article` objects. The `ticket` has the same relation with `TICKET_HISTORY`. Each `ARTICLE` may or may not have one or more `TICKET_HISTORY`, and one `TICKET_HISTORY` may or may not have one `ARTICLE`.



(a) (b)

Figure B.6: Create perspective with the Log splitter of PADAS in a and the visualized in PADAS in b.

Next, the *log is splitted*. To do this, one selects the data model and the event collection that have been created, and then loads them into the 'Log Splitter' window. The order of the events should be selected and the trace id pattern should be defined. We selected the `COMMON_ORDER` which is the order in which the events took place in reality. For the trace id pattern we selected the primary key (`PK_TICKET`) of the `TICKET` table as a root. We also selected two foreign keys that point to this primary key, these are the `(FK_TICKET_HISTORY_TICKET_ID_ID` of the `TICKET_HISTORY` table and the `FK_ARTICLE_ID_ID` of the `ARTICLE` table, see fig. B.6.

Moreover, missing data in the archived logs also scoped the extraction of the data. The source of the issue was explained in section 4.2. To solve this problem for the case study, we removed the

incomplete traces from the event log that we created with PADAS. The cases of which events had occurred in the 'gaps', are incomplete, i.e. these cases have become unreliable cases. One could say that the archived logs are corrupted in that sense. For mining & analysis (which is stage 4 of PM^2) purposes these cases are therefore useless since any results would not be trustworthy. Thus, these affected traces are identified and removed from the extracted event log.

B.2.3 Stage 2: Extraction with CPM

The steps taken for the Celonis process mining approach in the extraction stage are listed below. See fig. 5.1b for a process model that visualizes how theses were steps were executed. The remainder of this section explains these steps in more detail.

- ✔ Connect to the database with SQL Developer;
- ✔ Define scope;
- ✔ Query the database;
- ✔ Validate the query for the number of cases and events;
- ✔ Create a new table and compose a view on the database (for the events and for the cases);
- ✔ Merge cases with PL/SQL script;
- ✔ Connect to the database with CPM4;
- ✔ Create a new table and compose a view on the database (for the events and for the cases);
- ✔ Insert merged event data in the newly created table using SQL Developer;
- ✔ Create data mode;.
- ✔ Load new project in CPM4 and test.

The first six steps that are listed above are executed exactly the similar to as they were executed with the TPM approach, see appendix B.2.1 for the explanation. Accordingly, to *connect to the database with CPM4*, one can setup a connection using the templates that are offered in CPM4. We used to Oracle template and configured it as is given in fig. B.7. For a successful connection one needs: to define a name for the connection, to compose the connection string (with the port-names and the service name), fill in the user name and the password , and the schema name as well. For the driver, to connect to an Oracle database, one needs the oracle.jdbc.OracleDriver.

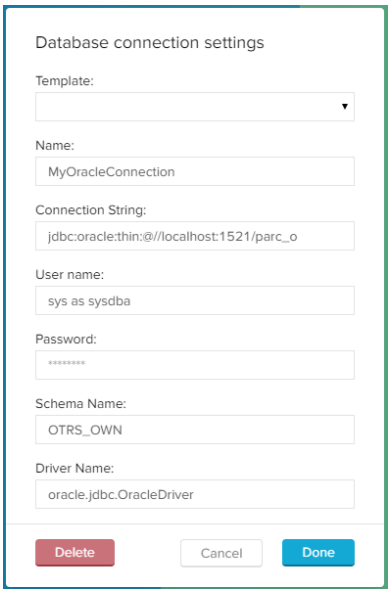


Figure B.7: Configuring the connection to the database with CPM4.

With CPM4, we selected two objects from the database to *create a data model*; the *case-view* in which each row represents a case, linked with the case identifier to the *event-table* in which each entry represents an event. This link can be created manually in Celonis, similar to a foreign key (attribute in event-table representing the case identifier) that points to the unique primary key (the case identifier) of the case view, see fig. B.8.

A different option for event extraction would be to import all the tables, that contain bits and pieces of information about the events, separately in CPM4. Then, one would not have to complete the queries in SQL Developer, and no views or tables would have to be created. However, this scenario requires extra activities in CPM4. Namely, one will have to manually configure all the relations between tables that are loaded in CPM4. Would we have applied this in the case that us currently under study, that would have resulted in a time consuming activity with an inconvenient result. Because in this case study, there are many tables and relations that keep information about the events.

As was explained in chapter 3, we measure input and output measures per stage of PM^2 to eventually determine the actual efficacy of each PM approach. The next section shows the resulting measurements for the extraction stage, for three different process mining approaches.

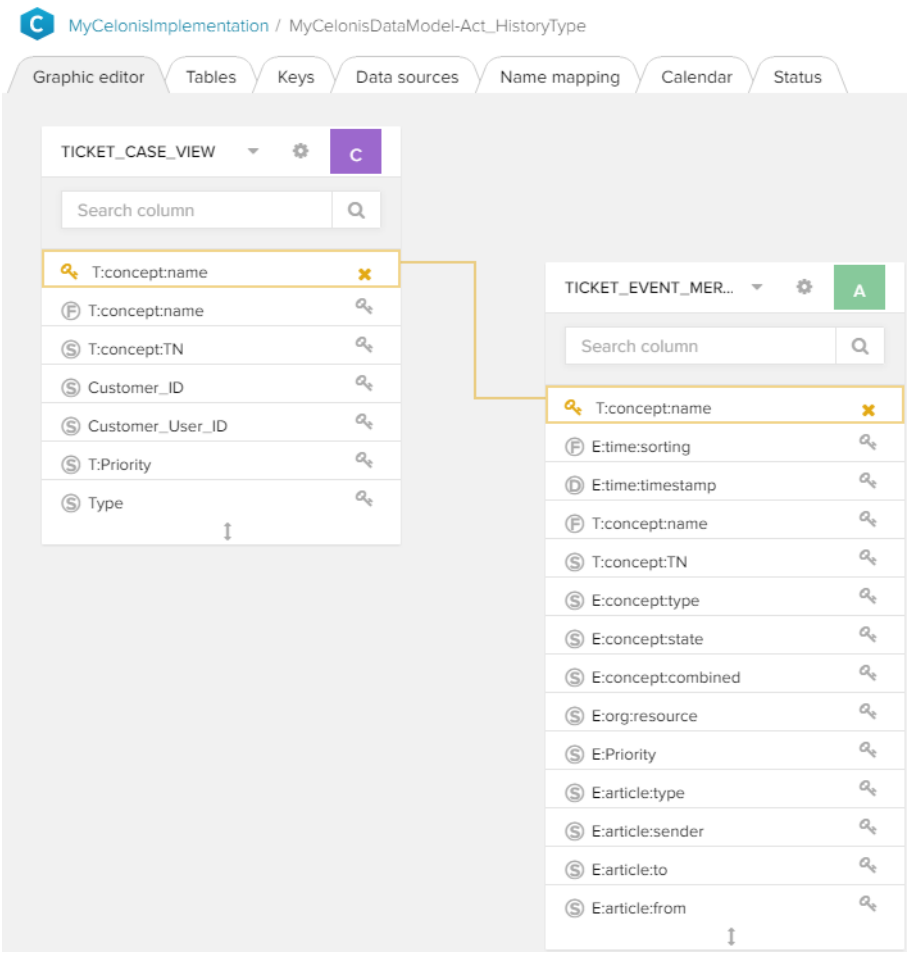


Figure B.8: Manually defined relation between cases and events in CPM4.

B.2.4 Resulting measurements for Stage 2: Extraction

The previous three sections (B.2.1 till B.2.3) described all the steps to complete the extraction stage of PM^2, for three different process mining approaches. As was explained in the *comparison framework for process mining approaches* that was presented in chapter 3, we structurally compare these process mining approaches for each stage of PM^2. Following this comparison framework for the extraction stage, data was collected with respect to the effort that was required to complete the stage, and also regarding the quality and quantity of the outcome (i.e. the event data). Table B.3 presents the results for the input and output measures for determining the actual efficiency and actual effectiveness of a process mining approach in the second stage of PM^2.

Table B.3: Metrics describing the efficiency and effectiveness of Stage 2 of PM^2.

Metrics to determine actual efficiency (input measures)

Index	Label	Brief description	TPM	RLPM	CPM
2.1	TIME	Time required to complete stage (hours)	112	230	112
2.2	IT-S	Iterations within current stage	8	2	9
2.3	IT-OS	Iterations to previous stages	0	0	0
2.4	DK-O	Domain knowledge objects	16	4	16
2.5	DK-R	Domain knowledge relations	19	4	19
2.6	COMPL	The task complexity	3	4	3

Metrics to determine actual effectiveness (output measures)

Index	Label	Brief description	TPM	RLPM	CPM
2.7	TIMECOV	Time window captured (days)	629	124*	629
2.8	M	Magnitude (# of cases)	17 822	3 019	17 822
2.9	A-NOT	Number of activity notions	2	2	2
2.10	SP	Support (# of events)	281 906	31 877	281 906
2.11	C-NOT	Number of case notions	1	1	1
2.12	$V(L)$	Granularity of events	31	23**	31
2.13	GRAN-T	Granularity of timestamps	seconds	seconds	seconds
2.14	ATTR	Total number of unique data attributes	15	81	17

*This is including the gaps. ** This is 52 if the activity mapping of section 4.3 would be applied.

The results from table B.3 are discussed in the chapter 6. Among others, this chapter visualizes the here presented results, and compares the differences in the efficiency and effectiveness. The next section of this chapter elaborates further on the completion of the data processing stage of PM^2, which is the next one after the extraction stage.

B.3 Completing Stage 3: Data Processing

The inputs for the data processing stage is the event data, and the outputs are event logs. Event logs are specific views on event data, created by defining case notions and event classes. Four types of activities that were identified for this stage are: creating views, aggregating events, enriching logs, and filtering logs [6]. For a more detailed explanation of this work, see section 3.1.

Table B.4, given below, provides a general description of the views that should be created in order to answer the BPQs successfully. The analysis is performed on the three largest customers and the three most active resources. Other customers and resources are not analyzed to simplify the analysis for this case study. Also, the analysis on the larger customers and more active resources is, logically, based on a larger data set and thus results in a more reliable analysis.

The desired outputs for this stage, i.e. event logs, were generated with each of the different PM approaches that are compared in this work. The following sections, B.3.1 till B.3.3, elaborate on the steps that were performed to complete the data processing stage with the three PM approaches under study: TPM, RLPM and CPM. Lastly, appendix B.3.4 presents the results for the input and output measures of the comparative framework that was proposed in chapter 3 for stage 3.

Table B.4: Identified outputs of the third stage, based on the Business Process Questions.

	Description of view
a.	All the start and end-event of each incident-type ticket; the timestamps, priority, customer, and resource, enriched with the solve-times
b.	All incidents, with an activity that is considered to be request followed (directly or indirectly) by an event that is considered to be a response: the timestamps, priority, customer and resource, enriched with the response-time
c.	All events where the case is of type 'incident', and all the attributes of relevance

B.3.1 Stage 3: Data Processing with TPM

The steps taken for the traditional process mining approach in the data processing stage are listed below. These steps were visualized previously in fig. 5.2a. The remainder of this section explains these steps in more detail.

- ✔ Apply trace attribute filter (in ProM);
- ✔ Remap values of attribute;
- ✔ Create view with enriched event log with the solve times;
- ✔ Create view with enriched event log with the response times;
- ✔ Apply event attribute filter (in RapidMiner);
- ✔ Export to disk.

Firstly, all tickets of the type 'incident' were selected to create a view with only incidents (view c. in table B.4). Therefore a *filter was applied on the trace attribute*, on the `type` of a ticket, with the 'Filter Log on Trace Attribute Values' plugin in ProM. Next, the values of the priority needed to be *remapped*, multiple values were mapped to eventually create four classes (Critical, High, Medium, Low). The 'Map' operator was applied in RapidMiner to create this mapping.

Also, a separate view was created from this event log, with for each case, the first and the last activity (including the timestamps), and a new attribute was derived from the existing attributes; the *solve time*. According to the SLAs from table B.1, the performance of the cases on the solve time depends on three inputs; start time, the end time, and the priority of the ticket. This attribute could take nine different values, where there a four classes of priority (see table B.1). To identify the priority of a ticket (i.e. the priority of the last event of a ticket), a values of the events regarding the priority attribute needed to be remapped. See fig. C.3 for a visualization of how

this mapping should behave. To find the start-event and end-event per trace, the event log was transformed to a data table in RapidMiner. After sorting the events in chronological order per trace, a custom Groovy script, used within the scripting operator, loops over all the events and outputs the desired information. The resulting data table contains for each case: one row with the start event and its timestamp, and one row the end event and its timestamp, and the priority of the end-event. Next, with this output as an input another scripting operator, using Python, calculates the solve times while taking into account the different service windows depending on the priority of a ticket, which is finally is written to a CSV file. Thus, the calculation of the solve time is straightforward, considering that some tickets (with a High, Medium, or Low priority) only the working hours from Monday to Friday can be accounted for.

The same strategy is followed to enrich the log with *response times*. However, firstly, a selection was applied to find only the tickets that are actual P2P-tickets, initiated by a customer with a message via phone or email (which is called a 'request'). Additionally, this request should (in)directly be followed by a response. A response is defined as a message via phone or email from a resource (i.e. a worker/agent of the company under study). To filter out all the cases that conform to these specifications, a RapidMiner model was created. In RapidMiner different operators were applied on the data table with all the incidents: for sorting, filtering, splitting, and removing. As a result, this RapidMiner model returns only the so-called request from the customer (i.e. the corresponding event) and the response (i.e. the corresponding event) to this request. The resulting data table contains for each incident with a request and a response, one row with the request-event and its timestamp, the response-event and its timestamp, and the priority, the customer, and the resource that belong to the ticket.

Accordingly, all these views that were created were *exported* to CSV files and XES files. These are the event logs that are the inputs for the fourth stage. This concluded the third stage of PM^2 with the traditional process mining approach. The next section describes how the data processing stage is completed with the redo log process mining approach.

B.3.2 Stage 3: Data Processing with RLPM

The executed steps for the redo log process mining approach in the data processing stage are listed below. Again, during the execution, the required time per step was logged by the researcher of this study. Consequently, the steps were visualized in a process model, which is presented previously in fig. 5.2b. The remainder of this section explains these steps in more detail.

- ✔ Map values of keys to meaningful values;
- ✔ Deduce attribute values for `null`-values;
- ✔ Compose an interpret activity attributes;
- ✔ Filter on specific values of specific attributes;
- ✔ Calculate solve-time for all cases;
- ✔ Calculate response-time for all cases;
- ✔ Export to disc.

The event data created with the redo log process mining approach in the previous stage of PM^2 goes through the same steps as the event data that was created with the TPM approach. However, some additional steps are required to enable filtering for example. Log manipulation activities were performed to ensure that each event in the log had a type-attribute with the correct value. Otherwise, the events from the `ARTICLE` table, which does not have a ticket-type attribute, will remain `null`. The log manipulation mostly includes, conversion of the event log to event data, mapping the existing values from key values to values with an actual meaning, merging of attributes, removing attributes, splitting, filtering and joining the event data.

Moreover, the activity attribute as it is defined in the XES file that was the input for this stage with the RLPM approoach, needed to be re-composed again due to the conversion to a data table. Also, these re-aggregated instances needed to be mapped to meaningful values for humans/workers of the case study. Furthermore, for identifying and selecting the traces with request-events an response-events, a separate RapidMiner model was developed specifically for the RLPM approach.

B.3.3 Stage 3: Data Processing with CPM

The performed steps (i.e. activities) for the data processing stage with the Celonis process mining approach are listed below. These were visualized with a process model, presented previously in fig. 5.2c. The remainder of this section describes the steps in more detail.

- ✔ Filter on trace attribute;
- ✔ Define global variables;
- ✔ Create dimension for event attribute;
- ✔ Create component filter;
- ✔ Create function to calculate solve time;
- ✔ Find tickets with request and response using variables, dimensions and component filters;
- ✔ Create function to calculate response time.

The first step is to *filter the event data on a trace attribute* to select only the 'incident' type tickets. This is performed using the filter bar; by clicking the 'Add New Selection' button and using the 'Attribute Selection' option (see fig. B.9). In CPM4, one works with a canvas on which components for analysis can be placed, like with a dashboard. The filtering is by default applied to all the components in the canvas. So with a few clicks, all the analysis components are updated to visualizing the result for a an updated set of event data.

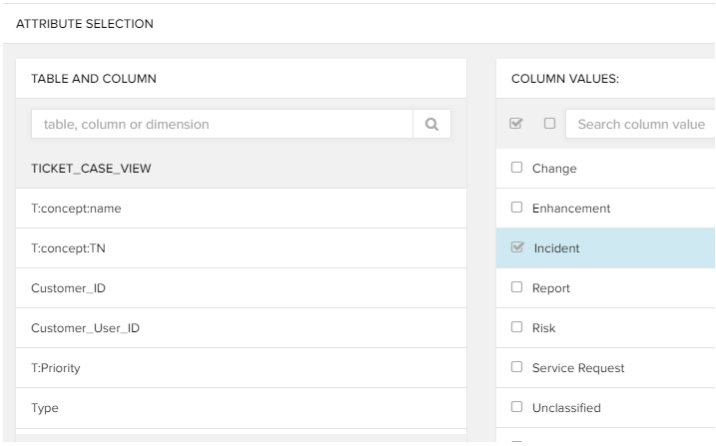


Figure B.9: Filtering the event data on a trace attribute in CPM4.

Next, we *defined global variables*. These variables can be created in a 'global' level in CPM4 and represent any function or constant. In fig. B.10, the declaration of three variables is represented. The variable named `LastPriorityCase` uses the 'pull-up' function, which can be used to find the value of an attribute (e.g. priority, timestamps, etc) of the last event of a trace. Likewise, one could apply the pull up function to retrieve the value of an attribute of the first event of a trace. The variable `AllHrs` represents the service window that was defined for the tickets that have the highest priority, which is 24/7. Likewise, the variable `WorkDays` represents the service window (i.e., 08:00-18:00, from Monday to Friday) for tickets with a lower priority than 'Critical'.

```
LastPriorityCase = PU_LAST("TICKET_CASE_VIEW",
  "TICKET_EVENT_MERGED"."E:Priority")

AllHrs = MONDAY 00:00 - 24:00 TUESDAY 00:00 - 24:00
  WEDNESDAY 00:00 - 24:00 THURSDAY 00:00 - 24:00
  FRIDAY 00:00 - 24:00 SATURDAY 00:00 - 24:00
  SUNDAY 00:00 - 24:00

WorkDays = MONDAY 08:00 - 18:00 TUESDAY 08:00 - 18:00
  WEDNESDAY 08:00 - 18:00 THURSDAY 08:00 - 18:00
  FRIDAY 08:00 - 18:00
```

Figure B.10: Examples of variables defined in CPM4.

Slicing on priority is done *by creating a dimension* for an 'OLAP table'. The event log is then split according to the different values in this dimension. An example a dimension is provided

in fig. B.11. This example uses a defining a `case when else end` function, which is a built-in function in CPM4 that can be applied as an if-statement.

```
CASE
  WHEN (<%=LastPriorityCase%> = '1 very low'
    OR <%=LastPriorityCase%> = '2 low'
    OR <%=LastPriorityCase%> = 'P4 Low') THEN 'Low'
  WHEN (<%=LastPriorityCase%> = '3 normal' OR
    <%=LastPriorityCase%> = 'P3 Medium') THEN 'Medium'
  WHEN (<%=LastPriorityCase%> = 'P2 High' OR
    <%=LastPriorityCase%> = '4 high') THEN 'High'
  WHEN (<%=LastPriorityCase%> = 'P1 Very High' OR
    <%=LastPriorityCase%> = '5 very high') THEN 'Critical'
  WHEN ISNULL(<%=LastPriorityCase%>) = 1 THEN 'NUll'
  ELSE 'Else'
END
```

Figure B.11: Example of a dimension in CPM4, in order to split the log on type of priority.

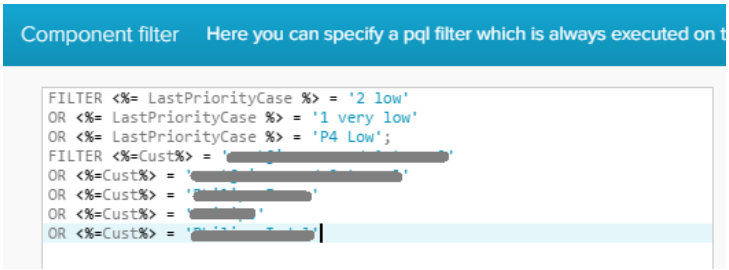


Figure B.12: Creating a component filter with PQL statements in CPM4 (the names of the customers are disclosed).

If one wants to apply a different filter on a specific component for the analysis for example the different customers in the process, this is configurable. To do that, one *adds a component filters* to a particular component. An example of these is given in fig. B.12, where a component is filtered to several customers and priorities.

Lastly, with the use of functions and variables, the solve and response times of a particular set of traces can be calculated. In the example given in fig. B.14 is the function that was created to calculate the *median* solve time. The example uses the built-in `remap timestamps` function of CPM4 in combination with the pull up function. The median solve time in minutes is, the median of the throughput times of all cases. Where for example, the throughput time is calculated by subtracting the amount of minutes since the Unix epoch (during workdays 08:00–18:00) of the last event from the amount of minutes since the Unix epoch (during workdays 08:00–18:00) of the first event. To gain more insights in the process conformance, the average was also calculated using the average function instead of the median function. Since the average and median solve times differed a great amount, it was interesting to visualize the distribution of the solve times of tickets. To that end, the histogram component was created. In fig. B.13 the distribution of the solve time of all incidents is presented.

This calculation is based on the restricted service window since the majority of the tickets are not 'Critical'. Most tickets (over 7130) are solved within the first bucket, that is within the first hour after arrival. Next the chart 'drops' quickly and it become clear that the amount of tickets solved between 1 and 2 hours is 715 (this becomes visible when hovering over the chart). Then it is clear that there is a super long tail of a substantial amount of tickets distributed over the other time buckets, up until a value of 250 workdays. This observation led to the choice of calculating the median response time for analysis.

For the response time a same type of function was created, although it was a little more complex, see fig. B.25. In the end, these derived calculations can not actually be added to the event log as an enrichment. However, these calculations can be used in the components for visualization in CPM4 (e.g. in graphs, OLAP tables, and other components).

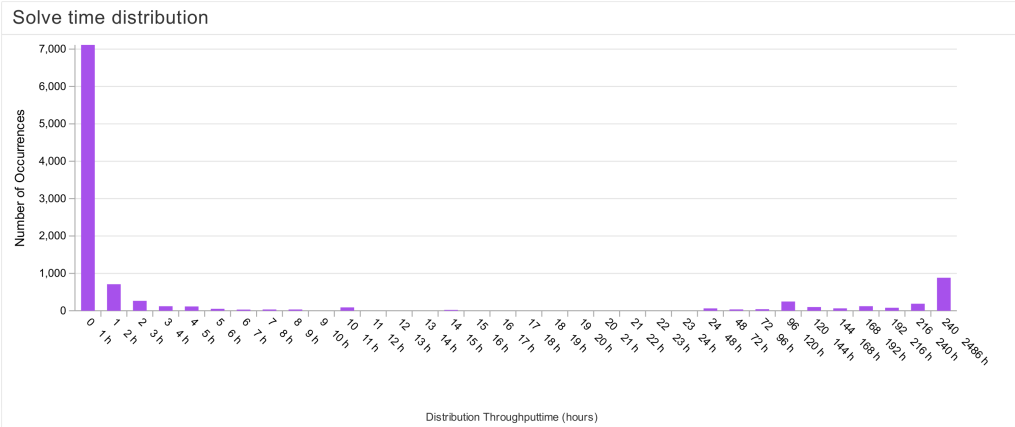


Figure B.13: Histogram in CPM4 plotting the distribution of the solve time of all cases of type 'incident'.

As stated chapter chapter 3, we measure input and output measures per stage of PM^2 to determine the actual efficacy of a process mining approach. The next section shows the resulting measurements for the data processing stage for three process mining approaches under study.

```
MEDIAN(
  CASE
  WHEN <%= LastPriorityCase %> LIKE 'P1 Very High' OR
      <%= LastPriorityCase %> LIKE '5 very high'
    THEN ( REMAP_TIMESTAMPS( PU_LAST(
      "TICKET_CASE_VIEW", "TICKET_EVENT_MERGED"."E:time:timestamp"),
          SECONDS, <%= AllHrs %>)
      -
      REMAP_TIMESTAMPS( PU_FIRST(
        "TICKET_CASE_VIEW", "TICKET_EVENT_MERGED"."E:time:timestamp"),
            SECONDS, <%= AllHrs %>)
      )/1000.0
  ELSE ( REMAP_TIMESTAMPS( PU_LAST(
      "TICKET_CASE_VIEW", "TICKET_EVENT_MERGED"."E:time:timestamp"),
          SECONDS, <%=WorkDays%>)
      -
      REMAP_TIMESTAMPS( PU_FIRST(
        "TICKET_CASE_VIEW", "TICKET_EVENT_MERGED"."E:time:timestamp"),
            SECONDS, <%=WorkDays%>)
      )/1000.0
  END
 )
```

Figure B.14: Function in CPM4 to calculate the median solve time in seconds of a set of traces, depending on value of the priority attribute of the last event in the trace.

B.3.4 Resulting measurements for Stage 3: Data Processing

The table B.5 provides an overview of all the measurements from the execution of stage three with all the different approaches. The previous three sections (B.3.1 till B.3.3) described the steps to complete the data processing stage of PM^2, for three different process mining approaches. According to the *comparison framework for process mining approaches*, which was presented in chapter 3, we use the input and output measures te determine the actual efficiency and actual effectiveness of a process mining approach in the third stage of PM^2. Table B.3 presents the results for the metrics that have been defined for stage 3.

Table B.5: Metrics describing the efficiency and effectiveness of Stage 3 of PM^2.

Metrics to determine actual efficiency (input measures)

Index	Label	Brief description	TPM	RLPM	CPM
3.1	TIME	Time required to complete stage (hours)	31	27	32
3.2	IT-S	Iterations within current stage	1	1	1
3.3	IT-OS	Iterations to previous stages	0	1	0
3.4	DK-O	Domain knowledge objects	27	39	27
3.5	DL-R	Domain knowledge relations	34	48	34
3.6	COMPL	The task complexity	3	2	4

Metrics to determine actual effectiveness (output measures)

Index	Label	Brief description	TPM	RLPM	CPM
3.7	M	Magnitude (# number of events)	124 075	6 372	124 075
3.8	SP	Support (# number of cases)	10 534	268	10 534
3.9	AE	Average number of events per case	11.8	23.8	11.8
3.10	LoD	Level Of Detail	6.5	7.7	6.5
3.11	G	Time Granularity (seconds)	0	0	0
3.12	V	Variance	29	23*	29
3.13	V_{min}	Minimal Variance	2	1	2
3.14	V_{max}	Maximal Variance	21	17	21
3.15	UT	Unique traces in the event log	475	109	477
3.16	ST	Structure	0.68	0.84	0.68
3.17	ST_{min}	Minimal Structure	0.66	0	0.66
3.18	ST_{max}	Maximal Structure	0.92	0.87	0.92
3.19	MA	Mean Affinity	0.65	0.34	0.65
3.20	A_{min}	Minimal Affinity	0	0	0
3.21	A_{max}	Maximal Affinity	1	1	1
3.22	DQ-PR	Problems w.r.t. event log data quality	5	6	5
3.23	MAT	Level of Maturity	⋆⋆⋆	⋆⋆⋆	⋆⋆⋆

*This is 50 if the activity mapping of section 4.3 would be applied.

The results from table B.5 are analyzed in the chapter 3. Among others, this chapter visualizes these results, and compares the differences in the effciency and effectiveness. The next section of this chapter explains the completion of the mining & analysis stage of PM^2, which is the next stage after the data processing stage.

B.4 Completing Stage 4: Mining & Analysis

For each of the derived event logs from the preceding stage, *data processing*, are the inputs for this stage: *mining & analysis*. In this stage, process mining techniques are applied on the event logs. In this stage, the aim is to to answer the BPQS and gain insight into processes performance and compliance. According to PM^2, the activities that are performed in this stage are: process discovery, conformance checking, enhancement, and process analytics [6].

In appendix B.4.1 the execution of these activities is explained for the traditional process mining approach. The method that is used here for the TPM approach also applies for the RLPM approach. The only difference would be the input (i.e. the event log). However, the event log created with the RLPM approach is not sufficiently large (i.e. it contains not enough cases to perform analysis and acquire reliable insights). But, in chapter 4 we already showed that for this case study the RLPM approach can actually retrieve a super set of the event data that the TPM approach retrieved (as long as all the redo logs are archived correctly). Thus we claim that performing the mining & analysis stage with the TPM approach can also represent the RLPM approach in this stage. Next, in appendix B.4.2 the execution of the Celonis process mining approach is demonstrated. Lastly, in appendix B.4.3, the results for the input and output measures defined in the comparison framework are presented.

B.4.1 Stage 4: Mining & Analysis with TPM (equivalent to RLPM)

The TPM approach also represents the RLPM approach since they both use the same tools and techniques. The only difference is in the input (those are the event logs). Since the missing data in the set of archived logs for the RLPM approach, explained in section 4.2 resulted in event logs with not enough cases for reliable analysis, these event logs were not analyzed in the fourth stage. However as was shown in section 4.3, the event log that can be created with the TPM approach, can also be created with the RLPM approach.

The steps that have been executed for the mining & analysis stage with the traditional process mining approach are listed below. The remainder of this section discusses how the analysis was performed for each BPQ.

- ✔ Determine process conformance;
- ✔ Comparing workflows with Process Comparator;
- ✔ Dotted chart analysis;
- ✔ Social network mining.

Process Conformance to the SLAs To answer BPQ 4 we determined the performance of the process with respect to the SLAs. The percentages of tickets and their time boundaries were given in table B.1. To derive the percentages of cases (i.e. tickets) that conform to these predefined SLAs, the solve and response times that have been calculated in the previous stage are now used. This completed by using functions in RapidMiner blocks ('Generate Attributes' operator), see the snippet B.15 for an example. The generated values can subsequently be summed, an instances counted, with an 'Aggregation' block, after with again an 'Generate Attributes' operator is used to calculate the percentages, see function B.16 for an example. The results with respect to conformance to these SLAs are given in table B.6.

From table B.6 we can observe that, on average, over all incidents, none of the SLAs for *solve time* is met. We found that, maybe even most importantly, the tickets with a 'Critical' priority are performing solved much too slow. Only 2 % of the tickets with this priority is solved in time. We can observe that the tickets with the 'High' and 'Medium' priority are solved correctly withing the 90 % boundary (namely, 97 % in 8 hours and and 93 % in 4 days respectively). However, the remaining tickets of these percentages are partly solved more slowly than the 10 % boundary of the SLAs. To be exact, 2 % is solved after 4 days and 6 % is solved after 7 days respectively. For the tickets with a 'Low' priority, 85,5 % is solved within 7 days, while this should have been

```
if(Priority == 1, if(SolveTime >= (8*60*60) , 1, 0),
      if(Priority == 2, if(SolveTime >= (4*10*60*60), 1, 0),
          if(Priority == 3, if(SolveTime >= (7*10*60*60), 1, 0),
              if(Priority == 4, if(SolveTime >= (14*10*60*60), 1, 0),9999)
          )
      )
  )
```

Figure B.15: Function in RapidMiner to find the cases with a solve time above a certain threshold, with for each attribute value of the 'priority' a different threshold.

atleast 90 % according to the service level agreements. Also, more then 12 % of the tickets with the 'Low' priority is solved after 14 days, which is completely out of bounds.

Regarding the *response time*, one can observe that less tickets are analyzed. This is because many tickets are not directly issued by a customer and replied to with an action by a resource. I.e. we found relatively little cases with a request-event and a response-event. For the tickets that do have them, we calculated the percentages of tickets within the boundaries of the SLAs and also the percentages of tickets that violate these. It can be stated that, on average, over all the incidents, over all the priority classes, the SLAs are nearly met, with a difference of 1 % (namely 11 % where it should have been 10 % and 89 % where it should have been 90 %). We found that his is mainly cause be the slow responses on the tickets with the 'High' and 'Medium' priority. Where the one ticket with the 'High' priority was responded to after 2 hours, whereas it should have been responded to within these two hours. According to the SLAs, 90 % of the 39 tickets with the 'Medium' priority should be responded to within one day. However, in reality this was only 67 %. It seems that only the tickets with a 'Low' priority, which is also the majority, is responded to within the boundary of the SLAs, 96 % is responded to in time (i.e. within 2 days).

```
[sum(Not conforming at all)]/[count(Not conforming at all)]*100

[sum(Conforming to the 10% rule)]/[count(Conforming to the 10% rule)]*100

[sum(Conforming to the 90% rule)]/[count(Conforming to the 90% rule)]*100
```

Figure B.16: Function in RapidMiner to compute percentages of cases above and below the certain thresholds.

Table B.6: Process conformance for *solve* and *response times*, answering all questions related to BPQ 4.

Priority	# Tickets	Median Solve Time (H)	Average Solve Time (H)	Violating the boundaries	In 10% boundary	In 90% boundary
Low	7403	0,017	45,63	12,22 %	2,26 %	85,52 %
Medium	1610	0	29,74	5,96 %	0,81 %	93,23 %
High	961	0	7,06	2,08 %	0,62 %	97,29 %
Critical	560	617,19	759,42	97,86 %	0 %	2,14 %
Average		**32,82**	**77,62**	**15 %**	**2 %**	**83 %**

Priority	# Tickets	Median Response Time (min)	Average Response Time (min)	Violating the boundaries	In 90% boundary
Low	135	112	384	4 %	96 %
Medium	39	227	703	33 %	67 %
High	1	850	850	100 %	0 %
Critical	0	-	-	-	-
Average		**142**	**458**	**11 %**	**89 %**

Analyzing Customers For BPQ 5, we looked at differences in the performance of the process when looking at different customers. To gain insights, we first determined which were the biggest clients by aggregating and counting the number of events per customer. Accordingly, we calculated the performance of the process for these three customers. See table B.7 for the resulting figures.

Table B.7: Process conformance for *solve* and *response times*, answering all questions related to BPQ 4.

Priority	# Tickets	Median Solve Time (H)	Average Solve Time (H)	Violating the boundaries	In 10% boundary	In 90% boundary
	Customer 1					
Low	4894	0	65,3	18 %	3 %	79 %
Medium	811	0	50,5	9 %	0 %	91 %
High	852	0	5,1	1 %	0 %	98 %
Critical	546	620,3	776,9	99 %	0 %	1 %
Average		**47,7**	**111,12**	**21 %**	**2 %**	**77 %**
	Customer 2					
Low	880	0	2,2	0 %	0 %	99 %
Medium	262	0	11,1	3 %	1 %	96 %
High	64	0	28,3	8 %	0 %	92 %
Critical	2	118,8	v171,5	100 %	0 %	0 %
Average		**0,2**	**5,8**	**1 %**	**0 %**	**98 %**
	Customer 3					
Low	252	0,02	8,33	2 %	1 %	98 %
Medium	250	0,00	1,81	0 %	0 %	100 %
High	3	12,17	51,19	33 %	33 %	33 %
Average		**0,08**	**5,36**	**1 %**	**1 %**	**98 %**

Priority	# Tickets	Median Response Time (min)	Average Response Time (min)	Violating the boundaries	In 90% boundary
	Customer 1				
Low	20	78	147	0 %	100 %
Medium	1	1	1	0 %	100 %
Average		**74,3**	**140**	**0 %**	**100 %**
	Customer 2				
Low	6	37	126	0 %	100 %
Medium	6	44	58	0 %	100 %
Average		**40,5**	**92**	**0 %**	**100 %**
	Customer 3				
Low	11	73	189	0 %	100 %
Medium	1	2	2	0 %	100 %
Average		**67,1**	**173**	**0 %**	**100 %**

Accordingly, to determine weather customers are treated differently or not, with respect to the workflows (BPQ 5), we applied the Process Comparator plugin in ProM. In fig. 6.15, one can compare the statistical differences in the workflows between customer 1 and 2. For this fig. 6.15, all the significance levels have been set to 1%. Rare transitions, with a transition frequency lower than 4% have been filtered out the model. State border thickness and transition thickness represents frequency. The state set is limited to two of the previous events. Moreover, we configured that the `state` attribute of the events is taken in to the transition system. Observations that can me made are listed below:

- Both logs start with 'NewTicket' and 'CustomerUpdate'. There is little statistical differences in occurrences and time to reach the state '[CustomerUpdate, Newticket]';
- After these two states, the transition system has a split in two branches, which join again

in the state '[Misc, OwnerUpdate]' where both event logs representing the two different customers show no statistical difference in the occurrence not the time to reach this state;

- The 'OwnerUpdate' activity forms the right branch. It is executed frequently in the merged log, but is significantly earlier executed, and more often executed for customer 2.
- Another third activity that is execute less often is the 'SendAgentNotification', which is more often executed for customer 1, however, significantly earlier for customer 2. The 'Lock' activity is often the consecutive activity, this indicates that a resource opens the ticket and has a look at it, which is frequently followed by 'Miscellaneous' and 'OwnerUpdate';
- The states of the transition system that is reached after the execution of this 'SendAgent-Notification', 'Lock' and 'Misc' is intensely blue and purple, which indicates that the event log of Customer 1 has significantly higher occurrences and traces that follow this pattern and these states are often reached earlier with the customer of the first event log;
- The state '[CustomerUpdate, OwnerUpdate]' is executed more often for customer 2;
- The two branches join in the state '[Misc, OwnerUpdate]', after which another split appears. The left, short branch (in red) is executed more frequently with customer 2 (95% of the traces in this log). This is also one of the final states of the transition system, the state of the ticket at this point is 'closed successful';
- The other branch going to the right-side in the model, with the transition 'AddNote', and all the following transitions and states ('Misc', 'Merged', 'TicketLinkAdd', 'StateUpdate', 'Unlock', and 'TicketLinkDelete') all are executed significantly more often for customer 1. The final state of this branch is 'merged'.

Having stated the above, we found that tickets of customer 2 more often are solved to a state which is identified as 'closed succesfully'. Whereas cases of customer 1 are merged more often. Also, it occurs more often for cases regarding customer 1 to display behavior with more activities, where cases involving customer 2 are solved in a lower number of activities. It can be stated that there are definitely clear differences in how these customers are serviced by the company of the case study; in types of activities that are executed, in frequency of occurring and time for reaching the state/executing the transition.

Additionally, comparing customer 2 and 3 (see fig. B.26), result in the observation that the process variants for customer 2 and 3 display minor differences only. These two types of customers are treated more or less the same. This is because all the states have thick lines, which indicates that the states are reached very frequently in the merged log. The same is observed for the transitions. Also, the colors that can be found in the transitions system are not so intense, which indicates that the differences are classified as small effect to medium effectsize ($\pm 0.2 \leq d \leq \pm 0.5$). A different observation is that the transitions 'OwnerUpdate', 'Misc' and 'StateUpdate', and the states '[CustomerUpdate, NewTicket]' and '[CustomerUpdate, Miss, OwnerUpdate]' occurs slightly more often in the event log representing cases for customer 2.

Lastly, comparing the customer 1 with customer 3 (see fig. B.27) resulted in the same type of observations that were made for customer 1 and 2. This could be expected, since customer 2 and 3 have also been compared and no major differences were found. The difference with this comparison and the one between customer 1 and 2 is that the states '[CustomerUpdate, OwnerUpdate]' and '[Misc, StateUpdate]' have a different color, namely that they are less intensely red, which indicates that these states are still occurring more group B, customer 3, but however, the effect is smaller, the same holds for the transitions form and to these states. It should also be mentioned that the there is no significant difference in the time for the execution of these transitions or reaching of states, which in the contrary was a difference in parts of the model comparing customer 1 and 2.

Summarizing, with these annotated transition systems we can discover significant differences between two variants of the same process. The process comparator allows for a comparison on many levels: in this case study, only a few parameters have been investigated. But the Process Comparator can be configured in many ways. Some features that have not been used are for example basing the color interval annotation on the remaining time, or the sojourn time. Another option for configuration is for example to set the state border thickness, and the transition thickness to indicate respectively elapsed time/remaining time/sojourn time, and duration.

Analyzing Resources For answering the following BPQ, "are the resources working as a team on solving the tickets, or are they working separately?", we first want to explore the event data. To start, by determining the three most active resources, we started out by creating a dotted chart, see fig. B.17. This dotted chart was created in RapidMiner (which is also possible in ProM), applied on the complete event log with all cases of type 'incident', the color mapping is based on the value for the resource. In the dotted chart, some resources are highlighted manually with a comment about them. We can for example observe that the most active resource is the Admin OTRS user, identified by number 0 in the graph. The events that are performed automatically by the system are not of interest for the analysis with respect to the resources. From these resources we chose number 1, 2, and, 3, to analyze them in more detail in order to find insights with respect to BPQ 6. The choice for these three was mostly motivated by the fact that these are currently all active resources in the process. Moreover, we expect to find some differences in the performance of these three since one of these employees is new (i.e. number 3).

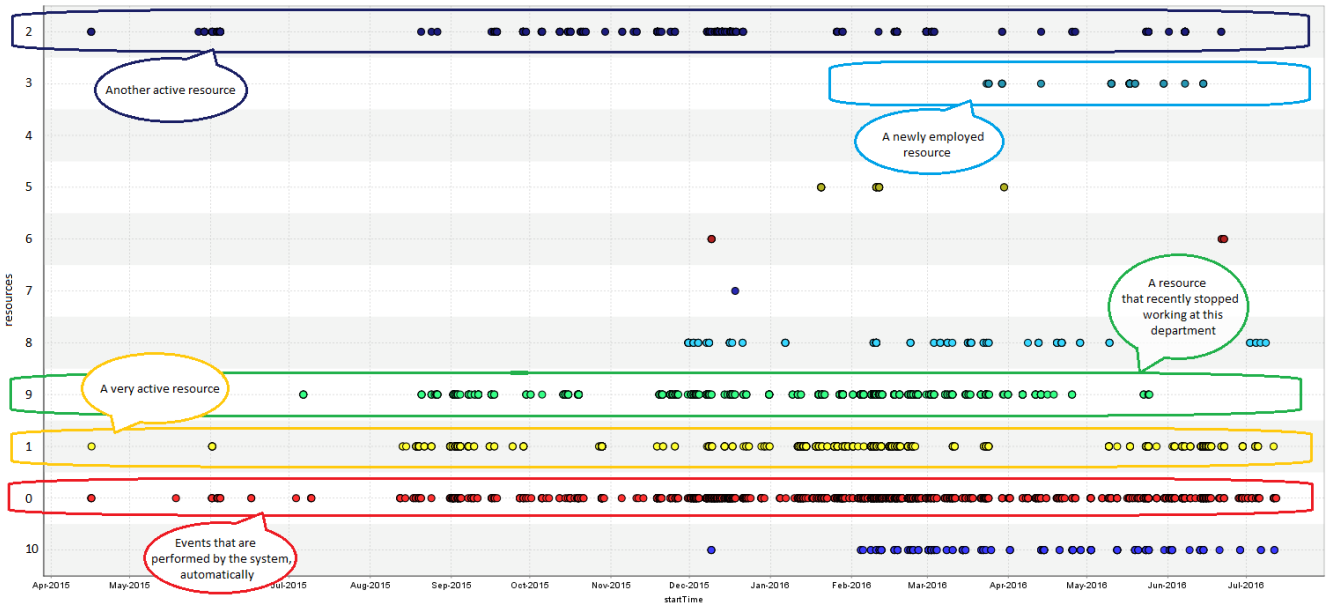


Figure B.17: *Dotted Chart* created in RapidMiner for **TPM**, visualizing the 'activity' of the resources.

In order to compare the performances of the different resources with respect to their average solve and response times of the resources, we calculated again the conformance to the SLAs. In table B.8, one can find the performances of the three most active resources. For this analysis, only the weighted averages over all the priorities are used. We can see from this table that resource number 3 has the best performance. This resource solves the tickets withing the SLAs, whereas the other two resources do not. Also, resource 2 responds the fasted to the customers, 100 % of his responses are on time. The other resources also answer to the customers on time, but only in 90 % ans 91 % of the cases.

Next, mining a social network graph in fig. B.18 provides insights in the question weather workers are working individually or in a team. The plugin 'Social network miner Handover Work' can be used either in ProM or in RapidProm. In this analysis we visualize the handover of work between different resources. Indirect successions are not considered in this analysis, for the fact that here, work is really performed by direct forwarding, via an email or via updating the owner of the ticket. With the tool, one can remove edges with a lower frequency with the slider bar. The idea was to visualize, how work (i.e. tickets) is picked up (and maybe returned to) the Admin OTRS resource, which is the default resource. The result is given in fig. B.18.

It can be noticed that the most active resources are in the center, in yellowish, which are the resources to handle most events (shape is set by ranking; which is set by degree; which is the total

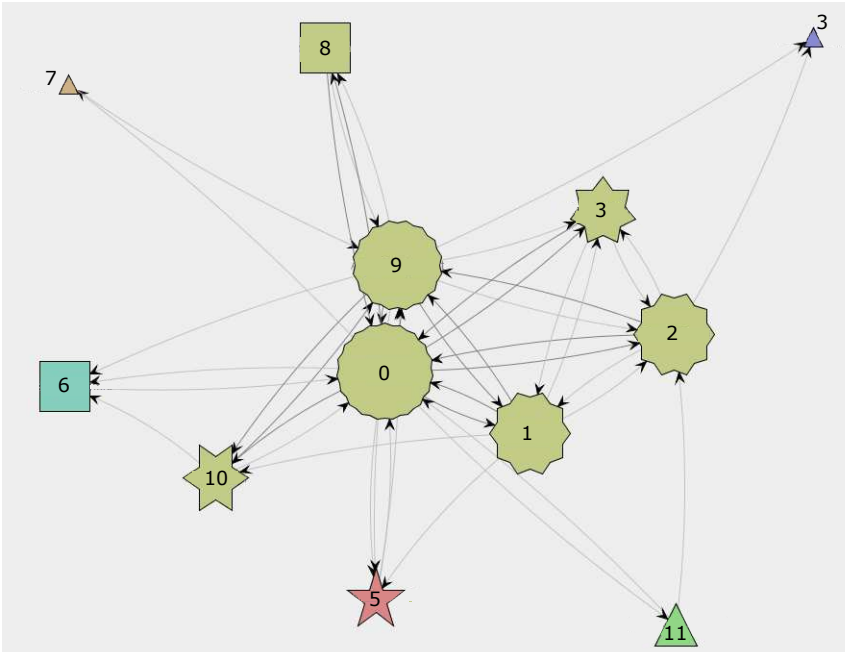


Figure B.18: Social Network, Handover of Work, created in RapidMiner with the TPM approach).

number of vertices adjacent to the vertex). The guiding principle for this layout is a force-directed placement, where edges are interpreted as springs that force neighboring vertices (i.e. resources) to be attracted and all other nodes to be repelled [45]. The connections indicate that some resources have more (or only) incoming nodes. We can observe a very active resource, number 9, begin almost in the center of the network. This person recently stopped working at this department of the company under study (see also fig. B.17). The next three very active resources (number 1, 2, and 3), appear to all handover a lot of the work, indicated by the arrows that connects them. These arrows are often in both directions, meaning that the work is handed over bidirectional. What can also be observed here, is that resource 3 doesn't hand over work to resource 9, but it does happen the other way around. This can be explained by the fact that actually resource 3 can working because resource 9 left. We can observe that there is a lot of 'teamwork' (i.e. handover of work, back and forth) between all the resources in the yellow-ish color. The other resources, with different colors in the model, seem to be outside of this team. These are only receiving work, and less work than the members of the yellowish team. We suppose they are the experts that are needed in rare occasions to solve tickets about very specific problems. This assumption was later-on confirmed by one of the resources. Moreover, it was confirmed that the team of resources that are handling the process under study, actually consists of the yellowish resources in the social network.

Table B.8: Process conformance for solve and response times, regarding BPQ 3.

Resource	# Tickets	Median Solve Time (H)	Average Solve Time (H)	Violating the boundaries	In 10% boundary	In 90% boundary
1	102	14,25	66,89	14 %	15 %	72 %
2	144	26,21	19,77	7 %	4 %	89 %
3	93	10,21	16,46	4 %	1 %	95 %

Resource	# Tickets	Median Response Time (min)	Average Response Time (min)	Violating the boundaries	In 90% boundary
1	34	55,00	169,15	9 %	91%
2	19	9,84	184,79	10 %	90 %
3	4	40,00	265,00	0 %	100 %

This concludes the extraction with the traditional process mining approach. Next, in appendix B.4.2 we explain how the mining & analysis stage is performed with the Celonis process mining approach.

B.4.2 Stage 4: Mining & Analysis with CPM

Process Conformance to the SLAs To find the percentages of tickets with solve and response times within the constraints that have been defined in the SLAs, components with filters are configured to select cases with a certain type of priority. For each text component functions are created. These functions calculate the fraction of cases, with a customer reporting and an agent responding, that have a response time under and above certain thresholds. The example block B.19, counts these specific cases that have a response time between 2 hours (i.e. 120 minutes), and divides that by all these specific tickets. This is a function used on a component that is filtered with a component filter to select only the tickets with a 'High' priority.

```
SUM(
  CASE
    WHEN ((ISNULL(<%= CustomerInc %>) = 0) AND (ISNULL(<%= Response %>) =
        0) AND ((MILLIS_BETWEEN(<%= CustomerInc %>, <%= Response %>)/1000)
        >= 0)) AND
         ((REMAP_TIMESTAMPS(<%= Response %>, MINUTES, <%= WorkDays %>) -
             REMAP_TIMESTAMPS(<%= CustomerInc %>, MINUTES, <%= WorkDays %>))
              < 120)
      THEN 1
    ELSE 0
  END
)
/
(
  (
    SUM(
      CASE
        WHEN ((ISNULL(<%= CustomerInc %>) = 0) AND (ISNULL(<%= Response
            %>) = 0) AND (MILLIS_BETWEEN(<%= CustomerInc %>, <%= Response
            %>) >= 0)) THEN 1
        ELSE 0
      END
    )
  )
/1.00)
```

Figure B.19: Example of function in Celonis to compute percentages of cases above and below the certain thresholds.

Accordingly, for each sub-BPQ of the first BPQ (see also B.1), an *Single KPI* component was defined to answer these questions. These have a color mapping added to them and result in an overview that is convenient for a quick analysis with the eye. The color red indicates that a KPI is trespassed or not met, and a green color indicates all is well within the agreements of the SLAs. Naturally, the calculation of the process conformance requires a different function for both the solve time and the response time.

The resulting percentages of the SLAs are exactly equal to the figures as were presented already in table B.6. This thus also results in the same set of insights that were presented already for the process conformance with the tradtitional process mining approach. The only significat differnce between the TPM and CPM approaches here is the visual representation. For the dashboard that was created in Celonis, see appendix B.10.

Analyzing Customers Comparing the customers with respect to the conformance to the SLAs, was performed just like with the traditional process mining approach. This even so resulted in the same exact figures as were presented already in table B.7. Thus, this confirmed again the insights that were gained in about the customers with the traditional process mining approach. The resulting visualizations for the CPM approach are provided in the appendices B.11.

The *Process Explorer* and the *Variant Explorer* components can be applied in Celonis to mine the from different event logs. Doing this for two event logs representing tickets of different customers was performed to compare the workflows of different event logs. However, the mined process maps turned out to be not convenient for an actual comparison of the work-flows. Several attempts were made to discover different process maps for different selections of data and compare work-flows with the naked eye. Some findings from this analysis can be made about differences in complexity of the different views (event logs), these are based on different amounts of process variants. Concluding we can state that these process discovery techniques are merely useful to give an impression of the work-flow, using the sliders to change the amount of edges and nodes (based on their frequencies), allow the user to generally explore the order of activities in a process. Also, the discovered process maps in CPM4 are not semantically executable. On these process maps, no log can be replayed with CPM4. Therefore no conformance checking can be performed. Concluding we can state that with CPM4, we could not find the answer to the business process question "Do the workflows display significant differences in terms of the duration of the activities, and the frequencies of the activities?".

Analyzing Resources For the performances of the resources we calculated the same percentages, again according to the SLAs, this time per resource. Just like with the process conformance, and the customer analysis, the same results we found as were already found for the traditional process mining approach. Thus, the insights gained from appendix B.4.1 are also valid here. The resulting visualizations for the CPM approach are provided in the appendices, see appendix B.12.

However, in CPM4, it was not possible to create social networks. Thus the question "Are the resources working as a team on solving the tickets, or are they working separately?" can not be answered with Celonis process mining.

B.4.3 Resulting measurements for Stage 4: Mining & Analysis

Table B.9 provides an overview of all the results for the input and output measures of the mining & analysis for the TPM approach and the CPM approach.

Table B.9: Results for metrics describing the effectiveness of Stage 4: Mining & Analysis of PM^2.

Metrics to determine actual efficiency (input measures)

Index	Label	Brief description	TPM	CPM
4.1	COMPL	The task complexity	5	3

Metrics to determine actual effectiveness (output measures)

Index	Label	Brief description	TPM	CPM
4.2	SEM	Executable semantics	YES	NO
4.3	PD	Process discovery techniques	Many	1
4.4	CC	Conformance checking techniques	YES	NO
4.5	ENH	Performance enhancement techniques	YES	Limited

These results are discussed in detail in chapter 6. In the following section, B.4.4, the most striking findings and insights are listed once more. The subsequent section continues to the next stage of PM^2, which is the evaluation stage.

B.4.4 Findings for the Case Study

Findings based on the analysis that has been executed, *irrespective* of the chosen approach are listed in this section.

Findings with respect to BPQ1 (see BPQ 4)

- Only 2% of the tickets with a critical are priority solved within 2 hours, this should be 90% according to the SLAs. There are 557 incidents with this priority.
- The remaining 98% of these tickets is even solved only after 8 hours, which is trespassing the maximum time that is set for solving such a ticket according to the SLAs.
- Tickets with a high and medium priority are generally solved within time, 98% and 93% is respectively solved within 8 hours and 4 days. However, 2% (high priority) and 6% (medium priority) are trespass the maximum solve times of 4 and 7 days.
- 86% of the tickets with a low priority are solved within 2 hours, this is close to the 90% goal from the SLAs. However, 12% of these tickets with this priority are solved in over 14 days, which is not agreed upon according to the SLAs.
- Only half of the tickets with a critical or high priority are responded to in time, this should be 90% according to the SLAs. However, there are respectively only 2 and 9 tickets that conform to the requirements to really measure a response. It is thus questionable whether these findings are representative for all the critical tickets.
- 72 percent of the tickets with a medium priority perform with a response time within one day, this should be 90% according to the SLAs. The number of tickets that are taken into account is 305, this is a much more solid basis for drawing conclusions about the performance of the support and customer service.
- The tickets with a low priority are, on the other hand, being responded to within the boundaries stated in the SLAs. This is the largest set of cases that have a message from a customer followed by a message from the agent, namely 387 tickets.

Findings with respect to BPQ2 (see BPQ 5)

The objective observations that can be made from the analysis:

- Customer number 3 has no events in the log with a priority that is critical (the highest priority), also this is the customer with the least tickets in the log.
- The KPIs are performing much better with the unfiltered log. That is since these automatically generated tickets from zabbix are solved much easier and faster. Tickets generated by people, i.e. employees from the customers, are usually different types of problems/issues they are experiencing. These types of tickets are not standardized.
- In the filtered situation, percentage wise, the KPIs are performing better with customer 2 and 3.
- Customer 2 and 3 have less case variants than customer 1: 57 and 47 against 210 variants respectively. After an iteration back to stage three, performing some filtering by removing 'zabbix', 'zabbix zabbix@xxxxx.nl', 'no-reply-cloudcontrol@xxxxx.nl' as customer-users, this is 25 and 28 variants against 35. Customer 1 seems to be a litte more complex, looking at these figures comparing the amount of case variants.
- The difference in complexity of the different logs is also visible when comparing the process models. The one from the log of customer 1 seems to be more complex as it has more edges and nodes.

Findings with respect to BPQ3 (see BPQ 6)

What we can derive from these figures are the following observations:

- Resource 1 is the most responding resource, however, this resources is (together with resource 2) the slowest 'responder'.
- The fastest 'responder' is resource 3.

- Only one ticket with a high priority seems to be issued by a customer and then answered by a resource (D), the others are of a lower priority.
- Four resources are mainly responding to tickets
- Resource I started responding to tickets in march and was gradually responding to more of them for three months, after which no responses were logged from this resource anymore.
- More resources seem to be involved in the process over time, for example resource 8 and 3 started responding to tickets December 2015 and February 2016 respectively.
- Some resources have been involved for a longer period of time already, 1 and 2 respond to tickets in every month.
- Resource 9 has been very active since the start, but stopped responding to tickets in April 2016.
- From the Social Network Graph it can be observed that a cluster of workers, in the yellowish color, are working together as team, handing over cases, and solving them together. However, now and then, the help of some more 'distant' workers is required to solve a ticket. Such tickets are handed over to these less frequently involved resources, the experts.

B.5 Completing Stage 5: Evaluation

The findings from the previous stage, answering the BPQs related to performance and compliance goals of the process, are evaluated in this stage to relate them to improvement ideas that achieve the process mining project's goals. This stage is not executed differently depending on the chosen process mining approach, and therefore this stage is executed only once and out of scope for the comparison framework.

Evaluating the findings with respect to the response times, it might be relevant to consider the idea that the logging of the tickets could be ambiguous. For example, only tickets that are of type 'incident' are used for analysis. However, a large amount of tickets is of type 'Unclassified'. These are 6780 cases to be exact, which are now completely ignored, while these could even so be incidents. Another attribute that might not be logged correctly in the PAIS is the ticket-'state', which could possibly be used as an event-classifier. This attribute is changed in value for a ticket by certain events, by the system, however, the logic behind this is unclear; sometimes the status remains merged, whilst a case is solved. And, although this sounds suspicious, there is no recording of any case 'closed unsuccessful', the cases are always 'closed successful'.

Activities that are executed very frequently for all three customers of interest are 'NewTicket', 'CustomerUpdate', 'OwnerUpdate', 'Misc', and 'StateUpdate'. This turns out to be an automatic process triggered by a an enterprise-class open source distributed monitoring solution called zabbix. This tool is used for all the customers of the case. The impact of these trace variants on the reliability of the results is unclear. This means that the cases with this behavior should maybe not be analyzed separately.

Another side note that should be given is that the *response time* is not an accurately measurable dimension. All tickets that are issued, are responded to via an auto-reply; in the same minute. This reply is not considered a real, actual response for the customer. Therefore, complicated selections had to be done, each ticket, having an event that is the receiving of a message (phone or email) send form a customer, and a later in the same case, also and event that stands for the sending of a message (phone or email) send by an agent. It turned out that not so many cases actually have these events in that order. In many cases, no customer issued a ticket, but the ticket was issued through zabbix. Moreover, in other cases, where a ticket was clearly issued be a customer, the agent made an internal note, and accordingly the closed the ticket, the customer was notified by an automatically generated message (mail) that the ticket was closed successfully. Since an internal-note is in the form of free-text, it can mean anything. No clear, standardized way of working (which is reflected by the log), has been discovered; therefore, the derived performance indicators for the response times rely on a very poor sample size. The results with respect to the response time-performances should handled with this information in mind (especially for the RLPM approach since these results rely on the smallest event log).

B.6 Queries

This appendix shows more details about the contents of the case study performed in appendix B. In fig. B.20, one can find the query that was executed in the extraction stage with the TPM and CPM approaches.

```sql
CREATE OR REPLACE VIEW OTRS_OWN.TICKET_EVENT_VIEW
AS
SELECT
  th.ID                                  "E:time:sorting",
  th.CHANGE_TIME                         "E:time:timestamp",
  th.TICKET_ID                           "T:concept:name",
  tt.TN                                  "T:concept:TN",
  tht.NAME                               "E:concept:type",
  ts.NAME                                "E:concept:state",
  (tht.NAME || '-' || ts.NAME )          "E:concept:combined",
  (u.FIRST_NAME || ' ' || u.LAST_NAME)   "E:org:resource",
  tp.NAME                                "E:Priority",
  NVL(ta.ArtType, 'Undefined')           "E:article:type",
  NVL(ta.ArtSenderType, 'Undefined')     "E:article:sender",
  NVL(ta.ArtTo, 'Undefined')             "E:article:to",
  NVL(ta.ArtFrom, 'Undefined')           "E:article:from"
FROM
  OTRS_OWN.TICKET_HISTORY th
LEFT OUTER JOIN
  OTRS_OWN.TICKET tt
  ON th.TICKET_ID = tt.ID
LEFT OUTER JOIN
  OTRS_OWN.TICKET_HISTORY_TYPE tht
  ON th.HISTORY_TYPE_ID = tht.ID
LEFT OUTER JOIN
  OTRS_OWN.TICKET_STATE ts
  ON th.STATE_ID = ts.ID
LEFT OUTER JOIN
  OTRS_OWN.USERS u
  ON th.CREATE_BY = u.ID
LEFT OUTER JOIN
  OTRS_OWN.TICKET_PRIORITY tp
  ON th.PRIORITY_ID = tp.ID
LEFT OUTER JOIN
  (SELECT
    art.ID Aid,
    artt.NAME ArtType,
    arts.NAME ArtSenderType,
    art.A_TO ArtTo,
    art.A_FROM ArtFrom
  FROM
    OTRS_OWN.ARTICLE art
  LEFT OUTER JOIN
    OTRS_OWN.ARTICLE_TYPE artt
    ON art.ARTICLE_TYPE_ID = artt.ID
  LEFT OUTER JOIN
    OTRS_OWN.ARTICLE_SENDER_TYPE arts
    ON art.ARTICLE_SENDER_TYPE_ID = arts.ID) ta
  ON th.ARTICLE_ID = ta.Aid;
```

Figure B.20: Query for selecting data regarding the events for the TPM and CPM approaches.

```sql
CREATE OR REPLACE VIEW OTRS_OWN.TICKET_CASE_VIEW
AS
SELECT
  ti.ID                                    "T:concept:name",
  ti.TN                                    "T:concept:TN",
  nvl(ti.CUSTOMER_ID, 'Undefined')         "Customer_ID",
  nvl(ti.CUSTOMER_USER_ID, 'Undefined')    "Customer_User_ID",
  tp.NAME                                  "T:Priority",
  nvl(tty.NAME,'Undefined')                "Type"
FROM
  OTRS_OWN.TICKET ti
LEFT OUTER JOIN
  OTRS_OWN.TICKET_PRIORITY tp
  ON ti.TICKET_PRIORITY_ID = tp.ID
LEFT OUTER JOIN
  OTRS_OWN.TICKET_TYPE tty
  ON ti.TYPE_ID = tty.ID
WHERE
  ti.ID IN
  (SELECT tev."T:concept:name" FROM OTRS_OWN.TICKET_EVENT_MERGED tev);
```

Figure B.21: Query for selecting data regarding the cases for the TPM and CPM approaches.

```sql
CREATE TABLE OTRS_OWN.TICKET_EVENT_MERGED
(
  "E:time:sorting"  NUMBER(20,0) PRIMARY KEY,
  "E:time:timestamp" DATE,
   "T:concept:name" NUMBER(20,0),
  "T:concept:TN" VARCHAR2(50 BYTE),
  "E:concept:type" VARCHAR2(200 BYTE),
  "E:concept:state" VARCHAR2(200 BYTE),
  "E:concept:combined" VARCHAR2(401 BYTE),
  "E:org:resource" VARCHAR2(201 BYTE),
  "E:Priority" VARCHAR2(200 BYTE),
  "E:article:type" VARCHAR2(200 BYTE),
  "E:article:sender" VARCHAR2(200 BYTE),
  "E:article:to" VARCHAR2(3800 BYTE),
  "E:article:from" VARCHAR2(3800 BYTE)
)
TABLESPACE OTRS
STORAGE(INITIAL 29696K);
```

Figure B.22: Query for creating a new table, so the PL/SQL script can store the merged events.

```
CREATE OR REPLACE PROCEDURE TICKETMERGE_RELABEL AS
  start_Message VARCHAR2(100) :=
    'The TicketMerge script was started at ' || TO_CHAR (sysdate, 'DD.MM.YYYY HH24:MI:SS');
  emptyOrNot NUMBER := NULL;
  finshed_Message VARCHAR2(200) := NULL;

BEGIN
  dbms_output.put_line(start_Message);

/*Start with empty tables*/
  EXECUTE IMMEDIATE '
    TRUNCATE TABLE OTRS_OWN.TMP'; /*Empty the table with the list with roots and nodes of these roots*/
  EXECUTE IMMEDIATE '
    TRUNCATE TABLE OTRS_OWN.TMP_2'; /*Empty the table with the list with roots and nodes of these roots*/
  EXECUTE IMMEDIATE '
    TRUNCATE TABLE OTRS_OWN.TICKET_EVENT_MERGED'; /*Empty table to be sure to start from a clean sleat*/
  EXECUTE IMMEDIATE '
    alter session set nls_date_format = "DD.MM.YYYY HH24:MI:SS"'; /*Change the date formatting*/
/*Create the initial TMP table with all instances that are parents/sources and that are not childeren!*/
  EXECUTE IMMEDIATE '
    INSERT INTO OTRS_OWN.TMP
    SELECT
     OTRS_OWN.LINK_RELATION.SOURCE_KEY,
     OTRS_OWN.LINK_RELATION.TARGET_KEY,
     OTRS_OWN.LINK_RELATION.CREATE_TIME
    FROM
     OTRS_OWN.LINK_RELATION
    WHERE LINK_RELATION.SOURCE_KEY NOT IN
      (SELECT OTRS_OWN.LINK_RELATION.TARGET_KEY FROM OTRS_OWN.LINK_RELATION)';
/*Select the count into the variable emptyOrNot*/
  EXECUTE IMMEDIATE '
  SELECT COUNT(*) FROM OTRS_OWN.TMP' INTO emptyOrNot;
  dbms_output.put_line('The number of items in OTRS_OWN.TMP ' || emptyOrNot);
/*Fill the new OTRS_OWN.TICKET_EVENT_MERGED table with for now two selections*/
  EXECUTE IMMEDIATE '
    INSERT INTO OTRS_OWN.TICKET_EVENT_MERGED
      SELECT *
      FROM OTRS_OWN.TICKET_EVENT_VIEW';

  WHILE (emptyOrNot != 0)
  LOOP
    begin
/*update all the values of the OTRS_OWN.TICKET_EVENT_MERGED table that contain the children of the
OTRS_OWN.TMP table to the parents value of these children. Moreover, not that an arbitrary choice is made to
select the 1st link if child has (parents>1), select most recent link with lowest ID when equal date equal*/
    EXECUTE IMMEDIATE '
      UPDATE OTRS_OWN.TICKET_EVENT_MERGED t1
      SET t1."T:concept:name" =
      (SELECT t2.SOURCES
      FROM OTRS_OWN.TMP t2
      WHERE ((t1."T:concept:name" = t2.TARGETS) AND (t2.CREATE_TIMES = (SELECT * FROM
        (SELECT MAX (t2.CREATE_TIMES) FROM OTRS_OWN.TMP t2 WHERE t1."T:concept:name"= t2.TARGETS) WHERE ROWNUM
            <= 1))))
          WHERE EXISTS
      (SELECT 1 FROM OTRS_OWN.TMP t2 WHERE t1."T:concept:name" = t2.TARGETS)';
/*Create a new list of new parents/source, these are the childeren of OTRS_OWN.TMP, with and their targets.
Thus, join is on targets/children in OTRS_OWN.TMP that are a parent/source themselves. And select the parent
value of these matches and their children one level deeper*/
    EXECUTE IMMEDIATE '
    INSERT INTO OTRS_OWN.TMP_2
      SELECT
       OTRS_OWN.TMP.SOURCES,
       OTRS_OWN.LINK_RELATION.TARGET_KEY,
       OTRS_OWN.LINK_RELATION.CREATE_TIME
      FROM
        OTRS_OWN.LINK_RELATION
      INNER JOIN
        OTRS_OWN.TMP
      ON
        LINK_RELATION.SOURCE_KEY = OTRS_OWN.TMP.TARGETS';
/*Clear the table OTRS_OWN.TMP*/
    EXECUTE IMMEDIATE '
      TRUNCATE TABLE OTRS_OWN.TMP';
/*Copy all the files of OTRS_OWN.TMP_2 into OTRS_OWN.TMP again*/
    EXECUTE IMMEDIATE '
      INSERT INTO OTRS_OWN.TMP
        SELECT * FROM  OTRS_OWN.TMP_2';
/*Clear the table OTRS_OWN.TMP_2 again*/
    EXECUTE IMMEDIATE '
      TRUNCATE TABLE OTRS_OWN.TMP_2';
/*Clear the variable @emptyOrNot
    EXECUTE IMMEDIATE '
      UNDEFINE emptyOrNot';*/
/* reset the value of emptyOrNot*/
    EXECUTE IMMEDIATE '
      SELECT COUNT(*) FROM OTRS_OWN.TMP' INTO emptyOrNot;
    dbms_output.put_line(emptyOrNot || ' is the amount of the next values the will be relabeld');
    end;
  END LOOP;
/*dbms_output.put_line(TO_CHAR (sysdate, "DD.MM.YYYY HH24:MI:SS"));*/
END;
```

Figure B.23: PL/SQL script that is used to merge the cases for the TPM and CPM approaches.

B.7 Stage 2: Extraction, RLPM

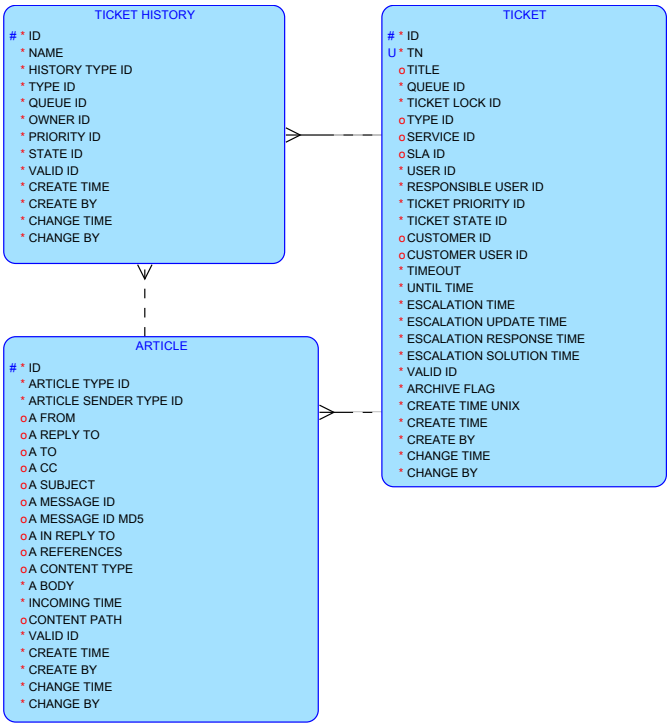


Figure B.24: Scope for selection data: extraction for the RLPM approach is performed for these three tables.

B.8 Stage 3: Data Processing, CPM

```
MEDIAN(
    CASE
        WHEN ((ISNULL(<%= CustomerInc %>) = 0) AND (ISNULL(<%= Response %>) = 0) AND (MILLIS_BETWEEN(<%=
             CustomerInc %>, <%= Response %>) >= 0)
            AND (<%=RequestPriorityCase%> = 'P1 Very High' OR <%=RequestPriorityCase%> = '5 very high'))
            THEN (REMAP_TIMESTAMPS(<%= Response %>, MINUTES, <%= AllHrs %>) - REMAP_TIMESTAMPS(<%= CustomerInc
                %>, MINUTES, <%= AllHrs %>))
        WHEN ((ISNULL(<%= CustomerInc %>) = 0) AND (ISNULL(<%= Response %>) = 0) AND (MILLIS_BETWEEN(<%=
             CustomerInc %>, <%= Response %>) >= 0))
            THEN
            (REMAP_TIMESTAMPS(<%= Response %>, MINUTES, <%= WorkDays %>) - REMAP_TIMESTAMPS(<%= CustomerInc %>,
                  MINUTES, <%= WorkDays %>))
        ELSE NULL
    END
)

CustomerInc = PU_FIRST("TICKET_CASE_VIEW", "TICKET_EVENT_MERGED"."E:time:timestamp",
  ("TICKET_EVENT_MERGED"."E:article:type" = 'email-external' OR
    "TICKET_EVENT_MERGED"."E:article:type" = 'phone')
  AND "TICKET_EVENT_MERGED"."E:article:sender" = 'customer')

Response = PU_FIRST("TICKET_CASE_VIEW", "TICKET_EVENT_MERGED"."E:time:timestamp",
  ("TICKET_EVENT_MERGED"."E:article:type" = 'email-external' OR
  "TICKET_EVENT_MERGED"."E:article:type" = 'phone') AND
  "TICKET_EVENT_MERGED"."E:article:sender" = 'agent')
```

Figure B.25: Function in CPM4 to calculate the median response time in minutes of a set of traces depending on the priority.

B.9 Stage 4: Mining & Analysis, TPM - BPQ 2

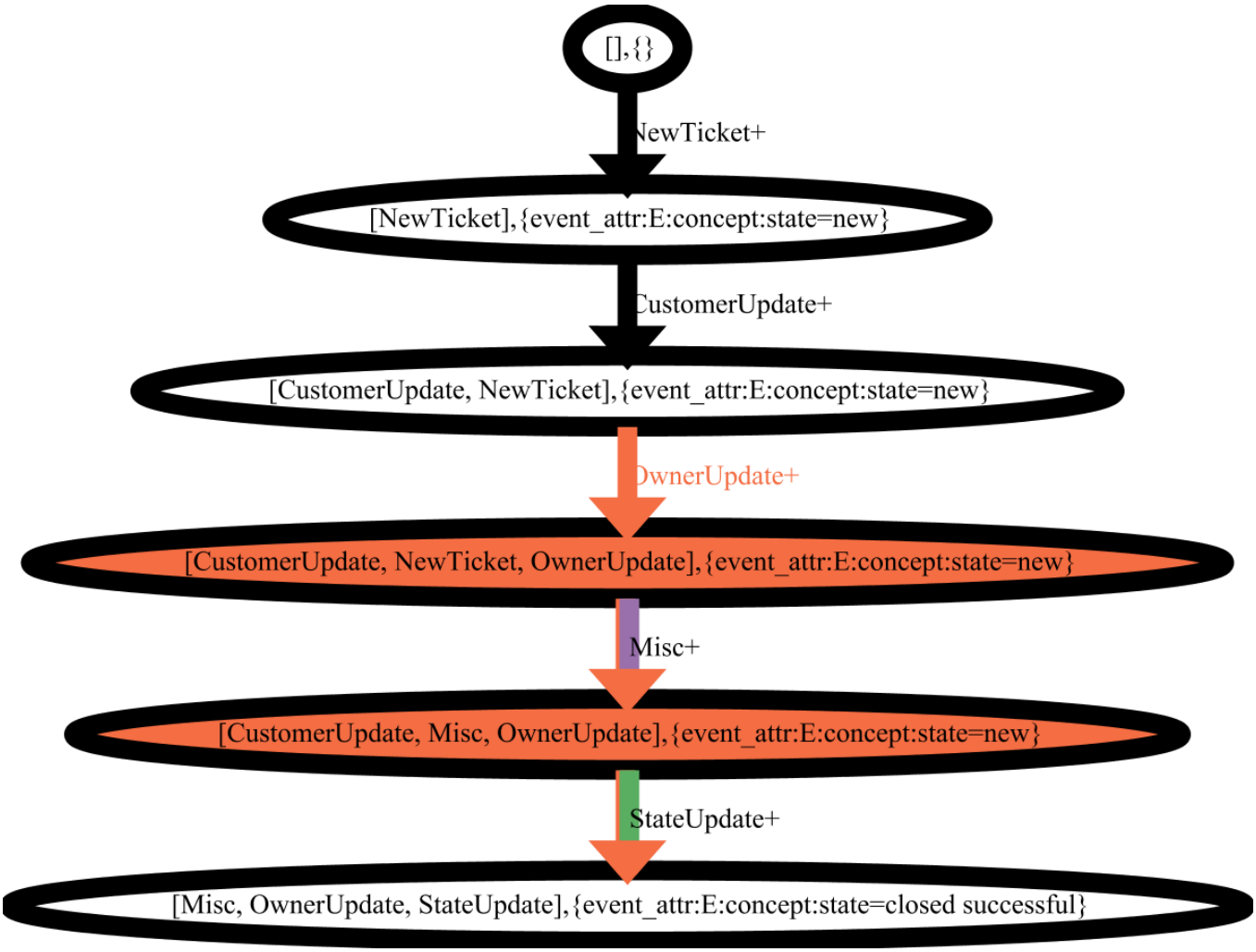


Figure B.26: Customer 2 and 3 with the Process Comparator.

For this graph, all the significance levels have been set to 1%. Rare transitions, with a transition frequency lower than 5% have been filtered out the visualization of the transition system. State border thickness and transition thickness represents frequency. The state set is limited to 3 of the previous events. The configuration includes the event name of the backward key.

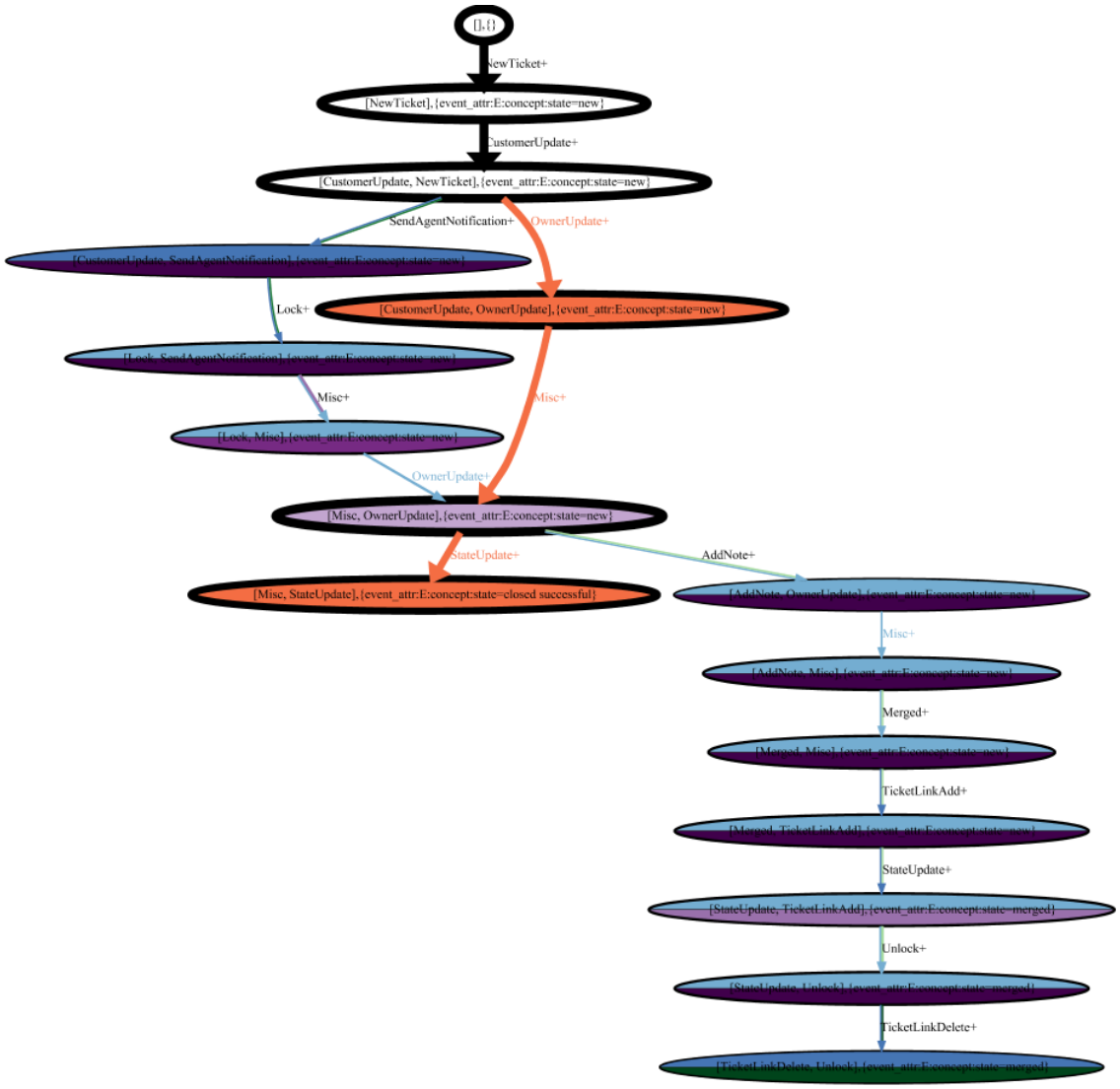


Figure B.27: Customer 1 and 3 with the Process Comparator.

For this graph, all the significance levels have been set to 1%. Rare transitions, with a transition frequency lower than 5% have been filtered out the visualization of the transition system. State border thickness and transition thickness represents frequency. The state set is limited to 2 of the previous events. The configuration includes the event name of the backward key.

B.10 Dashboards created with CPM - BPQ 1

Select ticket type

SOLVE TIMES FOR INCIDENTS

Select customer of interest

Metrics

Priority	Median Solve Time	Avg ST s.w. workdays	# cases	Avg # act/case
Critical	2,221,878 s	2,733,898 s	560	19
High	1 s	25,404 s	961	9
Medium	0 s	107,094 s	1610	13
Low	61 s	164,298 s	7403	11

Critical Priority tickets	High Priority tickets	Medium Priority tickets	Low Priority tickets
Median Throughput time	Median Throughput time	Median Throughput time	Median Throughput time
617 hours	0 minutes	0 minutes	1 minutes
Process start → Process end	Process start → Process end	Process start → Process end	Process start → Process end

Critical	High	Medium	Low
< 4 h	< 8 h	< 4 days	< 7 days
2% / 90 %	97% / 90%	93% / 90%	86% / 90%
>= 4 h and < 8 h	>= 8 h and < 4 days	>= 4 days and < 7 days	>=7 days and <14 days
0% / 10 %	1% / 10%	1% / 10%	2% / 10%
>= 8 h	>4 days	>7 days	>= 14 days
98%	2%	6%	12%

KEY PERFORMANCE INDICATORS

Figure B.28: Analysis of the solve times, BPQ 1, for the Case Study with Celonis Process Mining.

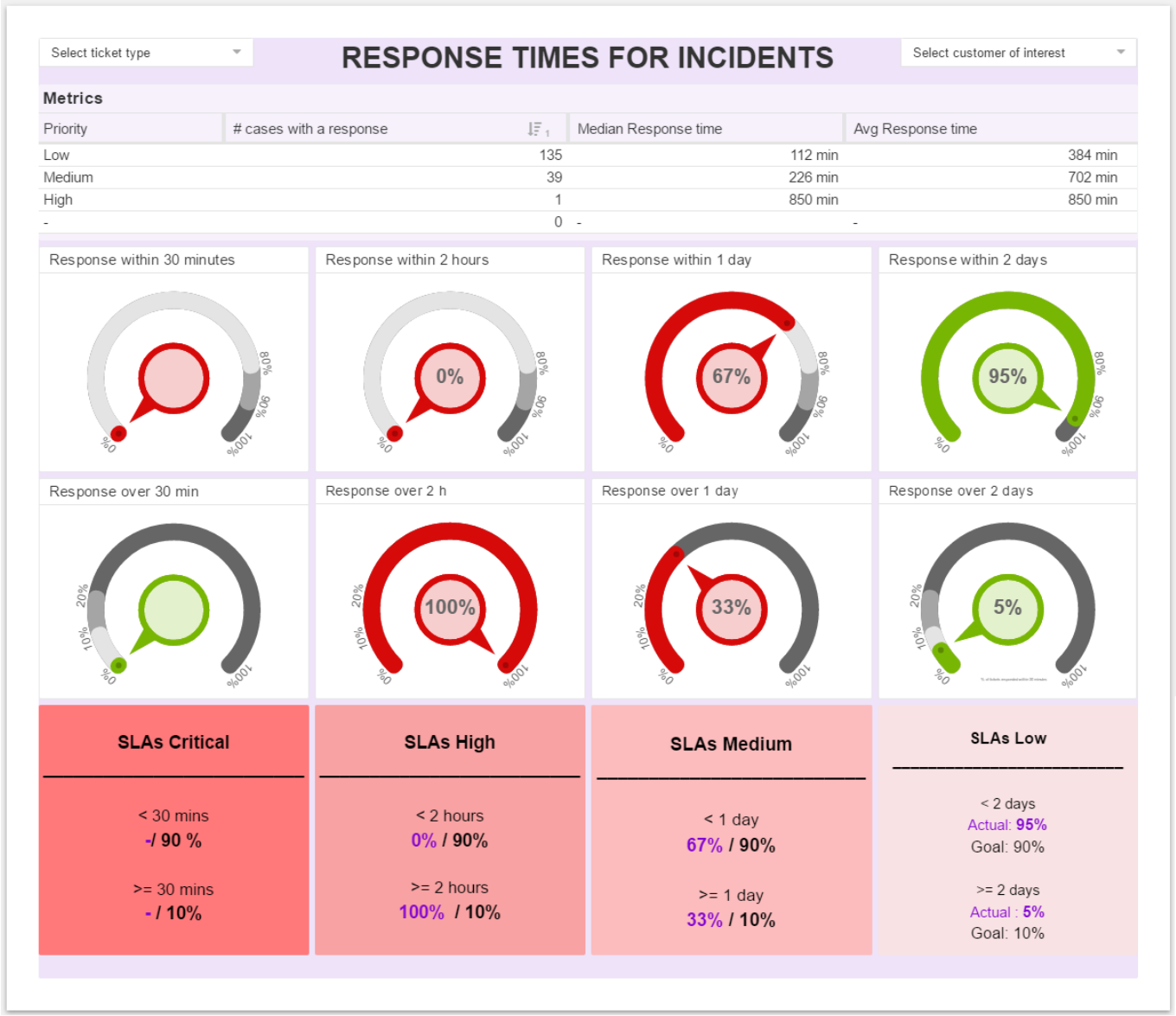


Figure B.29: Analysis of response times, BPQ 1, for the Case Study with Celonis Process Mining.

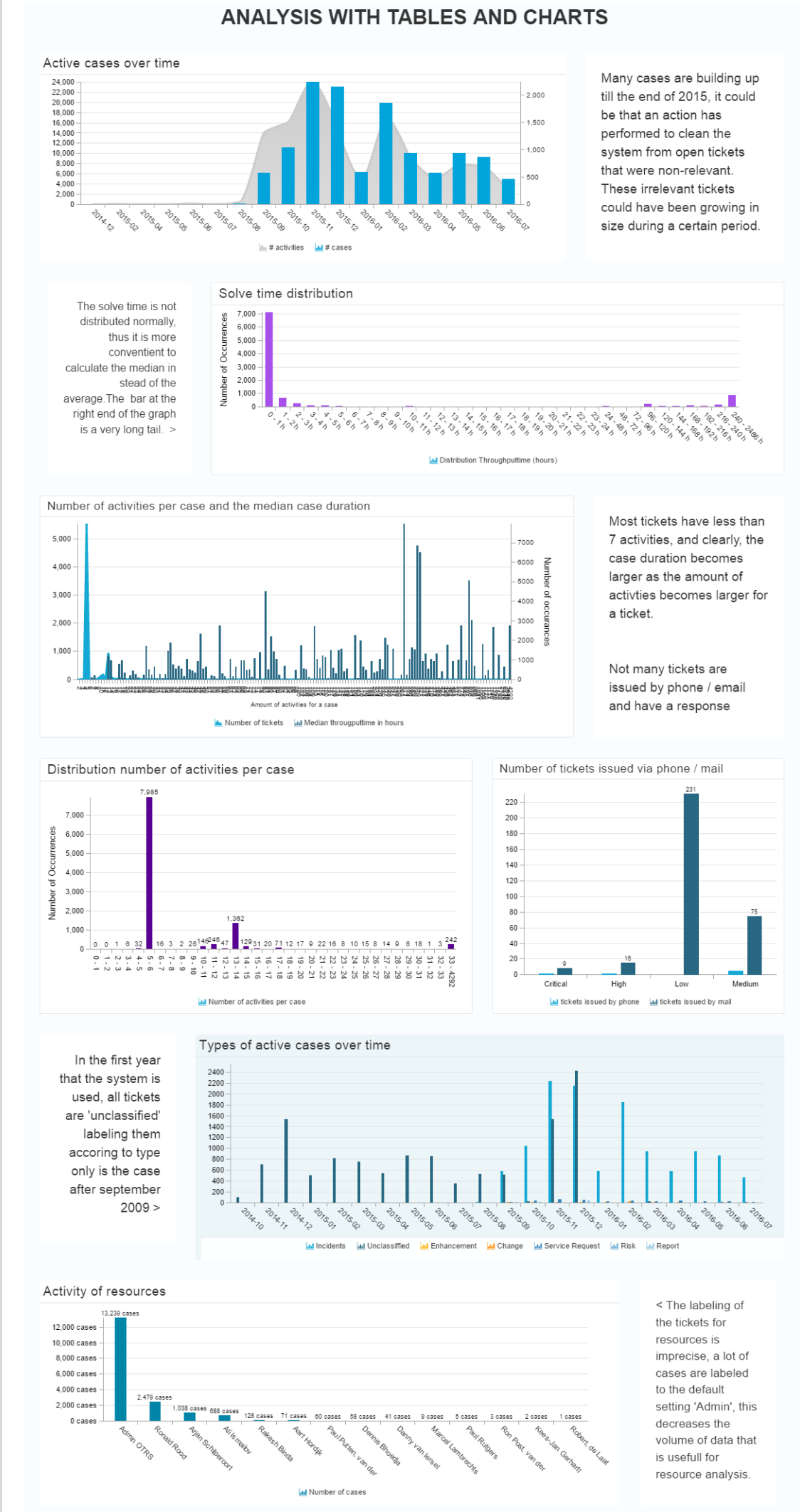


Figure B.30: Arbitrary set of Visual Analysis, for the Case Study in Celonis Process Mining.

B.11 Dashboards created with CPM - BPQ 2

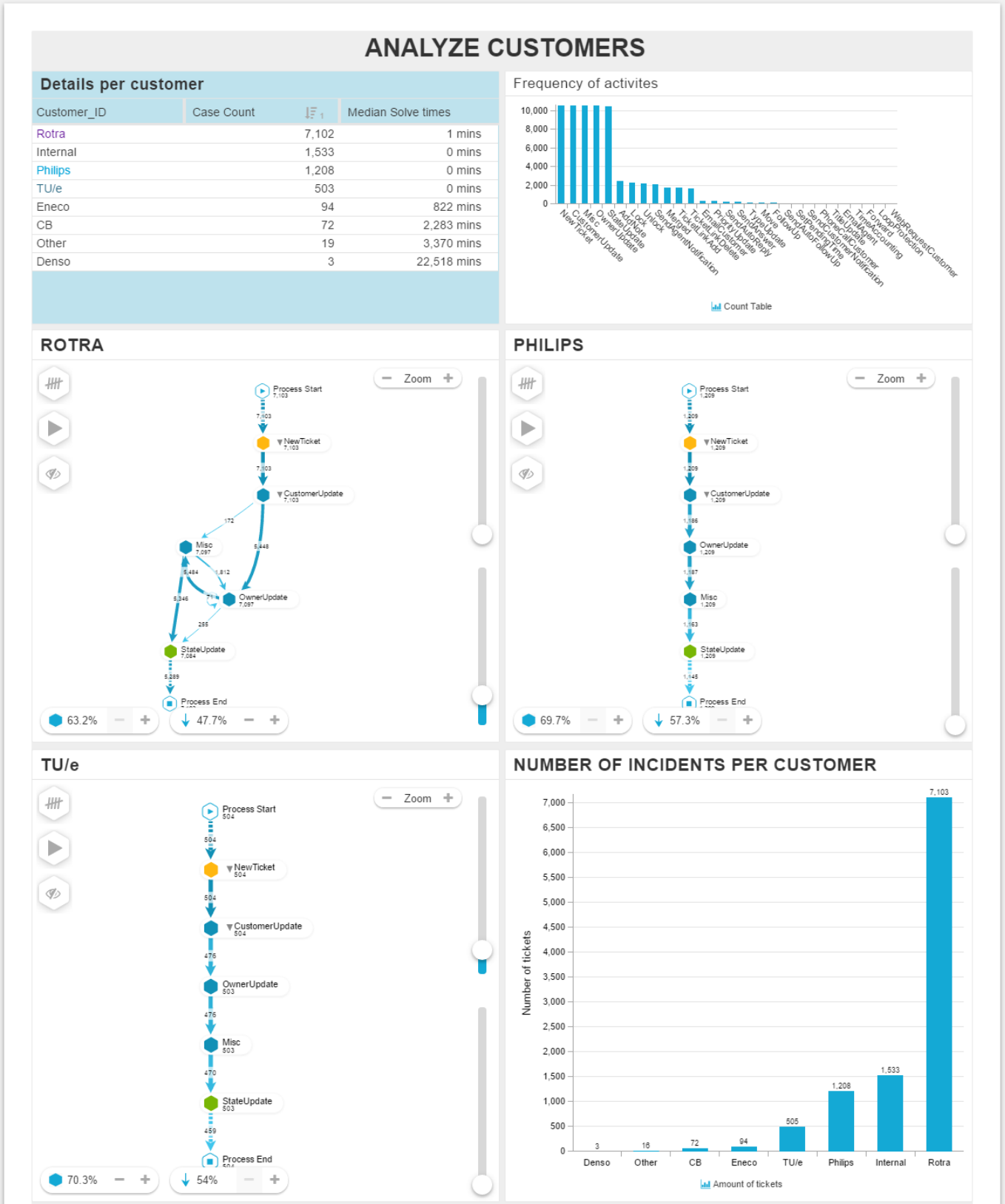
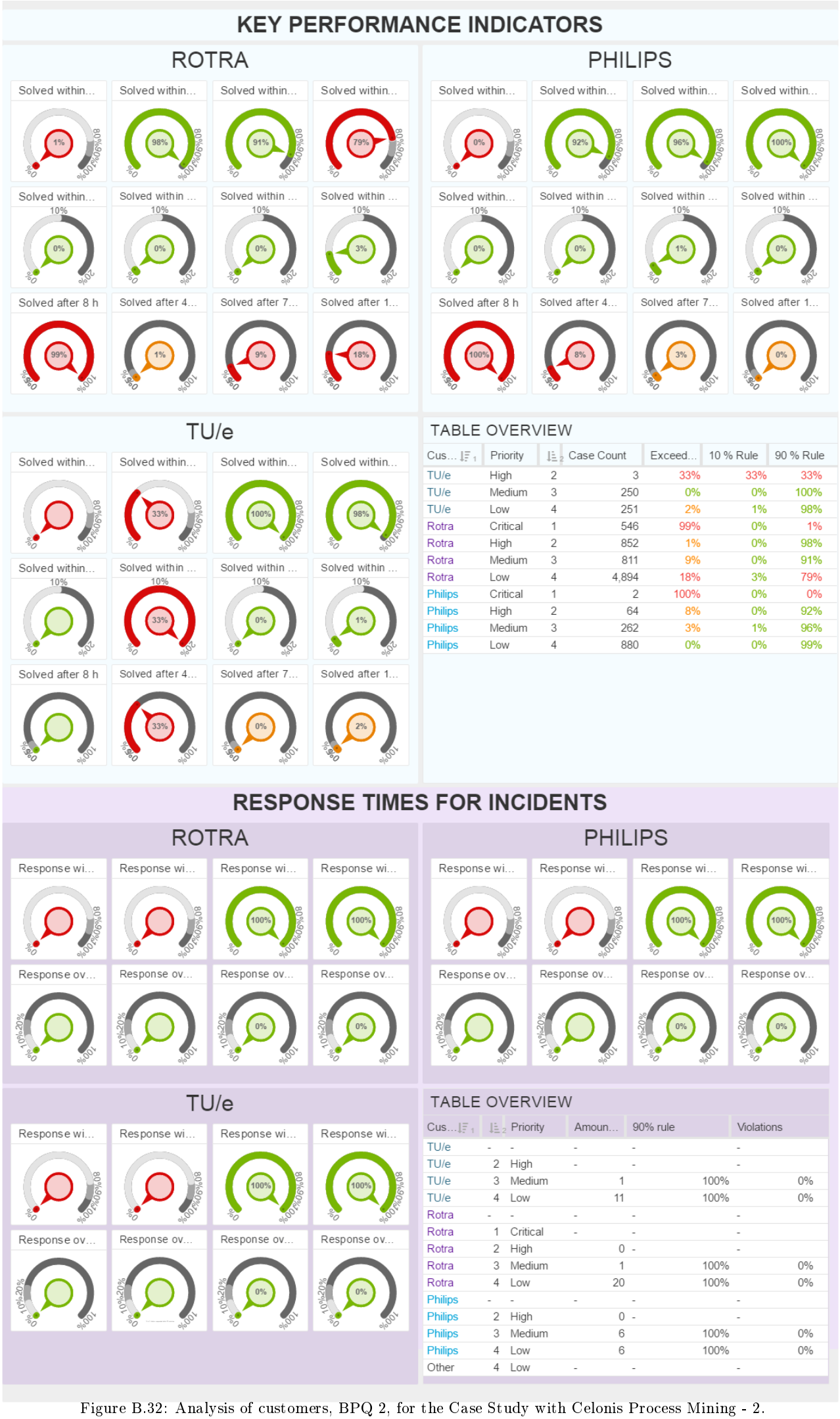


Figure B.31: Analysis of customers, BPQ 2, for the Case Study with Celonis Process Mining - 1.



Cus...	Priority	Case Count	Exceed...	10 % Rule	90 % Rule
TU/e	High 2	3	33%	33%	33%
TU/e	Medium 3	250	0%	0%	100%
TU/e	Low 4	251	2%	1%	98%
Rotra	Critical 1	546	99%	0%	1%
Rotra	High 2	852	1%	0%	98%
Rotra	Medium 3	811	9%	0%	91%
Rotra	Low 4	4,894	18%	3%	79%
Philips	Critical 1	2	100%	0%	0%
Philips	High 2	64	8%	0%	92%
Philips	Medium 3	262	3%	1%	96%
Philips	Low 4	880	0%	0%	99%

TABLE OVERVIEW

Cus...		Priority	Amoun...	90% rule	Violations
TU/e	-	-	-	-	-
TU/e	2	High	-	-	-
TU/e	3	Medium	1	100%	0%
TU/e	4	Low	11	100%	0%
Rotra	-	-	-	-	-
Rotra	1	Critical	-	-	-
Rotra	2	High	0	-	-
Rotra	3	Medium	1	100%	0%
Rotra	4	Low	20	100%	0%
Philips	-	-	-	-	-
Philips	2	High	0	-	-
Philips	3	Medium	6	100%	0%
Philips	4	Low	6	100%	0%
Other	4	Low	-	-	-

Figure B.32: Analysis of customers, BPQ 2, for the Case Study with Celonis Process Mining - 2.

B.12 Dashboards created with CPM - BPQ 3

COMPARE RESOURCES

Filter on a specific resource

The table below shows the different response and solve times (median and average) of cases that are responded to by different resources.

Metrics per responding resource

Resource	Priority	Case Count	Median Respons...	Average Res...	Median Solve Time	Avg Solve Time
-	-	-	-	-	-	-
A	Low	51	242 min	709 min	5 hrs	35 hrs
A	Medium	21	529 min	709 min	21 hrs	47 hrs
B	Medium	2	136 min	68 min	111 hrs	80 hrs
B	Low	32	80 min	174 min	55 hrs	98 hrs
C	Low	15	31 min	96 min	90 hrs	105 hrs
C	Medium	4	2,152 min	1,275 min	79 hrs	448 hrs
D	Low	13	133 min	248 min	96 hrs	119 hrs
D	Medium	6	306 min	1,199 min	38 hrs	49 hrs
D	High	1	850 min	850 min	57 hrs	57 hrs
G	Low	1	73 min	73 min	273 hrs	273 hrs
I	Low	4	421 min	264 min	22 hrs	22 hrs
K	Low	14	12 min	225 min	195 hrs	345 hrs
K	Medium	5	6 min	12 min	31 hrs	326 hrs

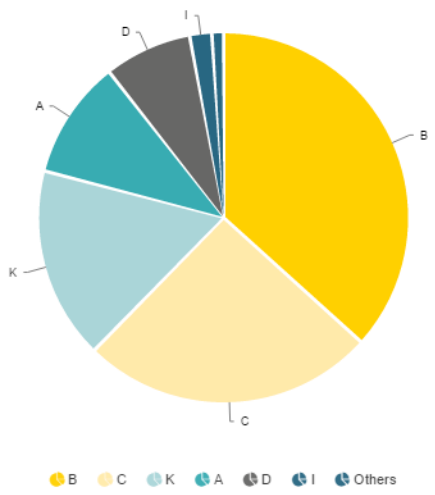
The table below shows the different solve times (median and average) of cases that are closed by different resources.

Metrics per ticket-closing resource

Resource	Priority	Case Count	Median Solve Time	Avg Solve Time
A	Low	50	5 hrs	20 hrs
A	Medium	16	13 hrs	15 hrs
A	Critical	1	0 hrs	0 hrs
B	Low	74	13 hrs	78 hrs
B	Medium	21	15 hrs	36 hrs
B	High	6	13 hrs	18 hrs
B	Critical	1	119 hrs	119 hrs
C	High	4	320 hrs	287 hrs
C	Medium	32	100 hrs	107 hrs
C	Critical	6	344 hrs	872 hrs
C	Low	72	41 hrs	67 hrs
D	Low	17	134 hrs	200 hrs
D	Medium	12	29 hrs	52 hrs
D	High	3	46 hrs	38 hrs
F	Low	2	18 hrs	9 hrs
G	Low	5	222 hrs	178 hrs
G	High	1	12 hrs	12 hrs
H	High	2	76 hrs	68 hrs
H	Medium	1	10 hrs	10 hrs
H	Low	1	3 hrs	3 hrs
I	Low	89	3 hrs	11 hrs
I	Medium	3	160 hrs	114 hrs

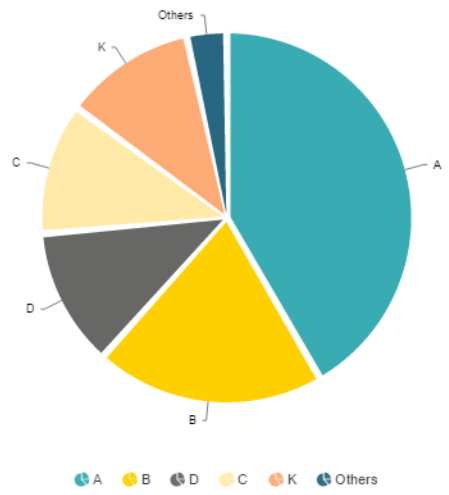
Figure B.33: Analysis of resources, BPQ 3, for the Case Study with Celonis Process Mining - 1.

Events from different resources



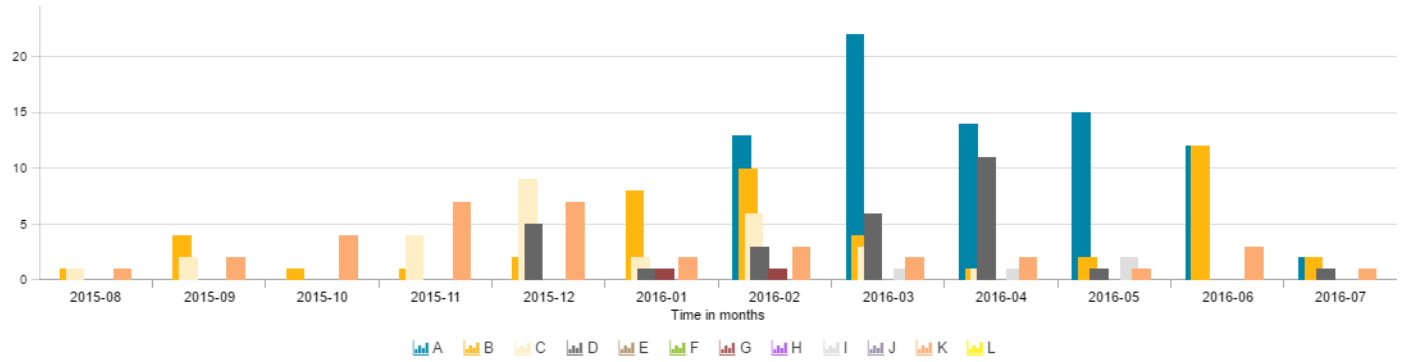
This component has a filter applied to it: without it, half of the pie-chart would be filled with the default setting for resource (Admin).

Responses from different resources



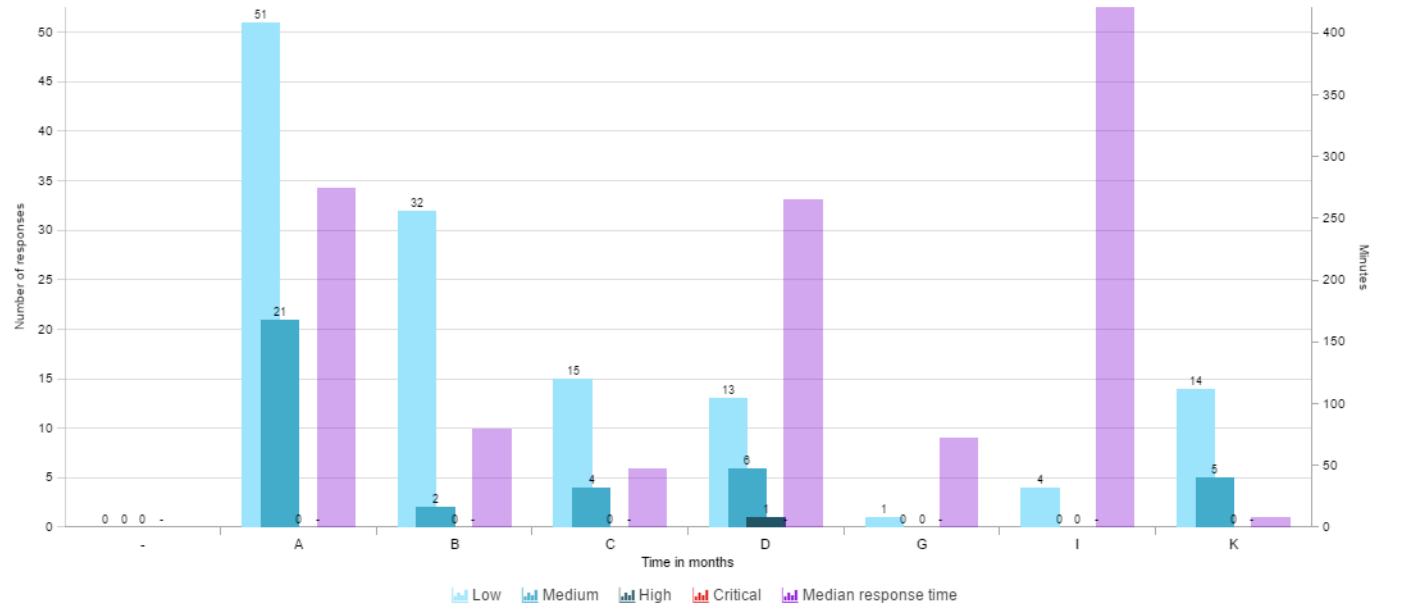
This component has a filter applied to it: without it, half of the pie-chart would be filled with the default setting for resource (Admin).

Amount of responses per resource



For the upped bar-chart, the graph only starts at August 2015 because before that period, events were not logged with resources in a consistent manner. For both bar charts, default setting for resource (Admin), is filtered out.

Response times of different resources



- Resource A is the most responding resource, however, this resources is (togerther with resource I) the slowest responder.
- The fastest responder is resource K.
- Only one ticket with a high priority seems to be issued by a customer and then answered by a resource (D), the others are of a lower priority.
- Four resources are mainly responding to tickets
- Resource I started responding to tickets in march and was gradually responding to more of them for three months, after which no responses were logged from this resource anymore.
- More resources seem to be involved in the process over time, for example resource A and D started responding to tickets february 2016 and december 2015 resectively.
- Some resources have been involved for a long peroid already, K and B respond to tickets in every month.
- Resource C has been very active since the start, but stopped responding to tickets in april 2016.

Figure B.34: Analysis of resources, BPQ 3, for the Case Study with Celonis Process Mining - 2.

Appendix C

Results Comparative Study

C.1 Actual Efficiency metric: Required Domain Knowledge

This section provides the ontology graphs that visualize the required domain knowledge for each of the process mining approaches in stage 2 to 3 of PM^2.

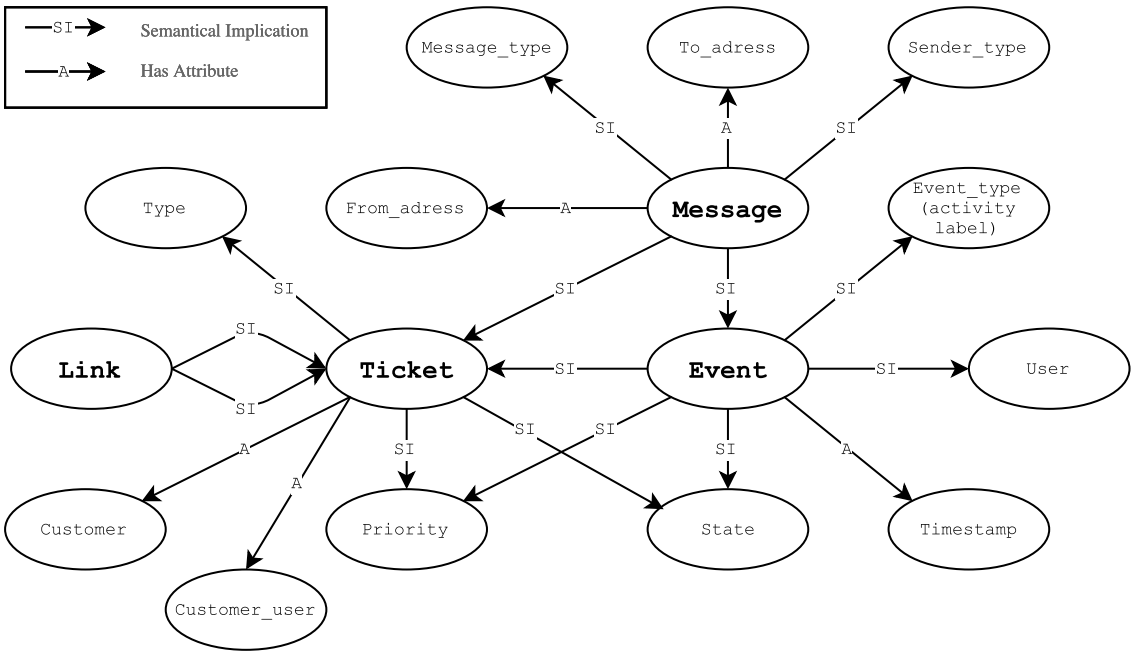


Figure C.1: Required domain knowledge visualized in ontology graph for stage 2, extraction, with TPM and CPM.

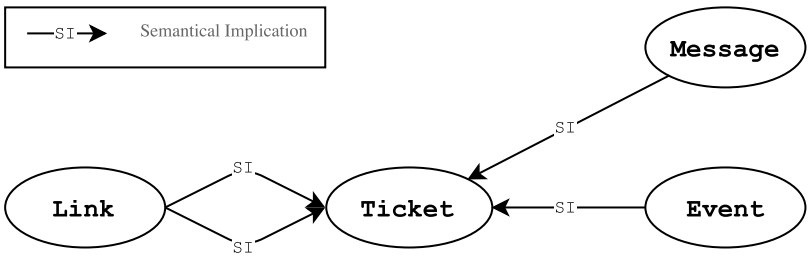


Figure C.2: Required domain knowledge visualized in ontology graph for stage 2, extraction, with RLPM.

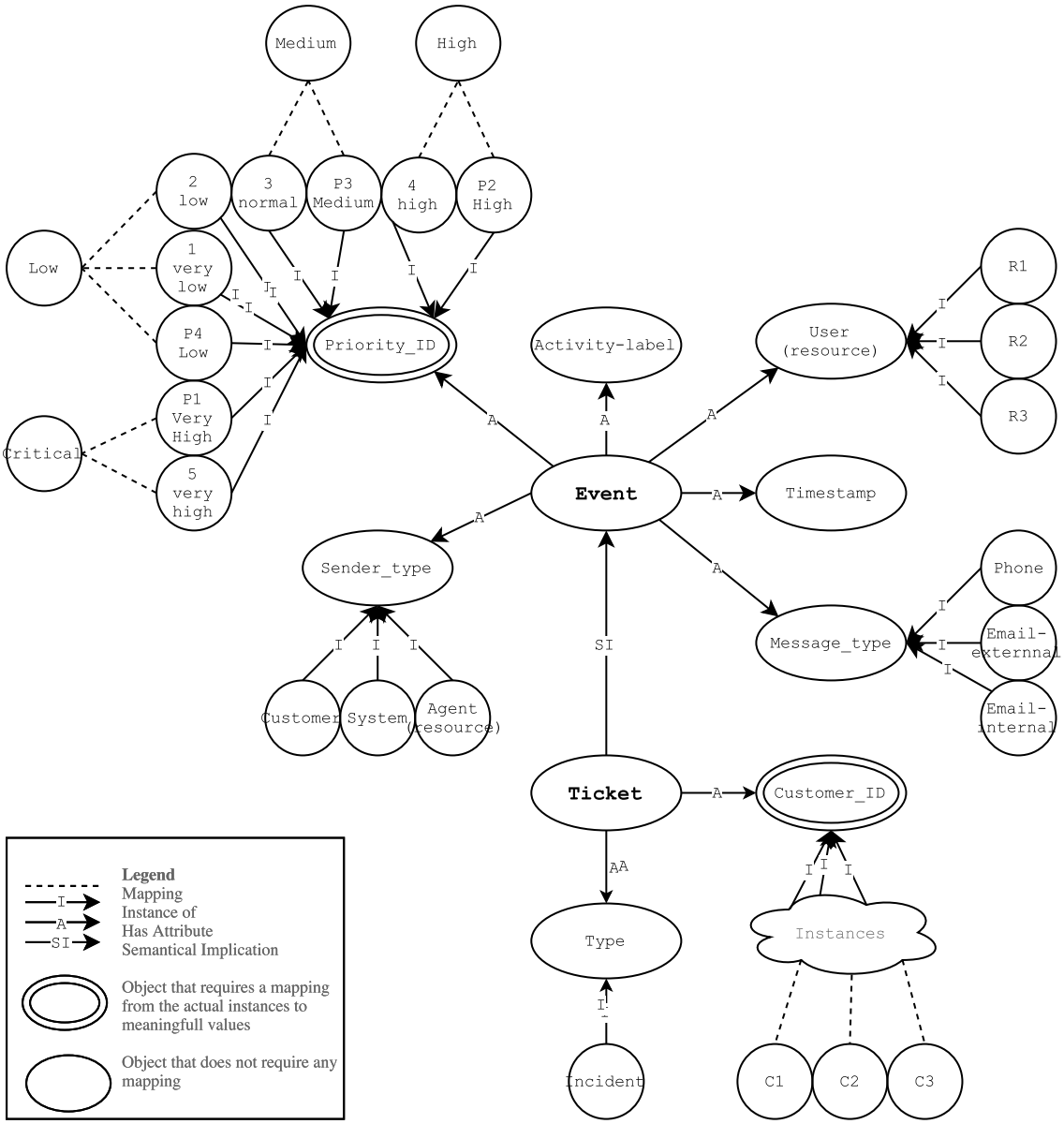


Figure C.3: Required domain knowledge visualized in ontology graph for stage 3, data processing, with the TPM and CPM approach (resources and customers have been anonymized).

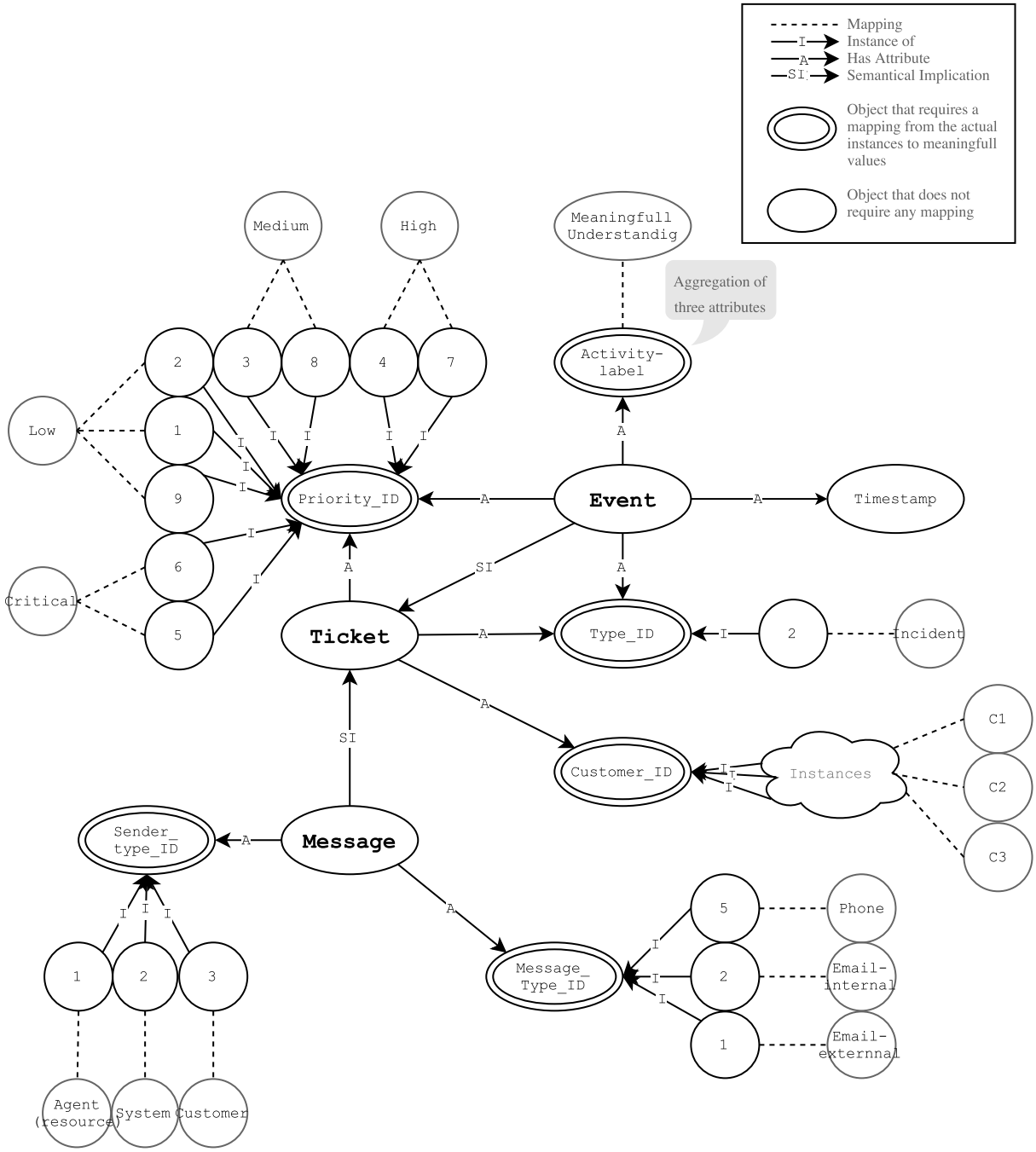


Figure C.4: Domain knowledge visualized as an ontology graph for stage 3: data processing with the *RLPM approach* (resources and customers are anonymized).

C.2 Comparison of Scores on Input Measures

Table C.1: Overview of the estimations for the input measures for all three approaches for all the stages per stage.

		Stage in PM[2]					
		1	2	3	4		
Input measure & approach		Planning	Extraction	Data Processing	Mining Analysis	Total	'Winning' approach
Time	TPM	-	112	31	-	**143**	TPM or CPM
	RLPM	-	230	27	-	257	
	CPM	-	112	32	-	**144**	
Required Domain Knowledge	TPM	-	35	61	-	**96**	None
	RLPM	-	8	87	-	**95**	
	CPM	-	35	61	-	**96**	
Number of Iterations	TPM	-	8	1	-	9	RLPM
	RLPM	-	2	2*	-	**4**	
	CPM	-	9	1	-	10	
Task Complexity	TPM	-	3	3	3	9	CPM
	RLPM	-	4	2	3	9	
	CPM	-	3	4	4	**11**	

*This extra iteration is the only iteration to a previous stage.

C.3 Comparison of Scores on Output Measures

Table C.2: Normalized results for metrics to estimate the actual effectivenesss of PM approaches for stage 2 of PM^2, scores in percentages.

	Stage 2: Extraction								
	TIMECOV	M	A-NOT	SP	C-NOT	V*	GRAN-T	ATTR	Average
TPM	100	100	=	100	=	0	=	100	**80.00**
RLPM	19.71	16.93	=	11.31	=	100**	=	59.50	**41.49**
CPM	100	100	=	100	=	0	=	99.00	**79.80**

*A suitable event granularity (i.e. variety/abastaction level) depends on the input data and BPQs of this specific case study [1]. We consider a V between 3 and 24 to be an acceptable range (i.e. 100), and everything above and below unacceptable (i.e. 0). Since 3 is the minimal amount of unique events that is expected in the log based on domain knowledge. We know that when V is less then three, compulsory activities are missing in the log. And 24 since process models with more than a few dozen of activities is unsuitable for human perception [8]. ** This would have been zero if the activity mapping of section 4.3 would be applied.

Table C.3: Normalized results for metrics to estimate the actual effectivenesss of PM approaches for stage 3 of PM^2, scores in percentages.

	Stage 3: Data Processing														
	M	SP	$\frac{LoD}{AE}$*	G	V**	UT	ST	ST_{min}	ST_{max}	MA	A_{min}	A_{max}	DQ-PR	MAT	Average
TPM	100	100	55.08	=	0	58.33	68	66	92	65	=	=	18.51	=	**62.29**
RLPM	5.14	2.54	32.35	=	100***	100	84	0	87	34	=	=	22.22	=	**46.73**
CPM	100	100	55.08	=	0	58.33	68	66	92	65	=	=	18.51	=	**62.29**

* The normalization of the AE and the LoD, is combined in one score. Namely by dividing the LoD by the AE and multiplying it by 100, to determine the amount of repetition in an event log. Repetition usually implies loops, which are often harder to handle by mining algorithms. Thus, more repetition is considered worse and results in a lower score, where less repetition results in a higher score. ** For normalizing V, the same rules are applied as were in table C.2. *** This would have been zero if the activity mapping of section 4.3 would be applied.

Table C.4: Normalized results for metrics to estimate the actual effectivenesss of PM approaches for stage 4 of PM^2, scores in percentages.

	Stage 4: Mining & Analysis				
	SEM	PD	CC	ENH	Average
TPM	90	100	100	100	**97.50**
RLPM	90	100	100	100	**97.50**
CPM	0	10	0	10	**5.00**

C.4 Comparison of Scores on Perceived Efficacy

In fig. C.5, C.6 and C.7 the percieved ease of use is estimated for each complete process mining approach.

1. Learning to operate TPM would be easy for me.

likely | extremely | quite | slightly | neither | slightly | quite | extremely | unlikely

2. I would find it easy to get TPM to do what I want it to do.

likely | extremely | quite | slightly | neither | slightly | quite | extremely | unlikely

3. My interaction with TPM would be clear and understandable.

likely | extremely | quite | slightly | neither | slightly | quite | extremely | unlikely

4. I would find TPM to be flexible to interact with.

likely | extremely | quite | slightly | neither | slightly | quite | extremely | unlikely

5. It would be easy for me to become skillful at using TPM.

likely | extremely | quite | slightly | neither | slightly | quite | extremely | unlikely

6. I would find TPM easy to use.

likely | extremely | quite | slightly | neither | slightly | quite | extremely | unlikely

Figure C.5: Perceived ease of use for the traditional process mining approach.

1. Learning to operate RLPM would be easy for me.

likely | extremely | quite | slightly | neither | slightly | quite | extremely | unlikely

2. I would find it easy to get RLPM to do what I want it to do.

likely | extremely | quite | slightly | neither | slightly | quite | extremely | unlikely

3. My interaction with RLPM would be clear and understandable.

likely | extremely | quite | slightly | neither | slightly | quite | extremely | unlikely

4. I would find RLPM to be flexible to interact with.

likely | extremely | quite | slightly | neither | slightly | quite | extremely | unlikely

5. It would be easy for me to become skillful at using RLPM.

likely | extremely | quite | slightly | neither | slightly | quite | extremely | unlikely

6. I would find RLPM easy to use.

likely | extremely | quite | slightly | neither | slightly | quite | extremely | unlikely

Figure C.6: Perceived ease of use for the redo log process mining approach.

1. Learning to operate CPM would be easy for me.

likely | extremely | quite | slightly | neither | slightly | quite | extremely | unlikely

2. I would find it easy to get CPM to do what I want it to do.

likely | extremely | quite | slightly | neither | slightly | quite | extremely | unlikely

3. My interaction with CPM would be clear and understandable.

likely | extremely | quite | slightly | neither | slightly | quite | extremely | unlikely

4. I would find CPM to be flexible to interact with.

likely | extremely | quite | slightly | neither | slightly | quite | extremely | unlikely

5. It would be easy for me to become skillful at using CPM.

likely | extremely | quite | slightly | neither | slightly | quite | extremely | unlikely

6. I would find CPM easy to use.

likely | extremely | quite | slightly | neither | slightly | quite | extremely | unlikely

Figure C.7: Perceived ease of use for the Celonis process mining approach.

In fig. C.8, C.9 and **??** the three PM approaches are assessed by the researcher for the *perceived usefulness* of each complete PM approach.

1. Using TPM in my Process Mining project would enable me to accomplish tasks more quickly.

likely | extremely | quite | slightly | neither | slightly | quite | extremely | unlikely

2. Using TPM would improve my Process Mining project performance.

likely | extremely | quite | slightly | neither | slightly | quite | extremely | unlikely

3. Using TPM in my Process Mining project would increase my productivity.

likely | extremely | quite | slightly | neither | slightly | quite | extremely | unlikely

4. Using TPM would enhance my effectiveness on the Process Mining project.

likely | extremely | quite | slightly | neither | slightly | quite | extremely | unlikely

5. Using TPM would make it easier to do my Process Mining project.

likely | extremely | quite | slightly | neither | slightly | quite | extremely | unlikely

6. I would find TPM useful in my Process Mining project.

likely | extremely | quite | slightly | neither | slightly | quite | extremely | unlikely

Figure C.8: Perceived usefullness for the traditional process mining approach.

1. Using RLPM in my Process Mining project would enable me to accomplish tasks more quickly.

likely | extremely | quite | slightly | neither | slightly | quite | extremely | unlikely

2. Using RLPM would improve my Process Mining project performance.

likely | extremely | quite | slightly | neither | slightly | quite | extremely | unlikely

3. Using RLPM in my Process Mining project would increase my productivity.

likely | extremely | quite | slightly | neither | slightly | quite | extremely | unlikely

4. Using RLPM would enhance my effectiveness on the Process Mining project.

likely | extremely | quite | slightly | neither | slightly | quite | extremely | unlikely

5. Using RLPM would make it easier to do my Process Mining project.

likely | extremely | quite | slightly | neither | slightly | quite | extremely | unlikely

6. I would find RLPM useful in my Process Mining project.

likely | extremely | quite | slightly | neither | slightly | quite | extremely | unlikely

Figure C.9: Perceived usefullness for the redo log process mining approach.

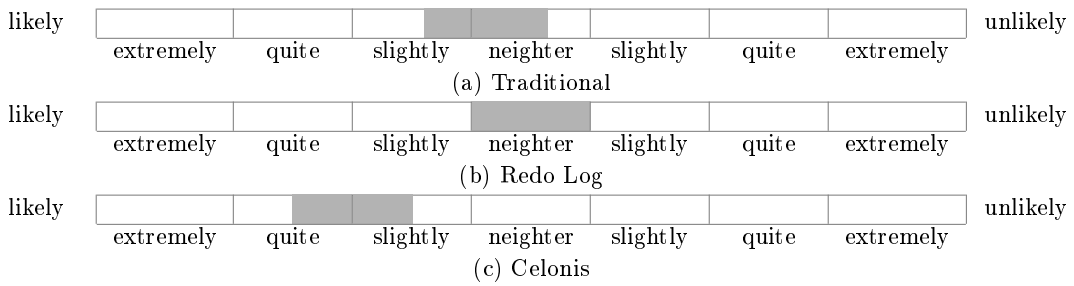


Figure C.10: Average Perceived Ease of Use for the three PM approaches: a for the TPM approach, b for the RLPM approach and c for the CPM approach.

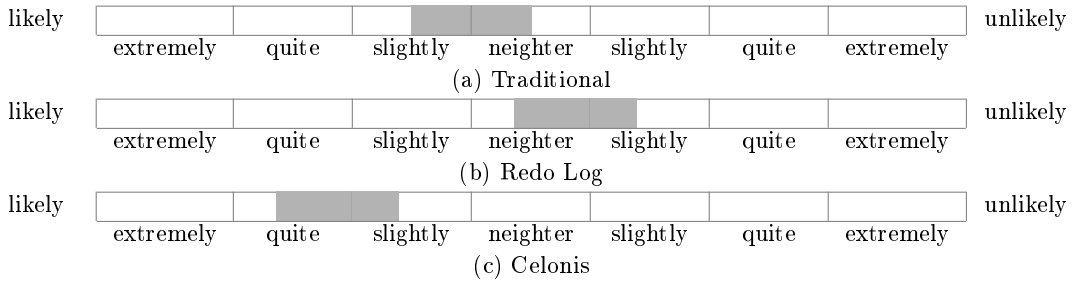


Figure C.11: Average Perceived Usefulness for the three PM approaches: a for the TPM approach, b for the RLPM approach and c for the CPM approach.

C.4.1 Normalized scores for Perceived Efficacy

Perceived Ease of Use								
	Question number							
	1	2	3	4	5	6	Average Score	Normalized Score
TPM	3	5	4	2	3	5	3,67	**47,62**
RLPM	3	5	5	5	3	3	4,00	**42,86**
CPM	1	5	2	2	3	2	2,50	**64,29**

Perceived Usefulness								
	Question number							
	1	2	3	4	5	6	Average Score	Normalized Score
TPM	5	2	4	4	4	2	3,50	**50,00**
RLPM	7	3	4	5	6	2	4,50	**35,71**
CPM	2	2	4	3	1	2	2,33	**66,67**

Average Perceived Ease of Use and Perceived Usefulness				
	. Perceived Ease of	erceived Use	Average PEoU + PU	Normalized Score
TPM	3,67	3,50	3,58	48,81
RLPM	4,00	4,50	4,25	39,29
CPM	2,50	2,33	2,42	65,48

Figure C.12: Overview and averages of the scores on the perceived ease of use and perceived usefullness.