A Survey of Process Mining Competitions

Iezalde Faria Lopes

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisor: Prof. Diogo Manuel Ribeiro Ferreira

Examination Committee

Chairperson: Prof. Luís Manuel Antunes Veiga

Supervisor: Prof. Diogo Manuel Ribeiro Ferreira

Member of the Committee: Prof. José Alberto Rodrigues Pereira Sardinha

July 2019

Abstract

Competitions involving the analysis of real-world data have been contributing to advances in data science in general, and in process mining in particular. The Business Process Intelligence (BPI) challenge is one of such competitions, which provides an event log from a business process and asks participants to analyze that process in order to address a set of business questions. Participants then report their findings, which often requires an analysis of the event log according to different perspectives. In this work, we present a survey of the BPI challenges from 2011 to 2018. We conclude that the analysis of a business process from a process mining perspective alone is often not enough, and many authors resort to data mining techniques to complement and expand their findings. Based on this insight, we propose an approach that combines process mining and data mining in order to understand the behavior of a business process, and we describe the application of this approach to the BPI Challenge 2019.

Keywords: Event Logs, Process Mining, Data Mining, Machine Learning

Resumo

Competições que envolvem a análise de dados reais têm contribuído para avanços na área de ciência de dados em geral e na extração de processos em particular. O desafio *BPI Challenge* (*Business Process Intelligence Challenge*) é uma dessas competições que fornece um registo de eventos de um processo de negócio e pede aos participantes que analisem esse processo abordando um conjunto de questões específicas. Os participantes então relatam suas descobertas, o que geralmente requer uma análise do registo de eventos de acordo com diferentes perspetivas.

Neste trabalho, apresentamos um estudo comparativo dos desafios *BPI Challenge* de 2011 a 2018. Concluímos que a análise de um processo de negócio numa perspetiva de extração de processo muitas vezes não é suficiente, e muitos autores recorrem a técnicas de prospeção de dados para complementar e expandir as suas conclusões. Com base neste estudo, propomos uma abordagem que combina extração de processos e prospeção de dados para entender o comportamento de um processo de negócio, e descrevemos a aplicação dessa abordagem ao *BPI Challenge 2019*.

Palavras-chave: Extração de Processos, Prospeção de Dados, Aprendizagem Automática

Acknowledgments

First and foremost, I thank God for everything.

Second, a special thanks goes to Prof. Diogo R. Ferreira for his utmost assistance, availability and guidance. The efforts he dedicated to this work greatly improved its quality.

To my parents, thanks for everything.

A special thanks to Felizarda Chilala.

Last but not the least, to my friends and colleagues a special thanks.

Table of Contents

List of figures

List of Tables

Chapter 1
Introduction

The field of data science is thriving with competitions where participants are asked to perform challenging tasks involving the analysis of real-world data. In the field of process mining, the competition that has brought the community together around real-world event logs is the Business Process Intelligence Challenge. For the past nine years, the BPI challenge has been providing event logs that have served as a testbed for many process mining techniques.

An interesting aspect of the BPI challenge is that it has drawn the attention not only of academic researchers, but also of practitioners working at the intersection between business process management and data analysis. Furthermore, the BPI challenge has served as an introduction to many students entering the field of process mining, and some of its event logs, namely the BPI Challenge 2012, have become a standard of reference for many authors.

In this work, we provide a synthesis of each BPI challenge, followed by an overview of the tools and techniques that have been most used across all BPI challenges. Based on this analysis we propose an approach to address future BPI challenges, which is applied to the BPI challenge 2019.

1.1. Motivation

There have been several BPI challenges across the years, and in this work we look at them from a holistic view. For example, “Which are the most successful process mining techniques?”, “Which process mining perspectives receive the most attention and why?”, “Are process mining tools enough to analyze real-world event logs?”.

We want to answer these questions by performing a comparative analysis of all BPI challenges, in terms of approaches, techniques, and tools used, together with the results obtained. We want to draw general conclusions that are useful for new BPI challenges and other process mining competitions involving real-world event data.

1.2. Objectives

The concrete goals of this work can be described as follows:

- Review all BPI challenges – our goal here is to analyze all submissions to each BPI challenge and gather information about the findings and the approach followed by each author, together with the techniques and tools that were used.
- Provide a comparative analysis of all BPI challenges – our goal here is to compare the techniques and tools that were used across multiples challenges. Here it will be interested in checking whether some techniques can always be used independently of the specific characteristics of the given event log.
- Understand which techniques are most used or most successful – since process mining comprises several perspectives and multiple tools within each perspective, it is useful to identify which techniques usually yield the best results and are preferred among process mining practitioners.
- Validate those findings by applying them to the BPI challenge 2019 – from the insights gathered from the previous analysis, our goal is to devise a general approach that can be applied to the analysis of event logs in similar competitions.

1.3. Document structure

This document is structured as follows. Chapter 1 introduces the work to be performed and gives the reader the main objectives of this work. Chapter 2 addresses the state of art. In this chapter, we briefly introduce each BPI challenge and provide a brief summary of each submission. In Chapter 3, we provide a comparative analysis. This comparison focuses on the use of tools and techniques in the BPI challenges 2011-2018. We conclude the chapter with a discussion in which we propose an approach to address future challenges.

In Chapter 4, we use the proposed approach to analyze the event log of the BPI challenge 2019. This chapter serves not only has a proof of concept illustrating the analysis method, but also provides insights regarding the business process for this competition. Chapter 5 summarizes our main contributions and describes opportunities for future work.

Chapter 2

The BPI Challenges 2011-2018

In this chapter, we provide an overview of each BPI challenge. After briefly describing the business process and the business questions to be addressed, we provide a brief summary of each submission, highlighting the tools and techniques used by each author. At the end of each BPI challenge, we provide a brief summary with our main conclusions.

2.1. BPI Challenge 2011

In 2011, the BPI Challenge involved an event log from a Dutch Academic Hospital. The event log concerned cancer patients in a Gynecology department, and it contained the diagnosis and treatment activities that were performed on those patients. The names that were given to these activities do not seem to follow a strict format, which suggests that they were either picked from a large number of possibilities or written as free text. This led to a relatively large number of task names, which means that any direct control-flow analysis of the event log is very likely to yield a spaghetti model [1].

In this edition of the BPI Challenge, participants could focus on a specific aspect and analyze it in depth or focus on a broader range of aspects without going into too much detail. Three submissions were received:

- J.C. Bose and W. van der Aalst [2] followed a three-step approach, starting by preprocessing the event log (filtering and splitting it into smaller, more homogeneous logs), analyzing the event logs (using both the Enhanced Fuzzy Miner [3] and trace alignment techniques [4] in ProM) and interpreting the results. Due to the heterogeneity of the process, the team proposed segregating homogenous cases based on properties such as the diagnosis code, treatment code, department (General Lab Clinical Chemistry, Pathology, or Radiology) and urgency. In the end, the team was able to present a compact process model. The trace alignment analysis carried out by the team also yielded common patterns of execution and exceptional/rare behavior.

- F. Caron et al. [5] assumed that the process model was spaghetti-like, and initial analysis conducted by the team using ProM proved their initial assumption. To analyze the process, the team adopted two strategies to create more homogeneous groups: first they analyzed sub-processes by department (using the Heuristics Miner [6], they created a process model for the radiotherapy department with 10 tasks) and then they analyzed the use of a specific therapy (Paclitaxel). Using the social network analysis plug-in in ProM with handover of work [7], the team concluded that departments are very interlinked due to their interactions.
- G. Varvaressos [8] defined a hospital ontology based on the attributes and values found in the event log, and then grouped events according to different levels in the ontology. The process turned out to be understandable only at the lower levels (e.g. at the level of individual patients). The author used the heuristics miner and performance metrics in ProM. He defined a BPMN model based on 6 process instances.

From these submissions, we can see that all teams used ProM combined with different preprocessing techniques to create subsets of cases which could be mined to obtain an understandable process model.

2.2. BPI Challenge 2012

In 2012, the BPI Challenge involved an event log from a Dutch Financial Institute. The event log came from an application process for personal loans or overdraft. Here, three different subprocesses could be immediately distinguished from the use of event prefixes, namely 'A_' (for application states), 'O_' (for offer states) and 'W_' (for work item states).

As in the previous edition of the challenge, participants could focus on a specific aspect and analyze it in depth or focus on a broader range of aspects without going into too much detail. In this challenge, there were four business questions: (1) estimators for the total cycle time; (2) which resources generate the highest activation rate of applications; (3) how the process model looks like; (4) which decisions have greater influence on the process flow and where they are. Six submissions were received:

- D. Bautista et al. [9] simplified the event log by removing redundant events. They were able to identify the standard control flow for a successful application and extended it incrementally, by adding other states. Then, in the performance perspective, the team analyzed the event log at a case level, event level, and resource level. Going a step further, they used classification and regression tree (CART) analysis to segment applications according to their approval result.
- J. C. Bose and W. van der Aalst [10] started with the organizational perspective. They analyzed the resource distribution per work day and discovered the resources that approve a loan/overdraft. Using a custom formula, they estimated the execution time of different activities. Multitasking was identified a possible bottleneck. Using ProM's trace alignment [4] and LTL-checker [11] plug-ins, they identified process deviations. Using the heuristics miner [6], they

discovered a spaghetti model. The team then partitioned the event log into more homogenous trace clusters, and presented more comprehensible and simpler process models.

- A. Adriansyah and J.C. Buijs [12] filtered the event log, and then used the α-algorithm [13], heuristics miner [6] and the ILP miner [14] ProM plug-ins to discover a process model. After picking the best model based on fitness and precision [15], they manually improved it. The model was used to perform a performance analysis, focusing on bottlenecks. Using alignments, they identified the non-deviating activity executions and used them to measure performance of process executions.
- T. Molka et al. [16] started analyzing the log with dotted charts [17] and statistics. Using their own prototype for creating dotted charts, the team identified six activity clusters (Monday through Saturday) and its main events, as well as their statistical distribution. When using the resources to classify the dots, the team identified the automatic (system) resource as well as the resource clusters. Using Disco, the team obtained the process model and provided the most common variant, covering 26% of cases.
- H. M. W. Verbeek [18] created three event logs (one for each event type) using ProM's Simple Log Filter plug-in [19]. The author analyzed the three event logs and two combinations of them. Using ProM's Transition Systems Miner [20], a model for each event log was found. The author further analyzed the loan applications by requested amount, and also identified the most frequent resources.
- C.J. Kang et al. [21] started by using Disco to discover the complete process model. Afterwards, they created separate process models based on the requested amount. Using Disco, they identified the mean cycle time and the main causes for cycle time increase. They also identified the three major points causing the process bottlenecks and concluded that handover of work (mainly self-looped) was the main reason for bottlenecks. In addition, the higher the requested amount, the more a case takes to complete and more likely it is to cause a bottleneck. Using ProM's social network analysis plug-in [7], the team identified the key resources that decide an application outcome.

In this challenge, we see that teams started using both ProM and Disco, and they also focused on several process mining perspectives (control-flow, organizational, performance), while using a variety of analysis plug-ins. Preprocessing did not play such a large role as in the previous edition, since there were three subprocesses that were clearly identified in the event log.

2.3. BPI Challenge 2013

In 2013, the BPI Challenge involved an event log from Volvo IT Belgium. The log contains events from an incident management and problem management system called VINST. Each event refers to a change in the status/sub-status of an instance of those processes.

In this challenge, there were four business questions of interest for the process owner: (1) whether incidents are pushed too often to second- and third-line support; (2) whether there is ping-pong behavior between teams; (3) whether the "wait user" status reveals performance problems; (4) whether process instances conform across departments. Twelve submissions were received:

- C. J. Kang et al. [22] performed an exploratory analysis and were able to answer questions one, two, and three. Using a footprint matrix to capture the activity precedence, the team concluded that there are several differences in the incident management process, but very little differences in problem management process, answering question four. They found that organizational structures (departments) were not conforming, which means a lack of process standardization.
- A. Bautista et al. [23] preprocessed the data and then discovered a standard process model for both processes. This enabled them to evaluate conformance, answering question four; their processes are roughly equivalent, with few exceptions. Considering only departments C and A2 (~86% of all cases), the team discovered that C mainly handles push-to-front cases, while A2 focuses mainly on cases originating at second- and third-line support. The team identified the top support teams and products exhibiting ping-pong behavior, and calculated the impact of ping-pong on completion time (ping-pong cases are on average 2.3x longer). For wait-user sub-status, the team found that the impact on completion time is 7x longer duration; the team also identified the most abusing support teams, departments and countries.
- E. Dudok and P. van den Brand [24] preprocessed the data and used Perceptive Process Mining (PPM) to identify push-to-front behavior and provide statistics answering question one. For question two, they identified ping-pong behavior based on a threshold in the social network analysis with handover of work, again using PPM. For question three, the authors focused on cases within the target value for the handle time and analyzed wait-user usage. They conclude that ~11% of those cases are misusing the status and pinpointed potential misusers. For question four, conformance was assessed from a control-flow perspective.
- M. Arias and E. Rojas [25] loaded the data into Disco and after filtering they provided statistics for questions one and three. To answer question two, the team exported the log to ProM and followed two strategies: (a) sequence clustering followed by social network analysis with handover of work; (b) trace alignment. Both strategies uncovered the ping-pong behavior. To answer question four, they used heuristic miner [6] to obtain three models (general and departments A2 and C) that were compared at a control-flow, performance and statistics level. The team then answered two extra questions, providing additional insight to the process.
- S. van den Broucke et al. [26] used Disco and ProM to obtain a spaghetti model, so they provided a simpler model with the top-5 variants (41% of all cases). After processing the event log in Disco, it was imported into R in order to answer question one. For question two, the team found the ping-pong behavior using a tandem repeats function in R; this approach was later validated in Disco. For question three, the team also filtered the log, inspected it and collected statistics. Finally, question four was answered in two ways: (a) by comparing departments based on mean case duration and (b) by defining a prescriptive process model and comparing it to the departments through ProM's trace alignment using the conformance checking plug-in.

- F. van Geffen and R. Niks [27] state that process mining can enhance Six Sigma's improvement cycle.
- J. Hansen [28] first sought an understanding of the process by using Disco. The author then answered question one by filtering out cases not pushed to front. For question two, the author filtered out cases not having ping-pong. Regarding question three, the author focused on the wait-user status recorded by users other than the automated system. And for question four, the author created a global process model in Disco and compared it to the specific models of departments. Finally, the author documented the process in BPMN using Enterprise Architect.
- J. Hevia and C. Saint-Pierre [29] mined the push-to-front behavior using ProM's heuristic [6] and fuzzy [30] miners. For question two, they performed a social network analysis with handover of work and mined a global model as well as a model for each country. For question three, the authors did data processing followed by process discovery in ProM. Due to the size of the model, trace filtering was applied and using both ProM and Disco the authors identified the wait-user problem, providing a country-to-country analysis based on the control-flow perspective. Question four was addressed separately for each process. For incident management, the conformance between departments was analyzed from a statistical and a control-flow perspective; for problem management, it was done from a control-flow perspective only.
- J. Martens [31] filtered the log in Disco and presented statistics to answer question one. The same approach was followed for questions two and three. For question four, the author performed a comparison of different departments based on a control-flow perspective.
- Z. Paszkiewicz and W. Picard [32] provided log statistics as well as the standard case flow for each process. The authors then focused on question one and provided a process model for push-to-front behavior. Using chord diagrams [33], the team tested some hypothesis regarding the organizational structure and provided statistics. With a social network analysis approach to address question two, the authors worked on three perspectives to find the ping-pong behavior: (a) product perspective, (b) cycle perspective, and (c) organizational structure. For question three, the authors considered a misuse of the wait-user status based on a threshold for both frequency and percentual occurrence. For question four, the departments were compared from a control-flow perspective taking into account the expected standard flow.
- S. Radhakrishnan and G. Anantha [34] joined all the logs into one and clustered it into four groups. For question one, they provided a process model showing all push-to-front behavior. For question two, they found the ping-pong behavior on a control-flow perspective. For question three, the authors provided statistics as well as fuzzy models depicting the behavior. They used Disco and Excel.
- P. van den Spiegel et al. [35] defined a standard process flow in BPMN and mapped it to the event log. For question one, they used Disco to filter the event log and collect statistics. For question two, they used ProM's social network analysis plug-in using handover of work [7] with additional statistics. For question three, they used a percentual threshold to identify misusage of the wait-user status. The authors also presented a correlation between misusage and throughput time. For question four, the log was analyzed in ProM and a process model was

obtained. Then the authors mined a process model for each department and compared it to a global process model; some statistics regarding the flow was also provided.

In general, all authors faced the need to perform preprocessing of the event log. In this edition of the BPI challenge, there is a noticeable trend towards the use of statistics. In a sense, this was to be expected since the business questions required ranking and comparison among process instances and case attributes. Disco and ProM were the most popular tools. Both the control-flow perspective and the organizational perspective played a key role in the analysis.

2.4. BPI Challenge 2014

In 2014, the BPI Challenge involved an event log from Rabobank Group ICT. The log concerned the ITIL processes [36] implemented by Rabobank, encompassing three desk service processes: interaction management, incident management and change management.

In this edition of the BPI challenge, a new category (student) was introduced, which targeted BSc, MSc and PhD students. The main goal of the challenge was to predict the workload of the Service Desk (SD) and IT Operations (ITO) when a new change is created. In this challenge, there were four business questions: (1) identification of impact patterns; (2) impact of such patterns on workload; (3) improvement of service level after each change; (4) creativity challenge, where participants could follow any creative analysis. Thirteen submissions were received:

- P. Buhler et al. [37] developed their own metrics to measure the improvement through change implementation. They found four impact patterns (acute impact, delayed impact, extended impact and problem resolution), answering question one. For question two, two approaches were tried: (a) using a decision tree, the authors classified those patterns according to their impact on the workload (favorable, unfavorable or neutral); and (b) using a multinomial logistic regression model, they were able to determine the probability of a specific change resulting in a given impact. For question three, the authors evaluated performance trends using their own metrics and discovered that SD is improving, but ITO is worsening. Additional insights to the process were also obtained, especially the main causes of process bottlenecks (inefficient incident assignment, slow customer updates, dependency on external contractors and incident resolution delay).
- J. Suchy and M. Suchy [38] conducted (a) data discovery, (b) data preparation and (c) model definition phases. In phase (a), they picked relevant attributes to be able to identify relations among interactions, incidents and changes. In phase (b), they defined the impact patterns and their parameters (questions one and two). In phase (c), they defined a predictive model. Linear regression was chosen to fit the predictive model. For question three, the team applied linear regression on the indicators and was able to determine whether service levels were increasing or decreasing.

- M. Dees and F. van den End [39] preprocessed the data to build four predictive models: predict workload of SD at configuration item level and at service component level, and predict workload of ITO for the same levels. Using R package changepoint, the authors detected changes in variation and the mean of the time series. After selecting the dependent variables, the authors used R package Rattle to build the model. From the package, the authors picked decision tree, logistic regression, random forest, support vector machine and Ada boost. In general, decision tree had the lowest overall error. The authors identified all impact patterns. For question three, the authors provided statistics to answer it.
- S. Buffett et al. [40] followed two approaches: (a) a macro analysis over the entire dataset, and (b) an analysis for each configuration item using sequence mining. After preprocessing the data and following approach (a), the authors used the Apriori algorithm [41] to identify large item sets with a minimum frequency support. Impact patterns were then mined, and parameters for each of them were identified using R-squared and visual inspection. For question three, the change in average steps to resolution was identified. On approach (b), the authors used association rules, sequential pattern mining and sequence classification to find the impact patterns. The authors also identified predictive patterns, using sequence classification and naïve Bayes classifier.
- P. van den Spiegel et al. [42] first sought an understanding of the process and its data. For question one, the authors identified the patterns using linear regression and decision tree (C4.5 algorithm) models. Disco was also used, after preprocessing the log, to obtain a process model in which the patterns were also identified. For question two, parameters were identified, and additional statistics were provided. For question three, the authors analyzed one service component and noted that change in average steps to resolution showed a downward trend.
- J. Hansen [43] sought a process understanding using Disco. Then the author focused on question one, and after analyzing the process concluded that interactions increase after implementing a change, whereas incidents have a weak correlation to changes. For question two, the author made a performance analysis as well as a control-flow analysis. For question three, statistics were provided. For question four, the author provided insights about the change process.
- T. Thaler et al. [44] applied an ETL procedure to the original logs. As ITIL has no interaction concept and Rabobank's process does have, the authors considered incidents and interactions the same thing. Using ARIS (by Software AG) the authors defined the ITIL incident and change management processes and pinpointed conformance on a control-flow level. The incident and change management processes were assessed using ITIL metrics and their performance was also assessed.
- M. Arias et al. [45] compared the number of opened/closed changes per week with the number of incidents and requests for information to understand the relationship between the change implementation and the SD workload. Based on the correlation, they created a prediction model for the number of incidents after a change implementation, using a multilayer perceptron classifier in WEKA, thus answering question one. To answer question two, the authors

performed a control-flow analysis. As the model obtained in Disco was a spaghetti one, they applied trace clustering which resulted into four clusters. The authors also developed their own flexible heuristic miner to help them answering question two. For question three, the authors analyzed the behavior of the service levels after each change implementation. Additionally, the authors analyzed the relation between the teams in the interaction, incident and change processes using ProM's social network analysis (handover of work, working together, doing similar tasks and subcontracting).

- G. Cacciola et al. [46] used their own algorithm that plotted the current number of interactions and incidents along with the number of changes each time an event happened. As no clear pattern was identified, the authors concluded that changes are not executed in response to interactions, answering question one. For question two, the authors concluded that a steady state is never reached. Their analysis showed that the cumulative number of closed interactions/incidents grows linearly. Therefore, the team could not characterize the increase/decrease in the number of closed interactions. For question three, the authors used again their own plotting algorithm. However, their analysis for each service component (SC) produced over 300 different diagrams, so for each SC they computed how often the average number of closed interactions increased or decreased between two changes. The authors found out that the average between two changes increases and decreases frequently, regardless of the number of changes applied. For question four, the authors used Disco and performed bottleneck and resource utilization analysis and they discovered the processes with the longest handle time. Still in Disco, the authors drew the dependency graph and discovered the resources delaying the process.
- B. Brandão et al. [47] started by providing statistics regarding each log. Then they analyzed interactions causing incidents. The authors then performed bottleneck (performance) analysis using Disco. During control-flow analysis, the authors uncovered potential process unconformity based on their flow. Afterwards, the authors performed social network analysis to uncover similar teams. They used a similarity measure for that purpose and implemented the social network analysis algorithm using Java and Gephi. The authors were able to identify groups of teams performing similar tasks. For impact pattern identification, the team used a declarative modeling language, Declare, and using the UnconstrainedMiner [48] they were able to mine declarative constraints that were later validated in Disco.
- S. Hong et al. [49] started their analysis by question four following van der Heijden's [50] process mining project methodology. The authors linked Rabobank's three processes as an end-to-end ITIL process. Whilst performing control-flow analysis, the authors pinpointed unexpected flows which could or might be unconforming. For question one, the authors performed a statistical analysis and by plotting the distribution of the workload they identified three impact patterns (increase, decrease, and no variance). For question two, a decision tree classifier was used to identify the parameters for those patterns. For question three, the authors performed a statistical analysis to assess if changes are improving or worsening the average steps to resolution, and concluded that for 79% of cases it is improving.

In this challenge, most teams turned to data mining tools and techniques to answer the required business questions. It is the first time that we see data mining being preferred over process mining tools, although Disco and ProM still had some role in the analysis.

2.5. BPI Challenge 2015

In 2015, the BPI Challenge involved event logs concerning building permit applications in five Dutch municipalities. In this challenge, there were six business questions: (1) roles of people involved in the process; (2) possible improvements to the organizational structures; (3) changes in the process due to relocation of employees; (4) effect of outsourcing in organizational structures; (5) throughput times for each municipality; (6) control-flow for each municipality. Nine submissions were received:

- U. van der Ham [51] focused on service time, compliance with regulations, and quality of the process as indicators to compare the five municipalities. The author then identified the average throughput time per municipality, their differences and the reasons (batch processing, objection and appeals) using Disco. Using decision trees in WEKA, the author tried to link the occurrence of objections and appeals to a resource, activity or both, but it could not be linked. However, when using rule learning in municipality 5, it was possible to link objections and appeals to old cases. Using concept drift analysis available in ProM, the author identified the major changes in the process. However, the author could not link the merge of offices to when it led to change.
- I. Teinemaa et al. [52] started analyzing the organizational structure using the Kleinberg algorithm, to identify key resources from a workload distribution perspective, and then discovered that municipalities having clear separation of roles perform better in high-level activities. For low-level activities, using Disco, the authors analyzed the handover of work, and found that any resource hands over the work to any other resource. The authors pinpointed lack of separation of roles tending to lead to longer handle time. For question three, a concept drift analysis was performed, and although this method detected change points in the process, the change points do not coincide. For question four, the authors performed a what-if analysis assuming three tasks could be outsourced and concluded that probably the resources will be better utilized in such scenario, and the duration of a case will also be lower. A possible drawback is an increase in the average waiting times. For question six, they used heuristics miner in ProM to obtain process models for each municipality. The models were used to perform conformance checking which yielded no significant differences.
- P. van den Spiegel and L. Blevi [53] used social network analysis in ProM and identified only one group of resources. Using dotted chart analysis, the authors were able to identify the groups of resources; answering question one. To answer question two, the authors performed another social network analysis in ProM and determined how individual cases are routed between resources. For question tree, using Disco to compare the process flow among the municipalities over time, the authors discovered the municipalities (2 and 3) and pinpointed the change. To

answer question six, using Disco to compare the main process flows of the municipalities, three main phases were discovered: receive permit application, determine procedure and evaluate permit application. Differences were pointed for each of them. For question five, the previously obtained process models were used to uncover the differences in throughput times.

- S. Buffett and B. Emond [54] preprocessed the log and using a filter in ProM kept only the most frequent events. Using the inductive miner plug-in, they created process models for each municipality using only the 90% most frequent sequences. The authors then performed social network analysis to create a resource-activity matrix per municipality, answering question one. For question two, a global process model was obtained and compared to each municipality, and improvements were pinpointed at a control-flow perspective. For question four, the authors calculated the impact of outsourcing at process and organizational perspectives. For questions three and five, the authors performed sequential pattern mining regarding case completion time. For question six, the authors used a sequence classification per municipality and chi-squared test to identify the differences in control-flow.
- P.M. Dixit et al. [55] performed an analysis at organizational, performance, and control-flow perspectives. For the organizational perspective, using a decision tree classifier, dotted chart analysis and social network analysis, the authors answered questions one and three. For the control-flow perspective, changes in the flow were pinpointed, and the authors also assessed the differences in the flow of all municipalities. Regarding performance analysis, they identified differences in average throughput times, and found that cases using the automated system tend to have higher throughput times.
- N. Martin et al. [56] preprocessed the log and performed a control-flow analysis. After discovering each municipality's process model (using heuristics and inductive miners), the models were compared to each other. The authors then moved to the performance perspective, and provided insights about throughput time. They found two factors influencing throughput time: some case attributes, and some subprocesses. Regarding the organizational perspective, the authors presented a resource overview, their involvement in the process, and resources in multiple municipalities.
- J. Martens and P. Verheul [57] reviewed the BPI Challenges from 2011 to 2014 and guided their analysis based on the lessons learned. For question one, the authors identified the roles and stages in the process using Disco, and compared it across municipalities. For question two, segregation of duties, specialization and clear structure procedures were pinpointed. For question three, Excel was used to identify resources working in multiple municipalities. Using Disco, the authors identified the change in the process flow. For question four, the authors identified some of the procedures which could be outsourced and the possible impacts. For question five, throughput times were retrieved and compared between municipalities. The differences and performance issues were also identified. For question six, the differences in control-flow were sub-route, self-loops, long sequence, and shortcut in initial flow.
- J. Suchy and M. Suchy [58] preprocessed the log and focused on question one. To answer it, the attributes in the event log were assessed and seven roles were identified for all

municipalities. When considering the tasks performed by resources, the team identified thirteen roles. For question two, a narrow specialization of resources was identified as well as an even distribution of workload (based on their own algorithm and social network analysis in Minit). For question three, the authors identified the changes in the process. For question four, the authors characterized the impact of roles in the municipalities. For question five, the analysis was performed at case attribute level, throughput times, number of steps, and time to complete. For question six, the authors pinpointed the differences in the flow based on the tasks performed.

- H. S. Choi et al. [59] first sought an understanding about the process and its data. For questions one, they provided statistics about the resources that were already present in the event log. For question two, based on segregation of duties, the authors identified readjustment of roles and responsibilities. For question three, the authors obtained process models prior to and after the relocation. Process flows for different municipalities were compared. For question four, four procedures were appointed to be outsourced and the impact of outsourcing was also assessed. For question five, authors classified the top-25 fastest and slowest cases, and applied decision trees (in WEKA). For question six, the authors defined a reference model and then compared it to the control-flow in each municipality.

In this challenge, the business questions were very related to the control-flow and organizational perspectives of process mining. Hence, most authors have used process mining tools (such as Disco, ProM and Minit) to perform their analysis. The use of data mining techniques (e.g. decision trees), although present in some submissions, had only a minor role.

2.6. BPI Challenge 2016

In 2016, the BPI Challenge involved event logs from the Dutch Employee Insurance Agency (UWV). The logs concerned the customer interaction through different channels (website, messages, and call center) when applying for unemployment benefits. The main goal was to provide insights about the way the website was used.

In this challenge, there were six business questions: (1) identification of usage patterns on the website; (2) change of usage patterns over time; (3) transitions from the website to other channels; (4) change in customer behavior after using other channels; (5) customer behavior leading to complaints; (6) any new insights that could be obtained from the event log. Five submissions were received:

- S. Dadashnia et al. [60] clustered the traces by age and gender, and then used Disco to analyze the control-flow for each cluster, but only taking into account the most frequent events. They also focused on customers with at least 15 sessions, and for these they identified the changes in usage patterns over time. In addition, the authors created a predictive model using a deep neural network (in TensorFlow) and trained it with the sequences of user actions to predict the

next user action. For question four, they analyzed events in separate time intervals. For question five, the authors used the sequence clustering plug-in in ProM to derive the usage patterns.

- U. van der Ham [61] started by calculating some log statistics. The author obtained an unstructured process model, and also analyzed the transition from the website to other channels. Using ProM, the author tried to link complaints to the 10 previous events in the log; however, no clear pattern was identified. Using the IBM Watson web service, the author found that longer traces, many messages and visits to a specific webpage were linked to complaints. The author created a tool in Excel that plotted the tasks on a hierarchical structure, and also performed a control-flow comparison by gender using a follows matrix.
- A. Jalali [62] applied an ETL procedure and created a data cube for OLAP analysis. The author analyzed the customer behavior across dimensions to discover how customers use different services. Using chord diagrams, the author created a ProM plug-in for social network analysis (with handover of work) and assessed website visits, deriving the website patterns, along with statistics and control-flow. For question five, the author performed an analysis in the cube using several dimensions to uncover behavior that might lead to complaint. The process model for those dimensions was also provided.
- G. Janssenswillen et al. [63] performed data exploration and data preparation to start their analysis. For question one, the authors used k-means clustering based on webpages to perform trace clustering. The most common tasks were presented for each cluster and compared to the overall tasks. To address other business questions, the tasks were reclassified into 11 tasks only. Statistics and process flow (using chord diagrams) were provided to answer the questions.
- F. Heidari and N. Assy [64] preprocessed the data using a custom-built ProM plug-in to filter out consecutively repeated tasks. For question one, the patterns were classified based on active cases per month and demographic-oriented statistics. For question two, the authors characterized the change based on the frequency of visited pages over time as well as using process variant comparison in Minit. For question three, using Disco the authors identified that the transition occurs very early in the process and predicted it based on the flow. For question four, the change in behavior of customers before and after using another channel was assessed using again process variant comparison in Minit.

In this challenge, the event logs were much larger than in previous editions (e.g. one of the logs had over 7 million events and about 1 GB in size). Due to the log size, some authors had difficulties using standard process mining tools, so they focused on simpler techniques based on filtering and dividing the data into smaller event logs. There were some attempts at using more sophisticated approaches (e.g. neural networks) but these were very time-consuming, especially during the training phase.

2.7. BPI Challenge 2017

In 2017, the BPI Challenge involved the same Dutch Financial Institute as in the 2012 edition, and the event log concerned an upgraded version of the same loan application process.

In this challenge, there were three categories: student, academic and professional. There were also four business questions: (1) throughput times, in particular the time a customer is waiting for the bank and vice-versa; (2) influence of incompleteness requests on offer acceptance; (3) comparison between single-offer and multiple-offer customers; (4) any other interesting trends. Twenty-three submissions were received:

- Rodrigues et al. [65] preprocessed the log and considered only four representative variants. Using Disco, the authors obtained a process model in which activities were clustered together. The authors created a Petri net model and calculated its fitness and balanced precision in ProM (Replay Log on Petri Net for Conformance Analysis, and Check Precision based on Align-ETConformance). This model was later improved based on the results of the replay plug-in. The authors then provided in-depth statistics. To analyze the process, the authors broke down their model into smaller parts and modeled them in BPMN, where they perceived three distinct phases. Each phase was analyzed at a control-flow level, and the answer for question two was provided. Using Disco, ProM and Yasper, the authors assessed the performance perspective and answered questions one and three.
- D. Ryu et al. [66] provided statistics, and performed a dotted chart analysis in which they identified an automated system and classified tasks as human- and system-tasks. For question one, the authors calculated the throughput times per activity. For question two, the authors analyzed the frequency of task A_Incomplete and its relation to completed or cancelled cases, and concluded that if customers pass several A_Incomplete, they are more likely to accept the final offer. For question three, the number of customers who ask more than one offer was calculated with a Python script. As for the conversion rate, Disco was used to identify the target cases, and those cases were then compared with the script result. In addition, the authors performed a T-Test to assess the difference between the applicants where one offer and more than two offers were made. For question four, the authors analyzed incomplete cases and predicted the remaining time for such cases using a finite state machine miner. Using Disco, the authors also analyzed decision points in the process to figure out if there are any decision rules among them. The organizational structure was also analyzed, using hierarchical clustering in R and the organizational miner (self-organizing map mining, and default mining) in ProM. As both approaches yielded complex results, the authors used K-means clustering in ProM to obtain three clusters of users. The authors identified groups of tasks performed by each cluster. The authors then analyzed the overall flow based on the occurrence or absence of W_Shortened completion.
- S. Dadashnia et al. [67] started by obtaining a process model in Disco and describing the general workflow, comparing it to BPIC 2012. The processes were compared at control-flow

level (mainly), and case duration. The authors assessed process similarity by calculating the percentage of common nodes, percentage of common nodes and edges, graph edit distance, causal footprints, and model similarity based on behavioral profiles. Using RefMod-Miner and Disco, the authors applied a K-means clustering approach to mine reference model components from the log. Seven clusters were obtained and each of them represented a subprocess. These subprocesses were then assessed in a performance perspective (question one). For questions two and three, the authors used the Chi-square ($\chi 2$) test of independence also known as Pearson Chi-square test, and concluded that incompleteness does not lead to negative results since more offers lead to higher conversion. In addition, the authors used convolutional neural networks (using TensorFlow) with variable length inputs to predict the next process steps. The training data was generated using a case-based sliding window approach.

- L. Blevi et al. [68] combined process mining with KPMG's customer experience methodology. Using Power BI, the authors provided overall statistics. From those statistics, question three was answered. The process model was obtained using Fuzzy miner in ProM and a control-flow analysis was performed. For question one, the authors conducted a performance analysis. Using R and Microsoft Azure Machine Learning Studio, the team created predictive models (using logistic regression, random forests, and neural network) to calculate the probability of selecting an application and to predict the occurrence of A_Pending. According to the mean decrease accuracy from the random forest, the degree of incompleteness is the third most important variable.
- F. Berger [69] used the dotted chart in ProM to get an overview of the process as well as the flow of each subprocess (A_, O_, and W_) using Celonis. For question one, a performance analysis was carried out in Celonis. For question two, the target cases were filtered in Celonis and statistics were provided. For question three, the same approach of filtering and providing statistics was employed. For question four, the author suggested opportunities for improvement to work events; calculated average processing cost of an application; created and calculated KPIs (e.g. return rates) for the entire dataset, as well as its variation over time, and KPI analysis using statistical process control.
- G. Scheithauer et al. [70] started by providing an overview of the process, i.e. process models were obtained in Disco and log statistics were provided focusing on start and end tasks. For question one, the authors conducted a performance analysis in Disco. For question two, the authors identified the number of requests for additional documents per application and calculated the conversion rates based on that number. In addition, patterns in requesting additional documents and conversion rate were identified using case attributes. For question three, using Disco the authors identified applications having more than one offer; separated cases with single and multiple conversations; identified the overall conversion rate; and identified conversion rate for single- and multi-offer applications.
- S. Smith and C. Day [71] started with a performance analysis in Celonis, where they assessed throughput time to answer question one. In addition, the authors presented several statistics. For question two, the authors calculated the number of incompleteness requests as well as the

number of end states to assess the hypothesis. For question three, the number of conversations and offers was obtained to calculate the conversion rate. For question four, the authors provided a relation between the final outcome of an application and its duration.

- L. Wangikar et al. [72] obtained the process model in Celonis. The authors broke down the model into eight phases and assessed it at control-flow and performance, providing statistics for each phase, thus answering question one. In addition, the authors linked the impact of waiting time to an application success. For question two, the authors calculated the number of times the task A_Incomplete took place and the trend in approval/cancellation. The authors then considered the number of times task W_Call incomplete files as a better indicator for incompleteness. For question three, the authors assessed the number of offers regarding single- and multiple-offers creation. The conversion rate was higher for multiple offers creation. For question four, they provided recommendations about which applications should be the focus of more attention.
- U. van der Ham [73] started by comparing the 2017 process model with the 2012 process in Minit. For question one, the author conducted a performance analysis. Using Excel, the time for all transitions was counted in a directly follows matrix, validated in Celonis. Social network analysis with handover of work points out that handing over cases causes bottlenecks in the process. Two clusters of resources were identified. For question two, the author converted the process models to a sales funnel. Analyzing the funnel, the author concluded that most cancellations occur during call after offers. Using Celonis, the author counted the number of times the task W_Call incomplete files-start was performed and their relation to end states. The author concluded that an increase in the number of W_Call incomplete files does not cause clients to cancel. For question three, the number of offers was counted as well as if it came from one conversation or multiple conversations. The author concluded that offers made in different days are more likely to be accepted by the customer. In addition, the author assessed the conversion rate based on other case attributes. For question four, the author found a process unconformity (cases in A_Pending though not accepted) and credit score influence in (non-acceptance).
- E. Povalyaeva et al. [74] created a BPMN diagram of the process in Celonis based on their log analysis. Using PI-conformance in Celonis, the model had a 99% consistency. Using random forests, the authors analyzed the outcome probability against case length. A concept drift analysis was performed in ProM, and the authors concluded that the process was stable. For question one, the authors carried out a performance analysis in Disco and provided a histogram showing the probability of each outcome. For question two, a random forest in Python was trained with cases having at least one request for completion. A list of feature importance was analyzed with statistical significance tests, and the number of completion calls was not a relevant feature. For question three, the authors analyzed the correlation of cases having single and multiple offers with their outcome.
- P. Badakhshan et al. [75] provided log statistics and obtained a process model in Disco. For question one, a performance analysis was carried out using Celonis. For question two, the

authors used filters (attribute and follower types) in Disco to retrieve successful cases having one occurrence of task A_Incomplete, no occurrences, and more than one. For question three, the authors identified cases having multiple offers from single and multiple conversations, and the relation between those cases to the final outcome.

- D. Jeong et al. [76] started by providing log statistics. The authors then obtained a process model and broke it down into three parts (application, offering, and validating documents and decision taking). For question one, a performance analysis was carried out by the team for each of the process parts. For question two, the authors conducted a statistical analysis on the ratio of end points of the process to the frequency of task A_Incomplete, using Disco. In addition, the authors tried to link incompleteness to loan goals, but no significant pattern was identified. Using MATLAB, the authors tried to derive classification rules (support vector machine, decision tree, random forests, etc.) for incompleteness. However, the accuracy was below 60% and no conclusion could be drawn. For question three, process models were obtained in Disco according to the number of offers; then the differences in tasks and paths were analyzed. For question four, the authors assessed pending cases vs. not pending cases (statistics, control-flow, and classification rules), case endpoints using K-means clustering, and social network analysis (originator by task matrix in ProM; next k-means clustering in MATLAB), resources execution, and waiting time analysis.
- A. Bolt [77] used transition systems to represent the lifecycle of offers. For question one, a performance analysis was conducted. For question four, the author developed a plug-in for ProM (process variant finder) which creates an event-annotated transition system from an event log and performs recursive partitioning by conditional inference using the data attributes available in the events. The idea is to determine if the variability of any event attribute can be reduced by splitting any other event attribute without any extra knowledge of the company or their customers. If so, such variability reductions can be considered as process variants. The developed plug-in was used to detect variants in control-flow, performance and all event data attributes not related to control-flow nor performance.
- E. Varas and C. Iglesias [78] preprocessed the log and broke down the overall process in two phases (application creation; application validation and offers creation). Process models for each phase were created. In order to calculate the throughput times, the authors used the Pandas library from Python to add time-related metrics to the event log. Afterwards, the authors analyzed the flow as well as the performance of both phases, answering question one and two. For question three, the authors filtered cases in Disco and analyzed them with Pandas.
- K. Park et al. [79] obtained a process model using Disco and ProM. For question one, the authors carried out a performance analysis using both tools. For question two, using Python the authors extracted a subset of cases to analyze in ProM. Using the plug-in Add Identities to Log, the authors found that the frequency of tasks W_Validating application and A_Validating is high when the loan is granted. For question three, the authors calculated the number of offers and its relation to task A_Pending using Python and ProM (inductive miner). Python was used to split the log (into single- and multiple-offer) and ProM to discover process models.

- M. Sani and H. Sotudeh [80] started by providing statistics and preprocessing the event log. The team used many process discovery algorithms available in ProM (alpha, alpha ++, ETM, Heuristic Miner, Inductive Miner, ILP* and their own algorithm) and chose the best discovery model based on soundness and F-measure. For question one, the authors used the ProM plug-in Replay Log on Petri net for Conformance Analysis. In addition, a performance analysis was carried out in Disco. For question two, the authors analyzed the number of times an application was requested for completion and its relation to the outcome. For question three, the authors analyzed the endpoint of applications based on the number of offers. For question four, the authors performed a dotted chart analysis, resource analysis, decision tree analysis in RapidProM, and patterns affecting fraud detection.
- C. Oh et al. [81] used Disco to get an overview of the process. For question one, a performance analysis was carried out. For question two, the authors analyzed the relation between the frequency of task A_Incomplete and the number of successful cases. In addition, the authors checked the relation between frequency of incompleteness and mean case duration, loan goal, and requested amount. For question three, the authors analyzed the relation between number of offers and loan granting. In addition, control-flow, performance, and requested amount were also analyzed. For question four, the authors assessed the organizational perspective in two ways: (a) in ProM, using social network analysis (handover, subcontracting and working together); and (b) segmenting tasks with their associated resources to make clusters based on the type of task.
- M. Sant'Anna and J. Leite [82] started by analyzing the throughput time in Disco. For question two, the authors filtered the log in Celonis and analyzed the relation between endpoints of the process with cases having multiple requests to completion. For question three, the authors first used follower filters in Disco to identify how many customers ask more than one offer; whether offers were created in the same conversation; and the conversion rate. For question four, statistics were provided regarding case attributes.
- H. Jang et al. [83] provided overall statistics about the log using Disco and ProM (statistics tab and dotted chart, respectively). After identifying the endpoints of the process, the authors created process models for both event logs. The authors then performed an organizational perspective analysis in which they assessed: the number of tasks per resource; the number of resource per number of tasks; the ratio of tasks by subprocess (A_, O_, W_) per resource; the number of resource per tasks; social network analysis with doing similar tasks developed by the team in R and dotted chart analysis in ProM; handover of work in ProM; task execution time per resource; and resource characterization per event attributes. For question one, the team did a performance analysis. For question two, the authors compared the process outcome according to the frequency of task A_Incomplete. In addition, the relation between loan goal and incompleteness was also analyzed. For question three, after having classified cases by frequency of task W_Create Offer, the authors compared the successful outcome according to the number of offers.

- M. Kwon et al. [84] started performing a dotted chart analysis in ProM followed by a process discovery in Disco. For question one, a performance analysis was carried out. For question two, the authors counted the frequency of incompleteness for each case and its relation to the final outcome. For question three, the authors analyzed the relation between number of offers and the final outcome.
- A. Carvallo et al. [85] provided overall log statistics and preprocessed it in Python. Using Celonis, the team obtained the process model. The authors then performed a control-flow analysis in Celonis and ProM (trace alignment plugin). For question one, a performance analysis was carried out in Celonis and Disco. For question two, the authors analyzed the influence of the frequency of task A_Incomplete on the final outcome. For question three, the authors analyzed the relation between the number of offers for each application and the proportion of each outcome per incompleteness frequency. In addition, the authors also considered the throughput time in their analysis. For question three, the authors divided cases into single- and multiple-offers and analyzed them. For question four, the authors assessed the influence of customers and resources – social network analysis in Celonis and ProM – to the process flow and throughput time. In addition, business rules were created and assessed at resource and flow level.
- F. Rojos et al. [86] preprocessed the log to reduce its overall complexity. The authors identified the bottlenecks and their causes using heuristic miner and performance analysis with Petri net plug-ins in ProM. For question one, performance analysis was done in Disco and Celonis. In addition, the authors analyzed if it was possible to predict if a customer would cancel the application, based on the correlation between requested amount and the execution time. In addition, the authors created a Python script to calculate how many resources could be removed until a task could no longer be performed. A social network analysis in Celonis validated their findings. Workload distribution was also assessed. The authors created a recommender system to facilitate user profiling and mitigate workflow bottlenecks. For question two, cases having no incompleteness requests were filtered out in Disco. The authors counted the number of tasks to complete an application. They concluded that 2/3 of applicants drop after five cycles of document requests. A performance analysis of those cases in ProM showed that they have a bottleneck. For question three, Disco and Python were used to identify single- or multiple-offer cases, as well as whether offers were from single or multiple conversations. Those cases were then linked to their final outcome.
- R. Carmona et al. [87] preprocessed the log. For question one, a performance analysis was carried out in Disco and Celonis. In addition, the authors assessed the performance perspective taking loan goal into consideration. For question two, the authors analyzed the relation between number of calls and process endpoint. For question three, the authors analyzed the relation between the number of offers made and successful endpoint. For question four, the authors created a probabilistic model to predict applications more likely to be cancelled. For that, social network analysis using similar tasks in ProM was used, as well as the organizational plugin. In

addition, to be able to predict behavior from new customers, the authors used the inductive miner in ProM.

In this challenge, there was a lot of process mining analysis, combined with data mining and machine learning techniques, such as neural networks, random forests, etc. We saw a variety of process mining tools being used (Disco, ProM, Celonis, Minit, etc.). Despite the large number of submissions, the analysis was mostly focused on the business questions, with similar results being reported by all the teams.

2.8. BPI Challenge 2018

In 2018, the BPI Challenge involved an event log from the European Agricultural Guarantee Fund, in Germany. The event log pertains to the yearly allocation of direct payment to farmers. The workflow is based on document types. Each document has a state that allows some actions to be performed.

In this challenge, there were four business questions: (1) detection of undesired cases (late payment or reopening); (2) improving the sampling of applications that are selected for inspection; (3) differences between departments and relation to undesired outcomes; (4) differences across time. There were three winning submissions:

- S. Pauwels and T. Calders [88] used the competition to test their method to detect concept drift. Their model, an extended Dynamic Bayesian Network, is able to capture conditional and functional dependencies. It provides three visual aids: a trace-score plot; a drift-plot; and an attribute-density plot (which characterizes the data in terms of the different attributes). After preprocessing the event log, they trained their model which detected two drift points, thus answering question four. For question three, the authors trained their model using the events from one department and tested it with the other departments. In addition, the authors analyzed individual traces and clusters of traces using their attribute-density plot to find differences.
- L. Wangikar et al. [89] started by providing log statistics and then preprocessing the log. Using Celonis, the authors obtained the process model. For question one, the authors created a predictive modeling framework to detect applications having undesired outcomes. A binomial logistic model was created for each time interval and year (day 0, up to day 90, and up to day 180). The authors created two sets of models: one with data from current and previous year, and another without the data from the previous year. For question two, the same approach was followed. However, the time interval was from day 0 to June 26. For question three, the authors analyzed the data by departments and year using statistics, throughput time, and conformance checking. For question four, the authors used the Concept Drift Analysis plug-in in ProM to identify control-flow changes across years. To identify the changes causing the drift, they used myInvenio's conformance checking feature. In addition, the authors quantified the impact of changes in the process.

- J. Brils et al. [90] started by obtaining a process model in Disco and calculating log statistics. The authors identified a third undesired outcome: reopened cases with additional payment. After processing the event log with Python, the authors assessed case duration, case distribution, and department workload, relating them to undesired outcomes. In addition, the undesired outcomes were compared to non-late payment cases at control-flow and performance perspectives. For question one, the authors created features for outcome prediction and used machine learning techniques in RapidMiner (Naive Bayes, Generalized Linear Model, Logistic Regression, Deep Learning, Decision Trees, Random Forests, and Gradient Boosted Trees). Gradient Boosted Trees was the best classifier with an accuracy of 95.52% for late payments, and 82.61% for reopened cases.

In this challenge, data mining techniques were predominant over process mining. This is due to the nature of some of the business questions, which focused on prediction. These questions received the most attention from participants. Process mining was used to get an idea of the control-flow, and to study concept drift in the process. Subsequent analysis was performed using machine learning techniques.

Chapter 3
Comparative Analysis

In this chapter, we present a comparative analysis of the techniques and tools that were most used in the BPI challenges 2011-2018. Our main goal is to understand, in general terms, why some particular techniques are preferred over others, and whether the use of such techniques depends on the business questions or whether there are relevant reasons to use those techniques independently of the specific business questions for each BPI challenge.

In particular, we look into the tools used across BPI challenges as well as the different uses of the same tool. We expect to see multiple tools being used and its usage variating across BPI challenges. We try to present a reasoning about the usage of those tools, i.e. whether the use of such tools is related to the nature of the data, to the nature of the business process, or even to personal preference.

3.1. Techniques

Typically, process mining focuses on the control-flow, organizational, and performance perspectives. Some BPI challenges involve all of these perspectives, while others involve only some of them. In general, however, all BPI challenges go beyond those perspectives to include additional types of analysis, as shown in Table 1.

Control-flow perspective techniques are the most used. This is to be expected since no process flow is ever provided, and one needs to understand the business process to be analysed. All teams used at least one control-flow mining technique, either to understand the business process or because a business question required it. Heuristics-based techniques and fuzzy miners were the most used for process discovery.

	2011	2012	2013	2014	2015	2016	2017	2018	Total
Control-flow perspective	X	X	X	X	X	X	X	X	**8**
Trace clustering	X	X	X	X		X	X	X	**7**
Social Network Analysis	X	X	X	X	X	X	X		**7**
Performance perspective	X	X	X	X	X		X	X	**7**
Log statistics		X	X	X	X	X	X	X	**7**
Conformance checking		X	X	X	X		X	X	**6**
Predictive modelling				X	X	X	X	X	**5**
Dotted chart analysis		X	X		X	X	X		**5**
Plotting / visualization			X	X		X	X	X	**5**
Trace alignment	X	X	X				X		**4**
Concept drift analysis					X		X	X	**3**
Spreadsheet analysis		X	X			X			**3**

Table 1: Process mining techniques used in the BPI challenges 2011–2018

Clustering is also among the most used techniques. Here the main reason is the fact that the process models derived were too complex to be understood and there was a need to make smaller and more homogenous groups to mine an understandable process model.

Although many authors worked on the performance perspective, most participants only analyzed it thoroughly if a business question required it. Otherwise, basic performance statistics were provided, which were obtained automatically from the tools.

The social network mining techniques have gained an increased popularity, mostly because of the implementation of those techniques in the ProM framework, which made it easy to use and was made possible by the availability of data in the competitions.

Log statistics have been widely collected, and exploratory analysis is the main reason participants use them. Then techniques such as predictive modelling, data plotting/visualization have also gained special attention. Predictive modelling techniques have been used mainly when a business question required it, but they have also been helpful to study the control-flow perspective.

3.2. Tools

In Table 2, a list of the most popular tools in the BPI challenges is presented. Disco (commercial) and ProM (academic) were definitely the most used. Despite the sponsoring of other tool providers in some editions of the BPI challenge (e.g. Celonis, Minit), Disco was still the most used. There seems to be plenty of room for commercial tools in the field of process mining.

	2011	2012	2013	2014	2015	2016	2017	2018	Total
Number of submissions	**3**	**6**	**12**	**11**	**9**	**5**	**23**	**3**	**72**
Disco		3	10	9	6	4	18	1	**51**
ProM	3	5	6	3	7	5	17	3	**49**
Excel		1	7	6	4	2	10		**30**
R			2	3	1	2	8		**16**
Celonis						1	13	1	**15**
Python			1				8	1	**10**
WEKA				3	2	1	2		**8**
Oracle		1	1	2	1		2		**7**
RapidMiner			1	1	1		2	1	**6**
Java			1	2	1	1			**5**
SQL Server				2		1	1		**4**
Minit					1	1	1		**3**
C#				1	1		1		**3**

Table 2: Process mining tools used in the BPI challenges 2011–2018

In general, it was observed that the use of these tools follows a cascading pattern:

1. Disco and/or ProM are used to get an overview of the process.
2. Additional statistics are collected from the event log using tools such as Excel and R.
3. Other tools are selected depending on the challenge and on the nature of the business questions. In some cases, authors have used database engines (e.g. Oracle, SQL Server) to compute some statistics.

3.3. ProM plug-ins

Since ProM is a framework that includes a wide variety of plug-ins, it is interesting to check which of those plug-ins have been most used. In Table 3, the focus is on the use of ProM plug-ins. Here, the heuristics miner, the dotted chart analysis, and the social network miner take the podium, with a significant lead over the remaining ones.

	2011	2012	2013	2014	2015	2016	2017	2018	Total
Heuristics miner	2	3	3	1	2		2		**13**
Dotted chart analysis		2	1		4	1	5		**13**
Social network miner	1	1	3		2	1	3		**11**
Inductive miner					1		4		**5**
Fuzzy miner			1	2			2		**5**
Alpha miner		1	1				2		**4**
Sequence clustering			1			2	1		**4**
Trace alignment	1	1	1				1		**4**
Concept drift					1		1	1	**3**
Organizational miner					1		2		**3**
Filter log simple heurist.		1	1		1				**3**
Guide Tree Miner	1	1							**2**
LTL-checker	1	1							**2**
Originator-by-task	1						1		**2**
Pattern abstractions	1	1							**2**
Trace align. w/ guide tree	1	1							**2**

Table 3: ProM plugins used in the BPI challenges 2011–2018

It is interesting to note that the social network miner is one of the top plugins, and has been used even in BPI challenges where there were no business questions involving specifically the organizational perspective. It seems that the social network miner is very useful to complement and/or corroborate results obtained in other analysis perspectives.

Also, it appears that the fuzzy miner would be among the most popular plugins, if it were not for some competition from Disco, which provides somewhat similar but improved functionality. Finally, it is worth noting that trace alignment, conformance checking, and concept drift have been playing an increasingly more important role.

3.4. Data Mining

As we came across several data mining / machine learning techniques during our survey, it would be interesting to analyze their usage in the BPI challenges. Table 4 presents the reader with the data mining / machine learning techniques used across the BPI challenges.

Although only two BPI challenges (2014 and 2018) included a prediction goal, we can see in Table 4 that this type of analysis is performed in editions where it was not apparently required. Decision Trees are by far the most used technique; two times more used than the second technique. Participants used tools (for example, RapidMiner) which provide the implementation of the technique and had only to be parametrized and/or customized to the data. Decision Trees were mostly used for classification purposes across the BPI challenges. For example, D. Jeong et al. [76] used them to classify incompleteness of a loan application.

Sometimes, Decision Trees are used to address specific business questions; other times, they are used to complement the analysis. Half the time Decision Trees were used, it was due to the fact that there was a business question requiring to predict a behavior; in the other half it was used, the aim was to provide additional insight to the analysis previously carried out during the process mining phase.

Logistic Regression and Random Forests also take part in the podium. We have noted that they are both the second choice after Decision Trees. They were also used for classification tasks as well as for predicting behavior.

In general, data mining and/or machine learning techniques tend to be used to enhance the business process analysis. When there is some aspect that cannot be explained by process mining techniques alone, data mining/machine learning techniques come as a mean to understand those issues.

In conclusion, the BPI challenges have been not only a testbed for process mining techniques, but have also brought many other approaches into the realm of process mining. Besides the analysis of the control-flow, organizational and performance perspectives, the business questions associated with the BPI challenges often require the use of data mining and/or machine learning techniques. Besides supervised machine learning, unsupervised techniques also play a role, especially with the use of clustering as a means of preprocessing to better understand the process and facilitate the analysis of the event log.

	2011	2012	2013	2014	2015	2016	2017	2018	Total
Decision Trees		1		4	3		3	1	12
Logistic Regression				2			2	2	6
Random Forest				1			3	1	5
Neural Network						1	2		3
Linear Regression				2					2
Support Vector Machine				1			1		2
Sequential Pattern Mining				1	1				2
Sequence Classification				1	1				2
Naïve Bayes				1				1	2
Ada Boost				1					1
Apriori Algorithm				1					1
Association Rules				1					1
Multilayer Perceptron Classifier				1					1
Binary Segmentation					1				1
K-means Clustering						1			1
Dynamic Bayesian Network								1	1
Generalized Linear Model								1	1
Deep Learning								1	1
Gradient Boosted Trees								1	1

Table 4: Data Mining/Machine Learning techniques used in the BPI challenges 2011–2018

3.5. Proposed approach

As we have seen, the analysis of real-world event logs requires a process mining component and a data mining component. The process mining component focuses on studying the sequence of events, whereas the data mining component focuses on studying the attributes associated to each process instance.

It becomes clear that it is necessary to make use of both components to analyze an event log. Based on the survey and the comparative analysis we have presented, we propose an approach that takes these two components into account, and in addition these components work iteratively.

What we propose is to look at the cases, from the most frequent ones (which represent most of the behavior captures in the event log) to the less frequent ones (which represent deviations in the process).

For each sequence of events (from the most frequent to the less frequent) we extract a control-flow model, and for each new sequence we compare it to the previous sequences, focusing on the differences that arise in the control-flow.

To study and explain these differences, we make use of data mining techniques, in particular decision trees, in order to classify process instances as belonging to one sequence or to another. The decisions inferred by the decision tree allow us to explain the reasons behind the differences observed between the sequences.

This two-component analysis (control-flow model extraction and decision tree induction) is iteratively repeated, for each new sequence, until all sequences in the event log have been traversed.

For those sequences that represent very rare behavior (for example, that represent less than 1% of the total number of cases) the analysis has less and less impact, and it can be terminated at that point because it will not provide much new information about the process flow.

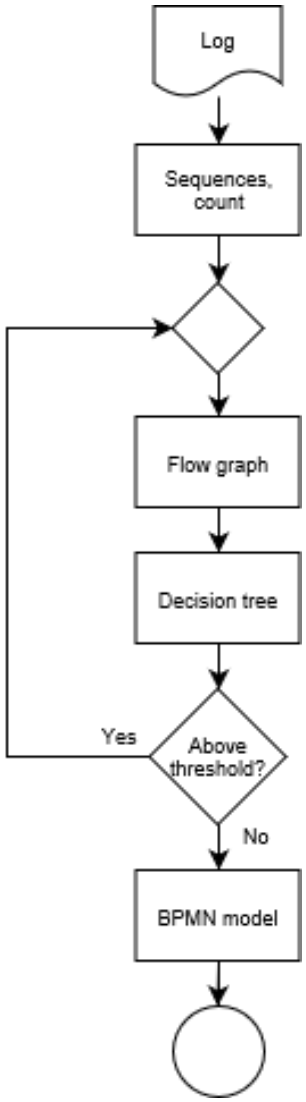

Figure 1: Process discovery approach

The proposed approach can then be described in the form of a flowchart shown in *Figure 1*.

First, we assess if the event log has all required attributes to perform the analysis. We check if it contains event attributes as well as case attributes. If the event log contains only event attributes, then only traditional process mining analysis can be carried out; in this case, we are unable to perform the data mining analysis as there are no case attributes to explain the deviations in the process.

Second, we count the frequency of all sequences of activities and order them from highest to lowest count.

Third, using a control-flow mining technique such as a transition counting algorithm [91], we derive a process model for the sequence with the highest count. For the remaining sequences, we derive the new process model (combining all previous behavior and comparing it to the new one) by studying the differences using a decision tree classifier (using feature importance as a scoring metric).

Lastly, we stop considering new sequences of activities when their count fall below a threshold. This is done to avoid performing the analysis with rare behavior which adds little or no new information to the analysis. At the end, using the knowledge collected from this analysis, we draw a BPMN diagram of the process.

In the next chapter, we illustrate the application of this approach to the BPI challenge 2019.

Chapter 4
The BPI Challenge 2019

In this chapter, we analyze the event log from the BPI challenge 2019 with a view towards understanding the business process. From the insights gained in our survey of past BPI challenges, we conclude that an event log is best analyzed using both process mining and data mining techniques. This is exactly what we do in this chapter, by extracting the process flow with process mining techniques, and by trying to understand that flow with data mining techniques, in particular decision trees.

4.1. Business Process

The BPI challenge 2019 involves an event log from a large multinational company operating in The Netherlands in the area of coatings and paints. The event log concerns the purchase-to-pay process. Each event refers to a step in the purchase-to-pay process from the purchase ordering until the goods are received and payed. In addition, each process instance has a number of attributes that characterize the item that is being purchased.

In the BPI challenge 2019, there are three business questions of interest for the process owner:

(1) How does the process model look like? One or more process models are desired, since there could be a different process model for each category of purchase item.
(2) What is the throughput of the purchase-to-pay process with respect to specific activities such as goods receipt, invoice receipt and clear invoice?
(3) Which instances of the purchase-to-pay process stand out from the log? Where are deviations in the processes? Which customers cause a lot of rework?

Question (1) relates to the control-flow perspective. We are explicitly asked to come up with one or more process models. We focus on simplicity and generalization as the main quality criteria of our models, while striving to produce models that can replay most of the behavior captured in the event log.

Question (2) focuses on the performance perspective. We use Disco to compute some statistics regarding the overall process. These results are presented in the appendix.

Question (3) also relates to the control-flow. The focus is not in what are the common scenarios, but what is the infrequent behavior in the process. Purchase orders may stand out from the log for several reasons: strange flow, unusual throughput time, data inconsistency, etc.

There are four categories of purchase items in the event log (to be explained below):

- 3-way matching, invoice after goods receipt;
- 3-way matching, invoice before goods receipt;
- 2-way matching;
- and consignment.

There is a clear way to distinguish each item category, since there is an attribute indicating it in the event log. The process flow is different for each of item category, hereafter called a subprocess. These differences can be explained based on the following business rules:

- In the *3-way matching, invoice after goods receipt* subprocess, the value in the goods receipt message, the value in the invoice receipt and the value put during creation of the item should all match.
- In the *3-way matching, invoice before goods receipt* subprocess, purchase items do not require the goods receipt message. When the goods are received and the value in the invoice receipt matches the value at creation, an invoice can be cleared. Otherwise, it remains blocked until a user or a batch process unblocks it.
- In the *2-way matching* subprocess, there is no goods receipt message. The value in the invoice should match the value at creation.
- In the *consignment* subprocess, there are no invoices at all for purchase items.

Additionally, there can be multiple goods receipt messages and corresponding clear invoice messages for each purchase item. For example, for university payment fees which can be payed in seven parcels, the purchase document has one item (the fee), but there can be seven goods receipt messages and their corresponding clear invoice messages, each clear invoice having 1/7 of the total amount.

In our analysis, we study the process flow for each subprocess separately. Our main goal is to extract and understand the behavior of the business process. Therefore, our focus will be more on business questions (1) and (3), and less on business question (2). While analyzing the business process to address question (1), we will also be identifying deviations that are useful to address question (3). As for question (2), it is more related to the performance perspective, and we address it in the appendix.

4.2. The Event Log

Over 1.5 million events are captured from the purchase-to-pay process in 2018. The event log is anonymized; however, some semantics were left in the data to help understanding the context of the purchase document and items.

There is a total of 20 attributes in the event log. However, we have used only a subset of these attributes; some attributes (e.g. *source*) were single-valued, while others (e.g. *vendor*) had too many values to be informative. In Table 5 we present the attributes that we selected for our analysis.

Case ID	This is the case identifier. It combines both the purchase document and the anonymized item identifiers.
Activity	This is the task that was performed.
Resource	The user that performed a task. It can be a normal-user (human actor) or a batch-user (automated process).
Timestamp	The date and time an event was recorded.
Item category	This indicates the subprocess (3-way, 2-way or consignment).
Spend classification text	A text field that explains the purchase item class.
Spend area text	A text field that explains the purchase item area.
Sub spend area text	An extra text field that further explains the purchase item area.
Item type	The type of the purchase item.
Cumulative net worth	The total value of a purchase document.
Document type	The type of the document.

Table 5: Data attributes in the BPI challenge 2019 event log

For illustrative purposes, Table 6 shows the attributes recorded in the event log for the case ID 4507000228_00010. This case ID is obtained by concatenating the purchase document ID (4507000228) together with an item ID (00010) within the document.

Table 6 Table 5shows the sequence of events for this case ID horizontally. As can be seen, the activity, resource, and timestamp change, since they represent the sequence of events for this case ID. However, the remaining attributes do not change, i.e. they have the same values for all events in that case ID.

This case has five activities, performed by five different resources (four humans and one unrecorded, represented as *NONE*). It started on 02/01/2018 at 06:28 and finished on 22/02/2018 at 15:15.

For the process mining analysis, where we will be extracting the process flow, we will use the *activity*, *resource*, and *timestamp* information. This is the information that is typically used by process mining techniques, to analyze the control-flow, organizational and performance perspectives.

Case ID	4507000228_00010	4507000228_00010	4507000228_00010	4507000228_00010	4507000228_00010
Activity	Create Purchase Order Item	Vendor creates invoice	Record Goods Receipt	Record Invoice Receipt	Clear Invoice
Resource	user_036	NONE	user_029	user_024	user_002
Timestamp	02/01/2018 06:28:00	03/01/2018 22:59:00	05/01/2018 10:10:00	05/01/2018 13:10:00	22/02/2018 15:15:00
Item category	3-way match, invoice before GR	3-way match, invoice before GR	3-way match, invoice before GR	3-way match, invoice before GR	3-way match, invoice before GR
Spend classification text	NPR	NPR	NPR	NPR	NPR
Spend area text	Sales	Sales	Sales	Sales	Sales
Sub spend area text	Products for Resale	Products for Resale	Products for Resale	Products for Resale	Products for Resale
Item type	Standard	Standard	Standard	Standard	Standard
Cumulative net worth	110.0	110.0	110.0	110.0	110.0
Document type	Standard PO	Standard PO	Standard PO	Standard PO	Standard PO

Table 6: Log excerpt for case 4507000228_00010

On the other hand, for the data mining analysis, we will be using the remaining attributes that have the same values within the same case ID. The main idea is that the differences in the values of these attributes across case IDs may be useful to understand the differences in the sequence of events for those case IDs.

4.3. Preliminary analysis

There are two types of data captured in the event log, namely case events and case attributes. The event data explain the process flow; each event comprises four fields: case ID, activity, resource, and timestamp. These fields are used to perform a process mining analysis, since there is a sequence of events for each case ID. Also, in some event logs, the resource information may not be captured, affecting the type of analysis that can be carried out during the process mining analysis phase. However, in this event log the resource field is available, which enables an analysis on the organizational perspective, if that would be necessary.

The case attributes have a single value for each process instance. These attributes can be used to perform a data mining or machine learning analysis. Usually, these attributes are used to explain some uncommon or rare behavior seen during the process mining analysis. The attributes that we will use for this type of analysis are the last seven attributes shown in Table 5.

There are 251 734 cases placed in 2018 as well as 1 595 923 events in the event log, leading to an average of about six events per case. These events represent the purchase order process.

The event log was supposed to capture only purchase orders in 2018. However, an inspection of the event log showed that there are events recorded as old as 1943 and as late as 2020, pointing to a data recording inconsistency. We decided to include all the data present in the event log as only ~3% of it is outside 2018. If such data would be discarded, 202 cases and 45 455 events would be lost.

Before delving into a deeper analysis, we decided to perform a control-flow analysis into the event log to validate if those out-of-2018 data could have a noticeable effect in the process. We compared the three processes flows: (a) all event log data, (b) only 2018 data, and (c) data outside 2018. We concluded that those ~3% of data do not introduce any strange behavior into the process flow.

The process discovery technique used to mine the process flows, in Figure 2 and Figure 3, is a transition counting algorithm based on the count of activities transitions within the same process instance, implemented in Python [91]. The transition counts are stored in a transition matrix which is then used to generate the process models as a graph.

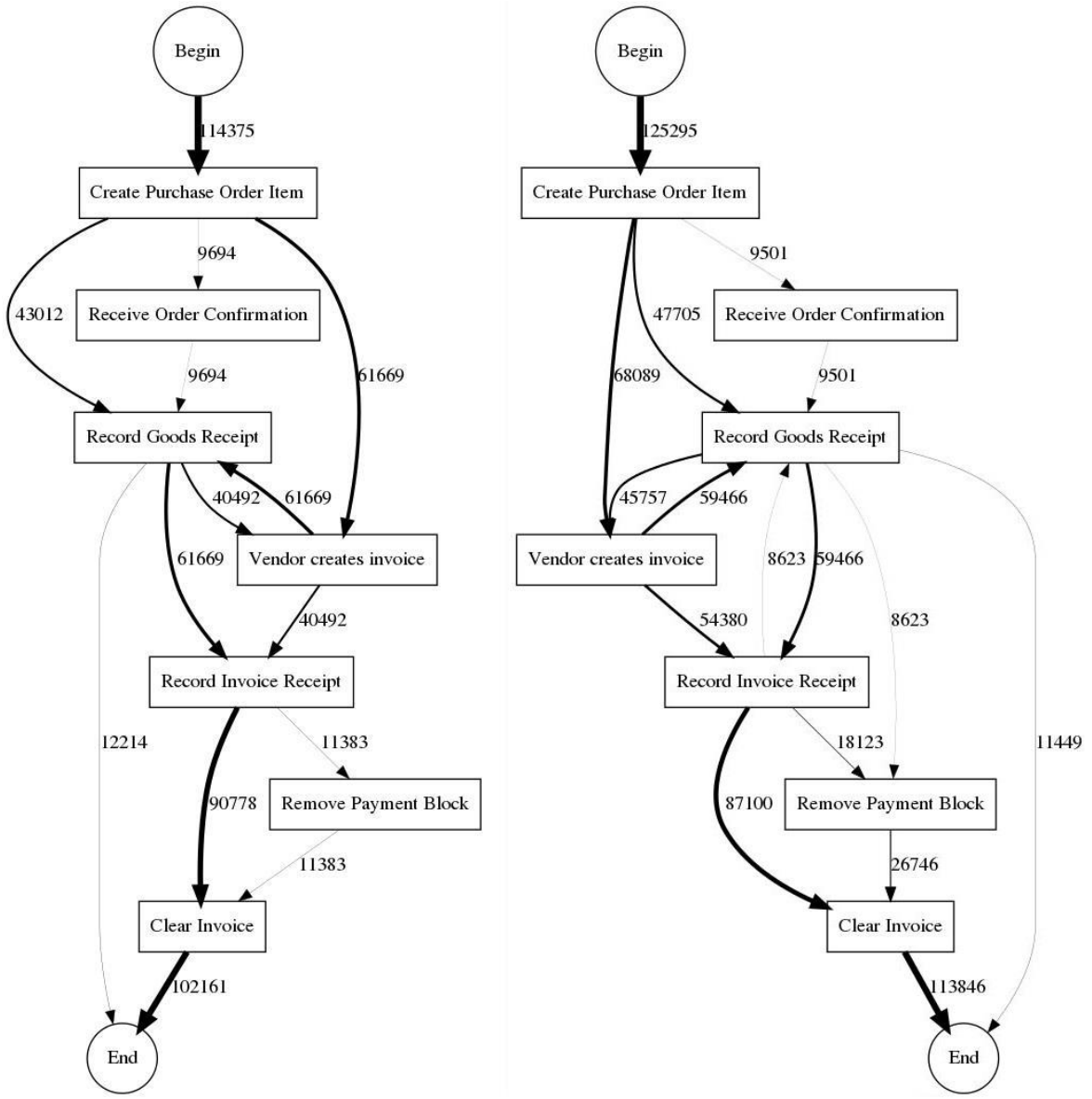

Figure 2: Overall process map (left) and process map for 2018 only (right)

Figure 2 shows two process models: on left, the overall process model using the whole event log, and on right the overall process using just the 2018 data. Both models show the most common activities and we can see that the activities are the same.

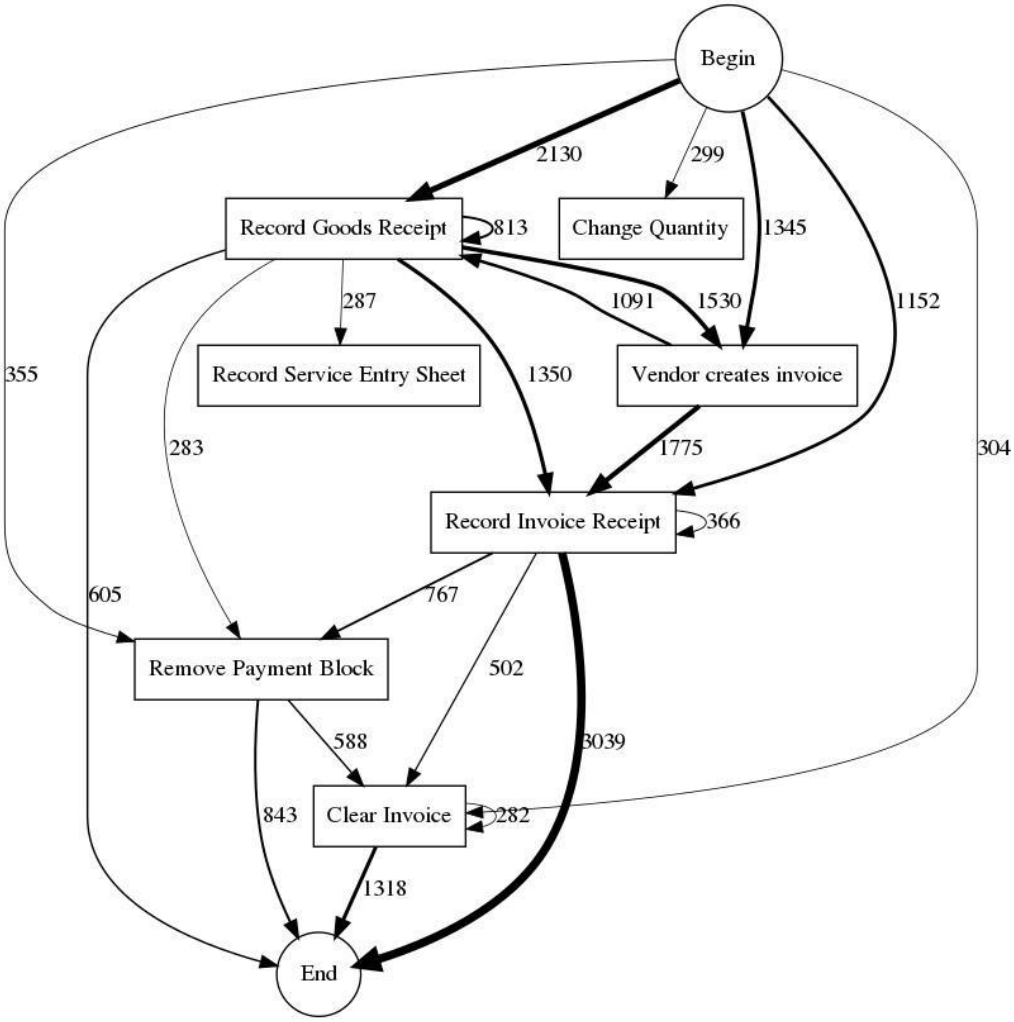

Figure 3: Process map for out-of-2018 events

The process model for the out-of-2018 data, depicted in Figure 3, shows that even then the process model is very similar. The process starting point is different from the one shown in Figure 2 because in Figure 3 the cases refer mostly to the process *2-way matching,* whereas the process model in Figure 2 is mostly composed by *3-way matching*. The process model is less structured because those cases are not entirely complete as they have events captured in the 2018 year.

4.4. Process mining analysis

When analyzing an event log with unstructured behavior, it makes sense to start the analysis by looking at the process instances that represent the most common behavior in the event log, and only then look at other process instances that represent less common, or even rare behavior in the event log.

Based on this idea, we start deriving a control-flow model by first taking into account the most common behavior, and only then incorporate in the model other variants that represent less frequent behavior. To extract the control-flow model, we used the transition counting technique to obtain a process flow.

Our approach can be described as follows:

1. We obtain the sequence of activities for each case ID.
2. We group case IDs by their sequence of activities, such that two case IDs with the same sequence end up in the same group.
3. We count how many case IDs there are in each group.
4. We sort the sequences according to their count, from highest to lowest.
5. For the first sequence (highest count) we derive a process flow based on the transition counting technique.
6. For the remaining sequences (with lowers counts) we incorporate each new sequence in the model, by deriving a process flow based on all the previous sequences plus the new one.
7. Every time we consider a new sequence, and we obtain a new process flow, we study the differences between this new process flow and the previous one.
8. We stop considering new sequences when their count falls below a certain threshold. The threshold that we used was 1% of the total count (i.e. total number of case IDs in the event log). If a sequence represents less than 1% of the behavior recorded in the event log, then we do not consider such sequence.
9. After having the process flow resulting from the previous analysis, we create a BPMN diagram for the process.

This analysis approach was applied separately for each item category, that we consider as being a different subprocess. In the following paragraphs, we present the results obtained for the *3-way matching, invoice before goods receipt* subprocess, which represents most of the behavior captured in the event log (~88% of the process instances). The methodology is identical for the other subprocesses and the results are similar.

The *3-way matching, invoice before goods receipt* subprocess has a total of 221 010 cases, so in our approach a sequence representing less than 2 210 cases will not be included as it is considered as uncommon or rare behavior. This subprocess contains 16 sequences above the 1% threshold.

Figure 4 (left) shows the flow for the first (most common) sequence in the subprocess. It contains five activities, which can be regarded as being the process backbone. This sequence represents 47 959 cases (~22% of the sequences in this subprocess). In Figure 4 (right), we include the second (most common) sequence, and we now see an increase of 29 410 cases, totaling 77 369 cases (~35% of the sequences in this subprocess).

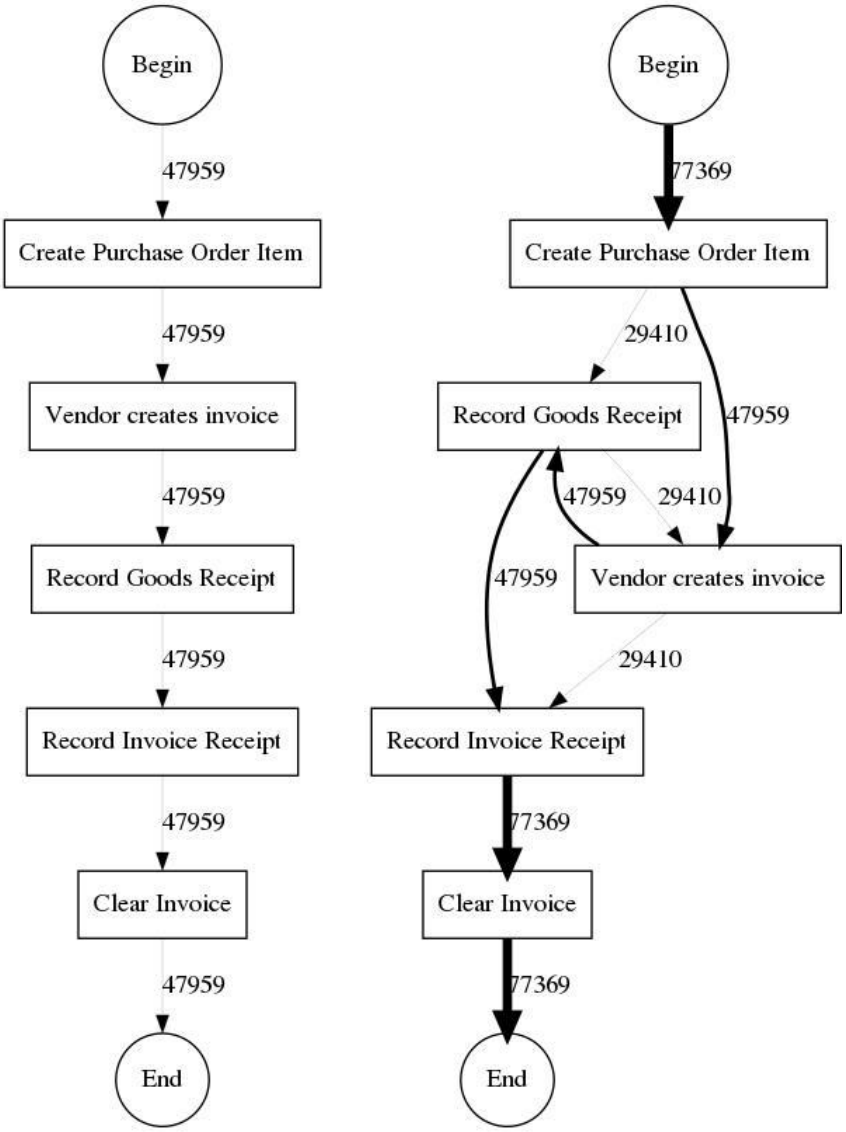

Figure 4: Top-1 sequence flow (left) vs top-2 sequences flow (right)

In this new flow in Figure 4 (right), we see that the activity *Record Goods Receipt* can occur either before or after *Vendor creates invoice*. Considering the high frequency of these two possibilities, we conclude that *Record Goods Receipt* and *Vendor creates invoice* are performed in parallel.

We now consider the third (most common) sequence. From the 77 369 cases in first top-2 sequences, there is an increase of 10 854 cases (see Figure 5), totaling 88 223 cases (~40% of the sequences in this subprocess).

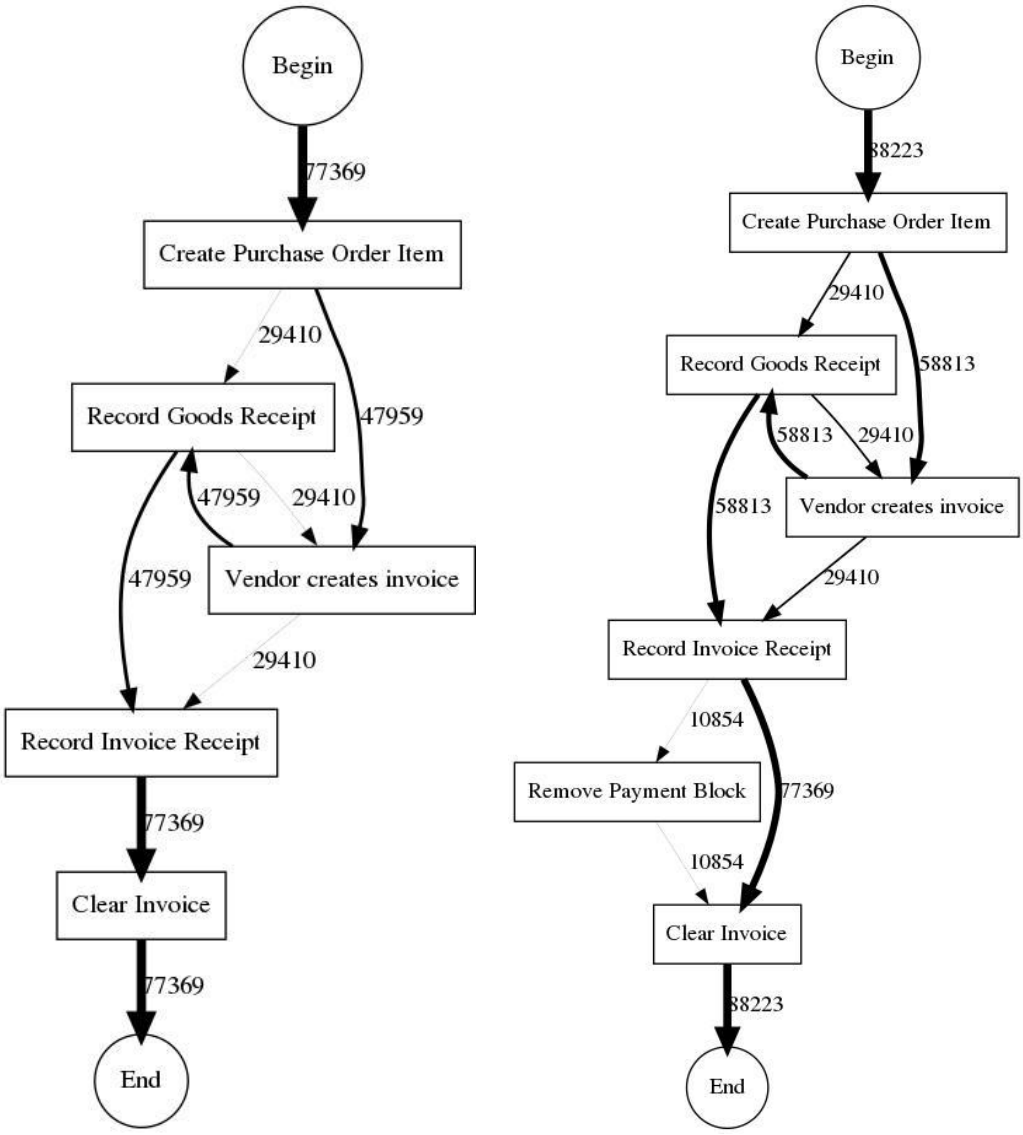

Figure 5: Top-2 sequences flow (left) vs top-3 sequences flow (right)

This new sequence, Figure 5 (right), introduces a new activity in the subprocess flow, namely *Remove Payment Block*. This suggests that the new activity is optional, and it might happen after *Record Goods Receipt* and before *Clear Invoice*.

We now consider the fourth (most common) sequence. In Figure 6 we see an increase of 9 694 cases, totaling 97 917 cases (~44% of the sequences in this subprocess).

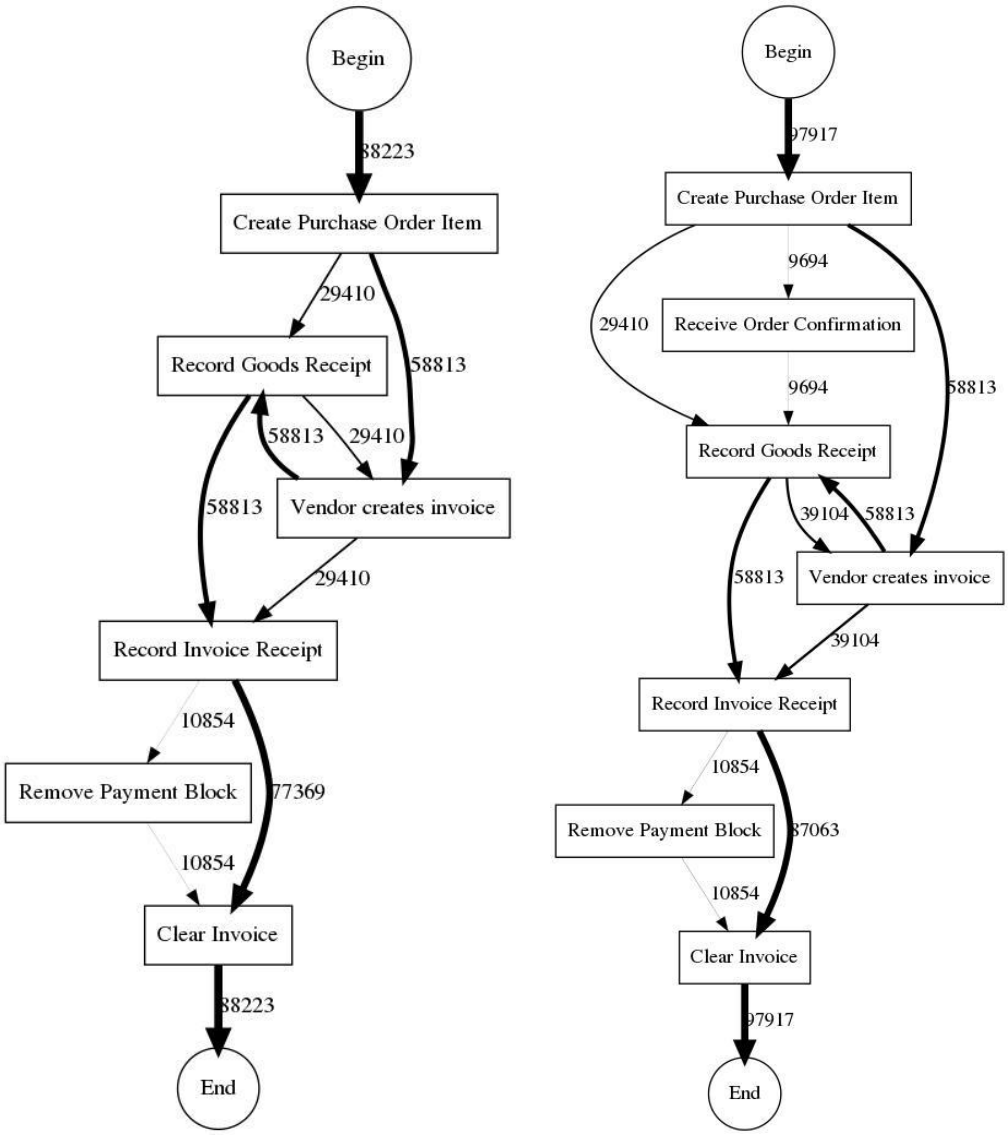

Figure 6: Top-3 sequences flow (left) vs top-4 sequences flow (right)

This new sequence in Figure 6 (right) introduces a new activity into the subprocess flow, namely *Receive Order Confirmation*. This suggests that the new activity is optional, and it might happen after *Create Purchase Order Item* and before *Record Goods Receipt*.

We now consider the fifth (most common) sequence. In Figure 7 we see an increase of 8 835 cases, totaling 106 752 cases (~48% of the sequences in this subprocess).

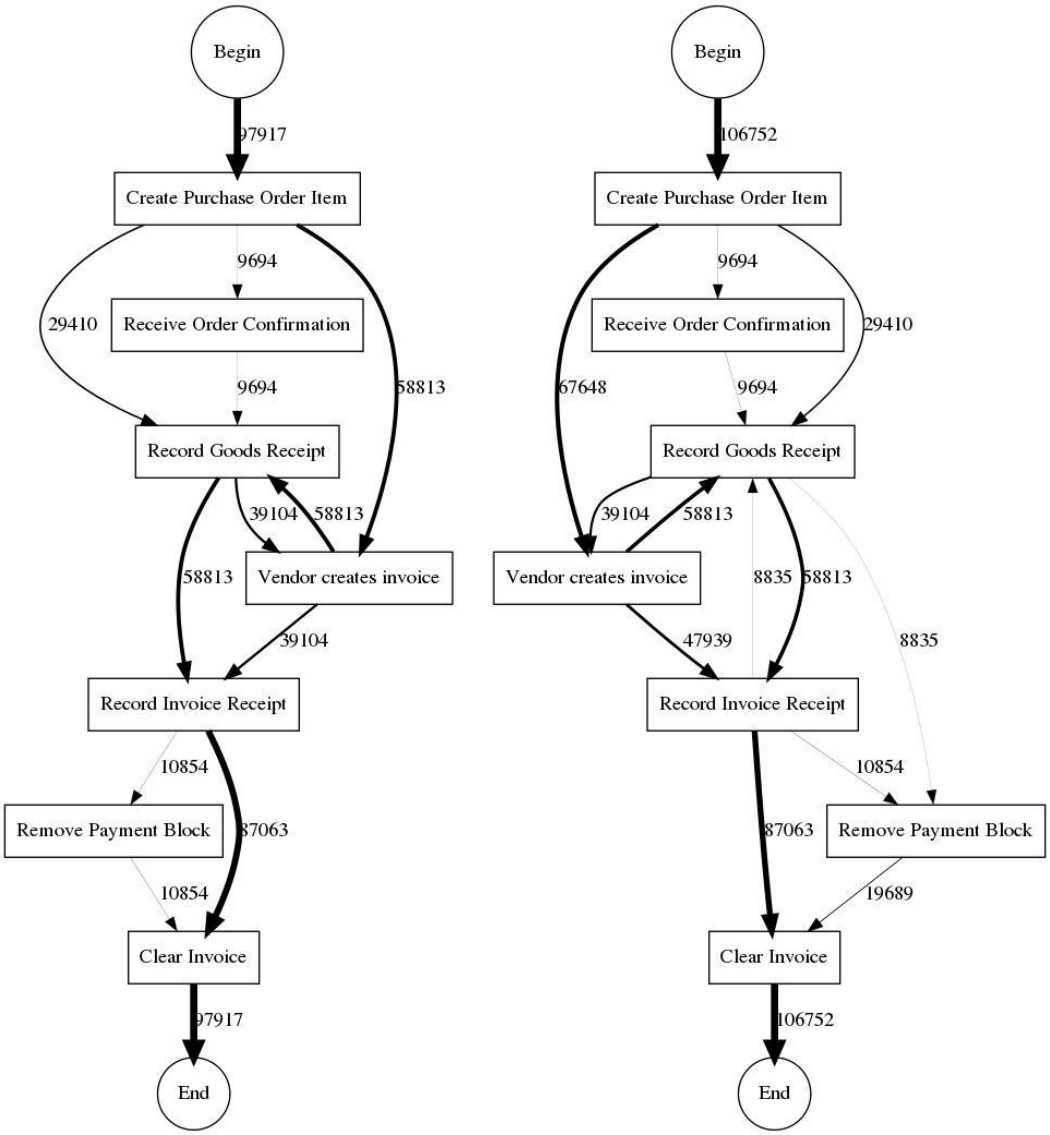

Figure 7: Top-4 sequences flow (left) vs top-5 sequences flow (right)

This new sequence introduces a new behavior into the subprocess flow, *Record Invoice Receipt* now comes before *Record Goods Receipt*. As this sequence represents only 4% of this subprocess, we decided not to consider the parallelism as being between *Vendor creates invoice, Record Invoice Receipt, and Record Goods Receipt*, but rather only between *Vendor creates invoice* and *Record Invoice Receipt* and then *Record Invoice Receipt* is performed.

It should be noted that this decision is based on the fact that 4% is a rare behavior. However, if this percentage increases considerably the intepretation regarding the parallelism should be reverted.

We now consider the sixth (most common) sequence. In Figure 8 we see an increase of 8 673 cases, totaling 115 425 cases (~52% of the sequences in this subprocess).

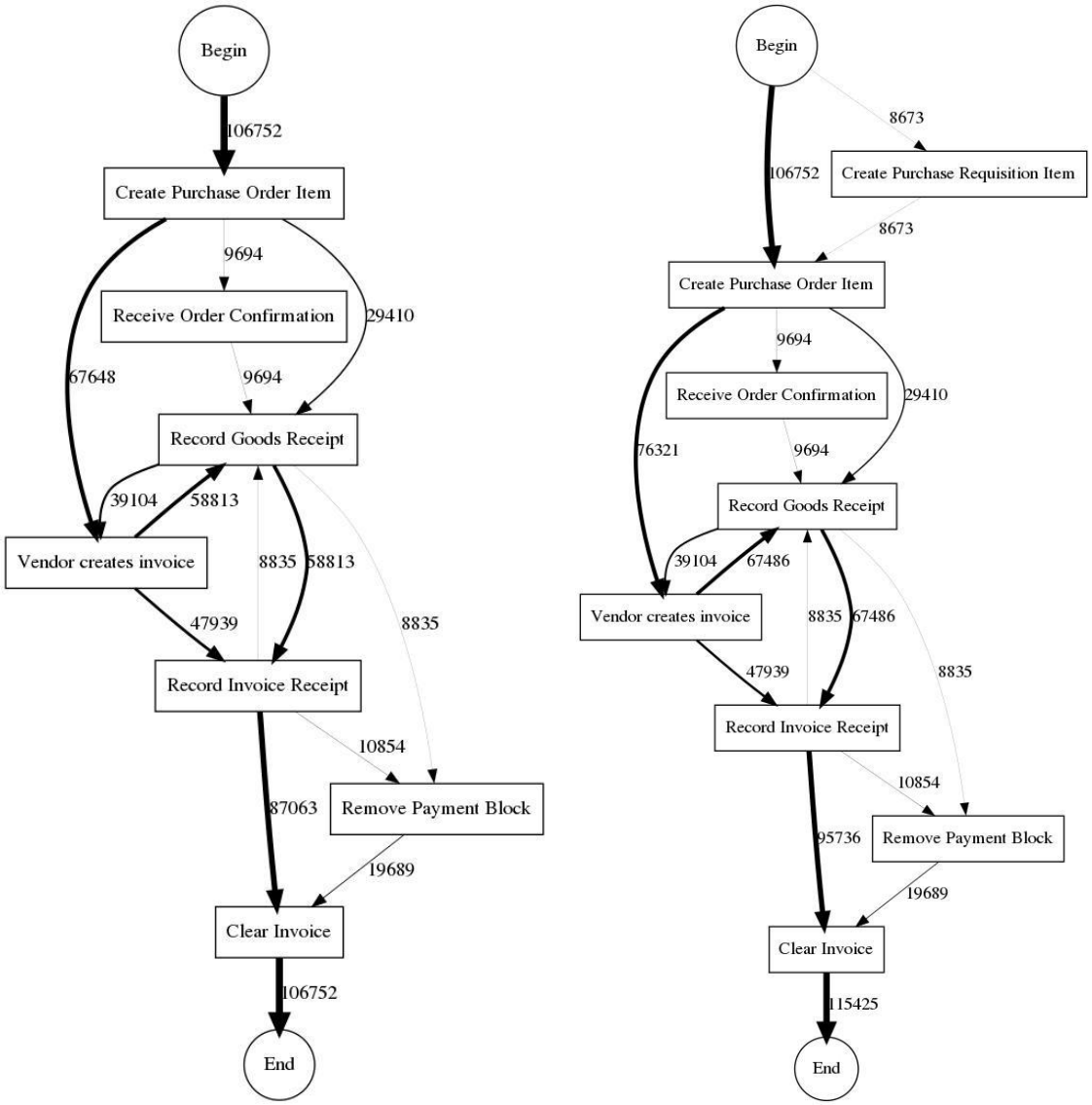

Figure 8: Top-5 sequences flow (left) vs top-6 sequences flow (right)

This new sequence introduces a new activity into the subprocess flow, namely *Create Purchase Requisition Item*. This suggests that the new activity is an optional starting activity and it flows to the other possible starting activity *Create Purchase Order Item*.

We now consider the seventh (most common) sequence. In Figure 9 we see an increase of 7 694 cases, totaling 123 119 cases (~56% of the sequences in this subprocess).

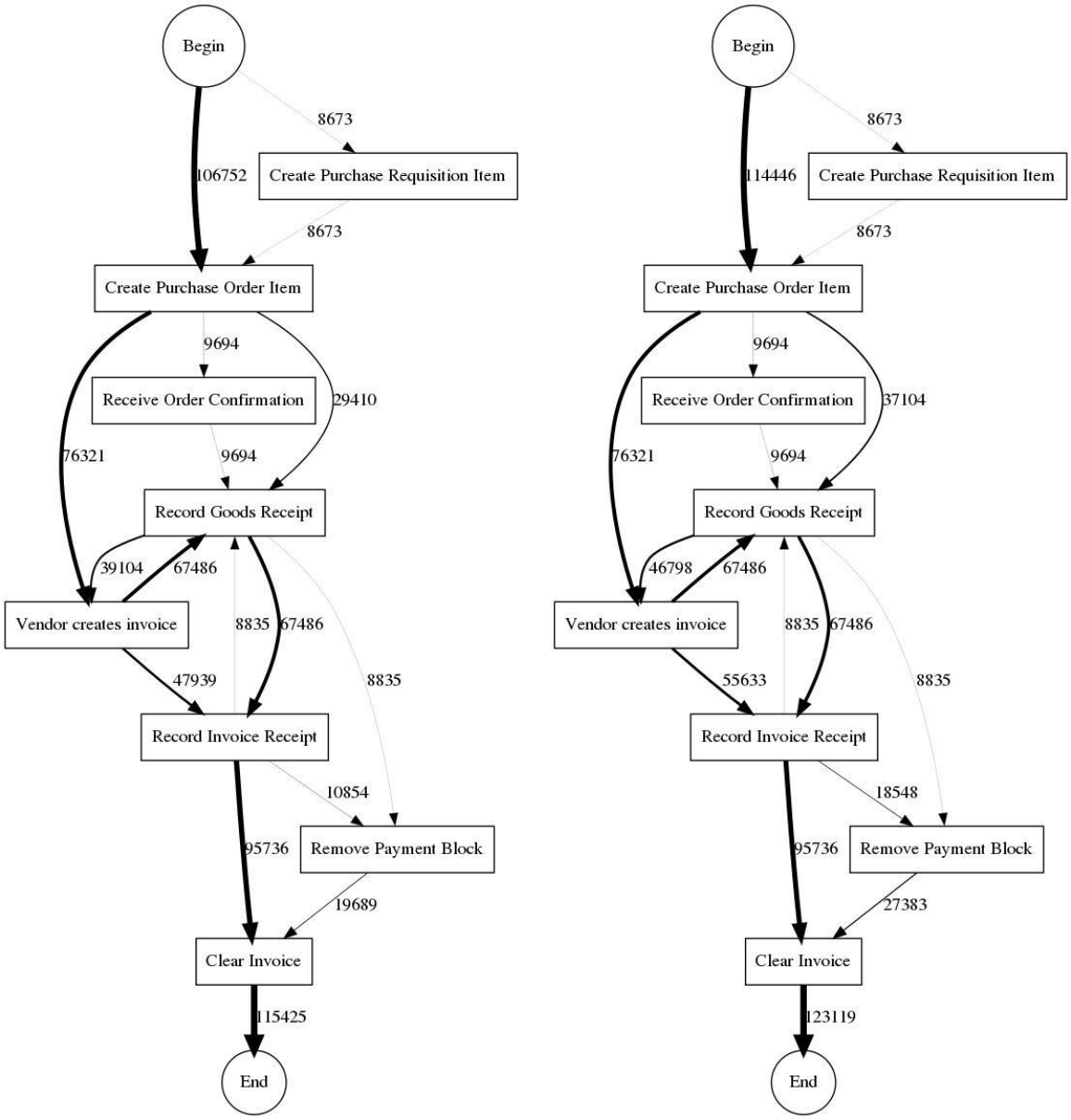

Figure 9: Top-6 sequences flow (left) vs top-7 sequences flow (right)

This new sequence introduces no new behavior into the subprocess flow.

We now consider the eight (most common) sequence. In Figure 10 we see an increase of 4 807 cases, totaling 127 926 cases (~58% of the sequences in this subprocess).

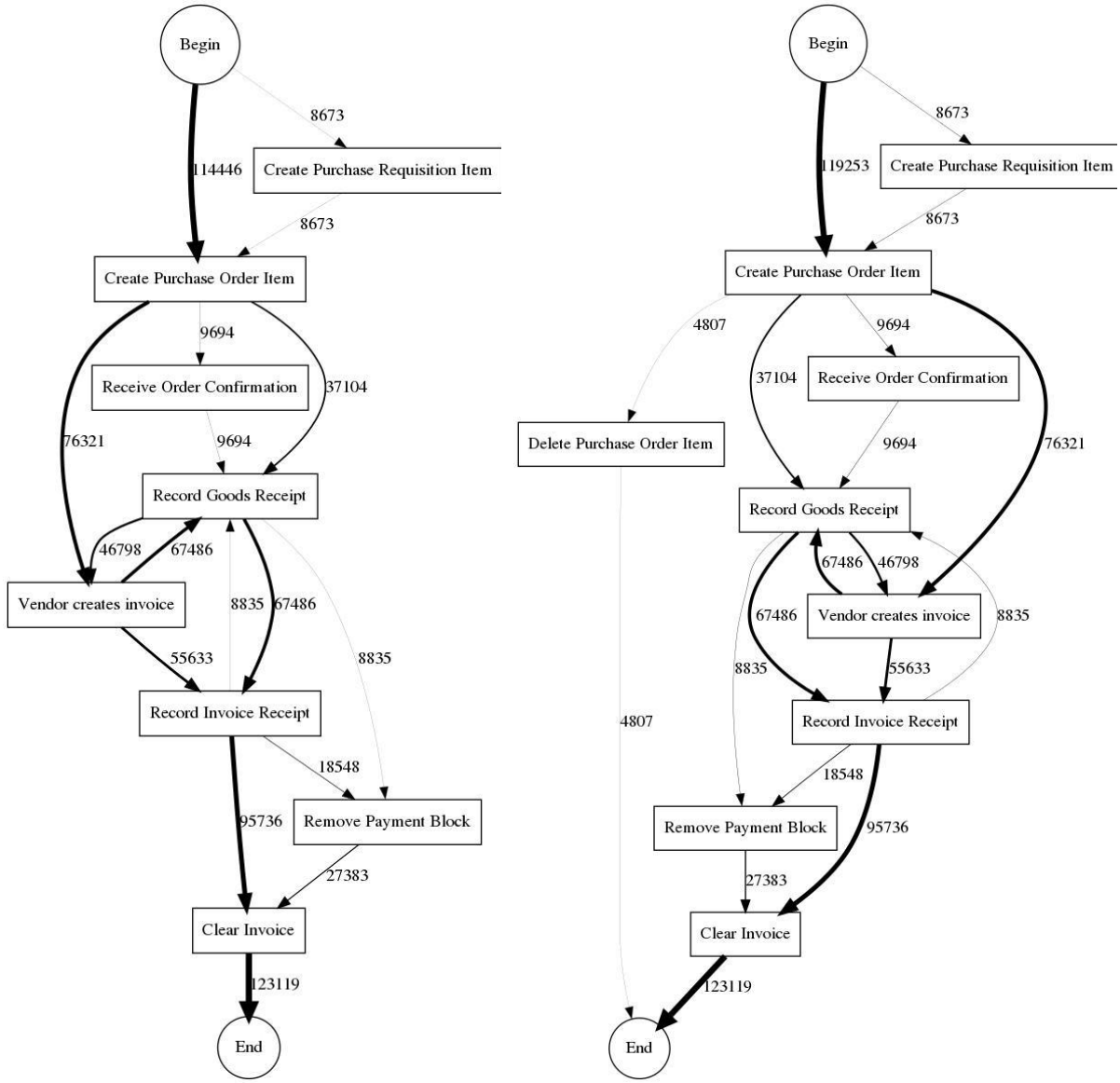

Figure 10: Top-7 sequences flow (left) vs top-8 sequences flow (right)

This new sequence introduces a new activity into the subprocess flow, namely *Delete Purchase Order Item*. This suggests that the new activity is performed after a checking, i.e. probably there is a decision if a purchase order is valid and if not, it is deleted.

This analysis has been carried out further by considering more sequences, but the observed behavior did not show any significant change to the overall subprocess flow. Based on the previous analysis, we developed a BPMN model for this subprocess. The BPMN model is shown in Figure 11. This model does not capture all the detailed behavior, but only the main behavior that is comprehensible and easy to explain to a business user, in order to convey a main idea about how the subprocess works.

It should be noted that there no process mining techniques to extract a BPMN model directly from an event log. This type of model can only be obtained by systematizing the insights obtained about the behavior of the business process. This is exactly what we have done here: after analyzing the behavior sequence-by-sequence, we have identified the parallel activities and the optional activities in the process. The BPMN model represents a distillation of that knowledge into a single process model that captures the essential behavior of the process.

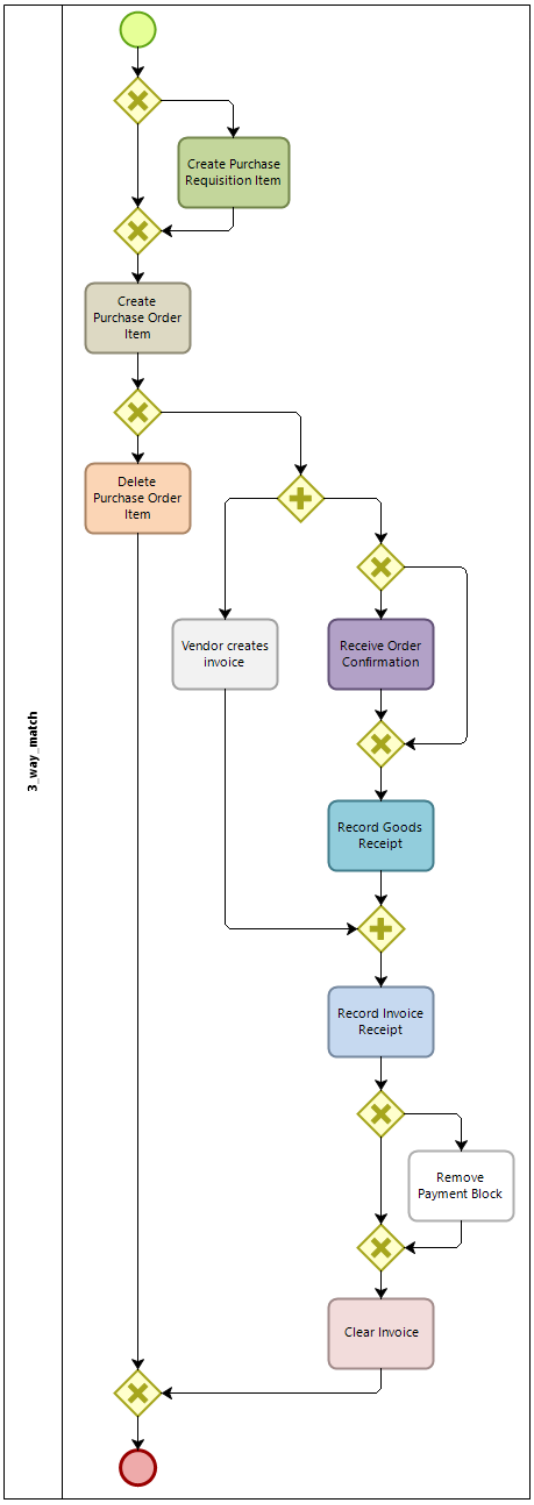

Figure 11: BPMN process model

4.5. Data mining analysis

In the previous analysis, we found many differences from sequence to sequence. Therefore, we employ a data mining analysis to further investigate the reasons behind those differences. We saw in Table 4 in Chapter 3 that decision trees were the most used data mining technique, so we decided to use them to classify the changes between sequences.

Our classification approach can be described as follows:

1. We obtain the sequence of activities for each case ID.
2. We group case IDs by their sequence of activities, such that two case IDs with the same sequence end up in the same group.
3. We count how many case IDs there are in each group.
4. We sort the sequences according to their count, from highest to lowest.
5. For the first sequence (highest count) we do nothing as there is only one behavior.
6. For the remaining sequences (with lowers counts) we use a Decision Tree Classifier to fit the data into one of two classes: one class represents all previous sequences, and the other class represents the new sequence.
7. We calculate the feature importance. The feature importance is a metric that provides a scoring for each attribute in the data and indicates how important an attribute is to decide between classes, i.e. the more a feature is used to decide, the higher its importance.
8. Every time we consider a new sequence, we study the differences between this new sequence and all the previous sequences.
9. We stop considering new sequences when their count falls below a certain threshold. The threshold that we used was 1% of the total count (i.e. total number of case IDs in the event log). If a sequence represents less than 1% of the behavior recorded in the event log, then we do not consider such sequence.

This analysis approach was applied separately for each item category, that we consider as being a different subprocess. In the following paragraphs, we present the results obtained for the *3-way matching, invoice before goods receipt* subprocess, which represents most of the behavior captured in the event log (~88% of the process instances). The methodology is identical for the other subprocesses and the results are similar.

The first sequence has 47 959 cases and is the following:

Create Purchase Order Item → Vendor creates invoice → Record Goods Receipt → Record Invoice Receipt → Clear Invoice

The second sequence has 29 410 cases and is the following:

Create Purchase Order Item → Record Goods Receipt → Vendor creates invoice → Record Invoice Receipt → Clear Invoice

The decision tree that was obtained from the classification of case IDs into these two classes (i.e. sequences) is presented in Figure 12.

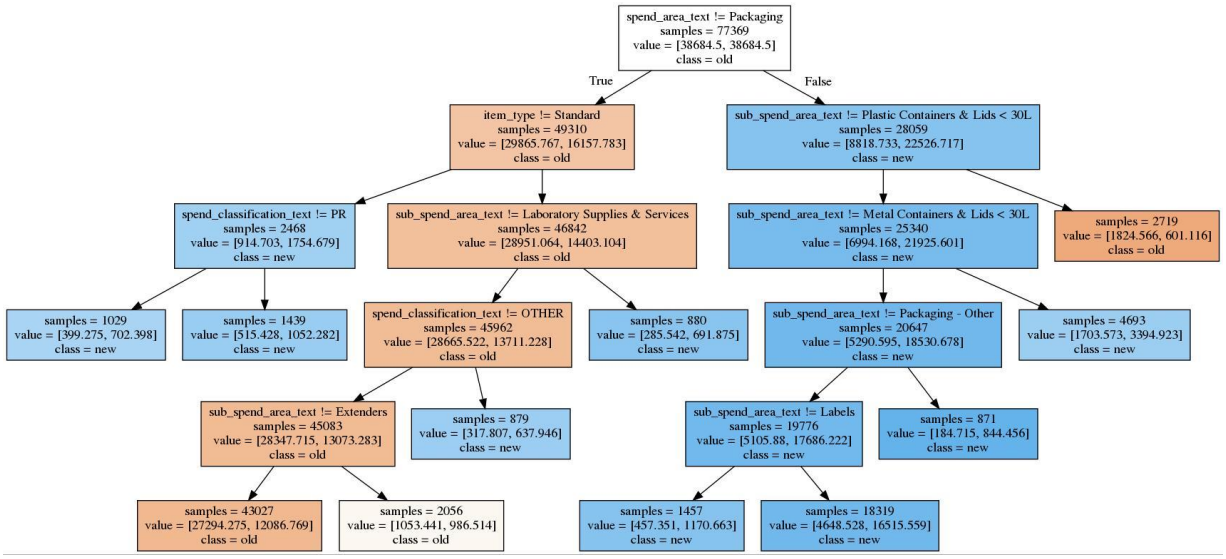

Figure 12: Decision tree for the top-2 sequences

According to the induced decision tree, the change between the two classes was introduced by the following main reasons (see Table 7).

Ranking	Feature importance	Case attribute	Attribute value	Previous percentage	New percentage
1	0.672480	Spend area text	Packaging	22.8%	58.2%
2	0.155579	Sub spend area text	Plastic Containers & Lids < 30L	4.7%	1.6%
3	0.070944	Item type	Standard	97.6%	95.5%
4	0.037662	Sub spend area text	Laboratory Supplies & Services	0.8%	1.8%
5	0.030872	Spend area text	Others	0.8%	1.6%

Table 7: Feature importance in the classifier for top-2 sequences

From this analysis, we conclude that *spend area text* being *Packaging* is the main reason for the new behavior, since it has a feature importance of 0.672480. In practical terms, this means that *Packaging* items can be handled differently in the process (*Record Goods Receipt* can occur before *Vendor creates invoice*).

The previous two sequences comprise 77 369 cases. The new flow, third sequence, has 10 854 cases and is the following:

Create Purchase Order Item → Vendor creates invoice → Record Goods Receipt → Record Invoice Receipt → Remove Payment Block → Clear Invoice

The decision tree that was obtained from the classification of case IDs into these two classes (i.e. sequences) is presented in Figure 13.

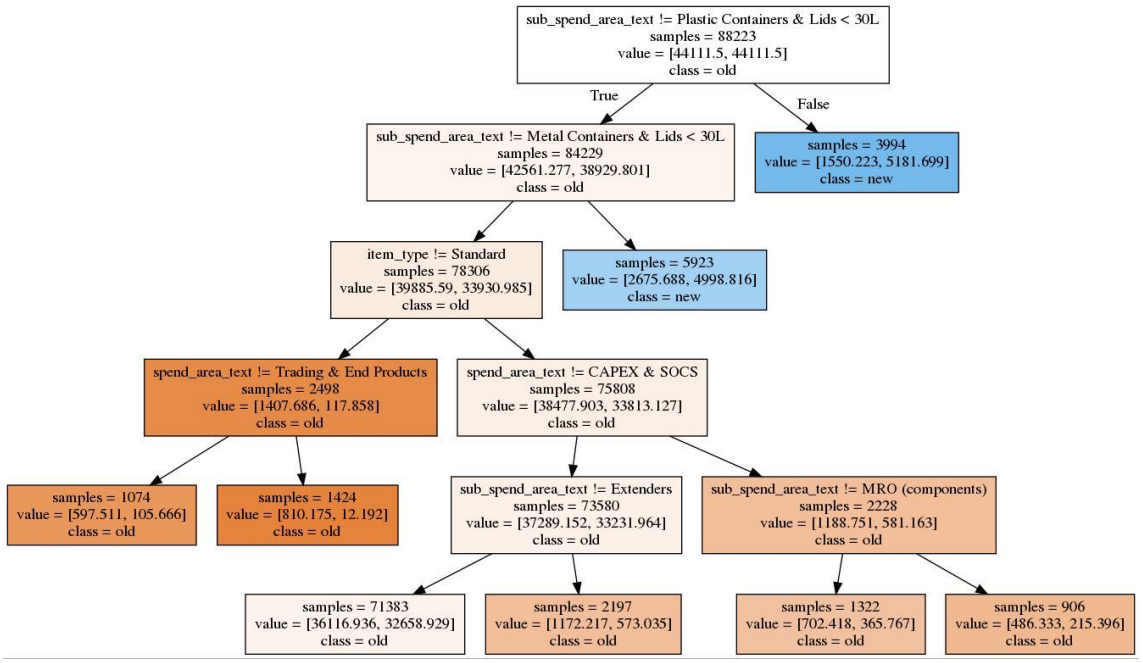

Figure 13: Decision tree for the top-3 sequences

According to the induced decision tree, the change between the two classes was introduced by the following main reasons (see Table 8).

Ranking	Feature importance	Case attribute	Attribute value	Previous percentage	New percentage
1	0.485216	Sub spend area text	Plastic Containers & Lids < 30L	3.5%	11.7%
2	0.233764	Sub spend area text	Metal Containers & Lids < 30L	6.1%	11.3%
3	0.208472	Item type	Standard	96.8%	99.7%
4	0.033440	Sub spend area text	Extenders	2.7%	1.3%
5	0.032256	Spend area text	CAPEX & SOCS	2.8%	1.4%

Table 8: Feature importance in the classifier for top-3 sequences

From this analysis, we conclude that *sub spend area text* being *Plastic Containers & Lids < 30L* is the main reason for the new behavior, since it has a feature importance of 0.485216. In practical terms, this means that *Plastic Containers & Lids < 30L* items can be handled differently in the process (Remove Payment Block can occur after *Record Invoice Receipt* and before *Clear Invoice*).

The old sequence is now comprised by the previous three sequences (88 223 cases). The new flow, fourth sequence, has 9 694 cases and is the following:

Create Purchase Order Item → Receive Order Confirmation → Record Goods Receipt → Vendor creates invoice → Record Invoice Receipt → Clear Invoice

The decision tree that was obtained from the classification of case IDs into these two classes (i.e. sequences) is presented in Figure 14.

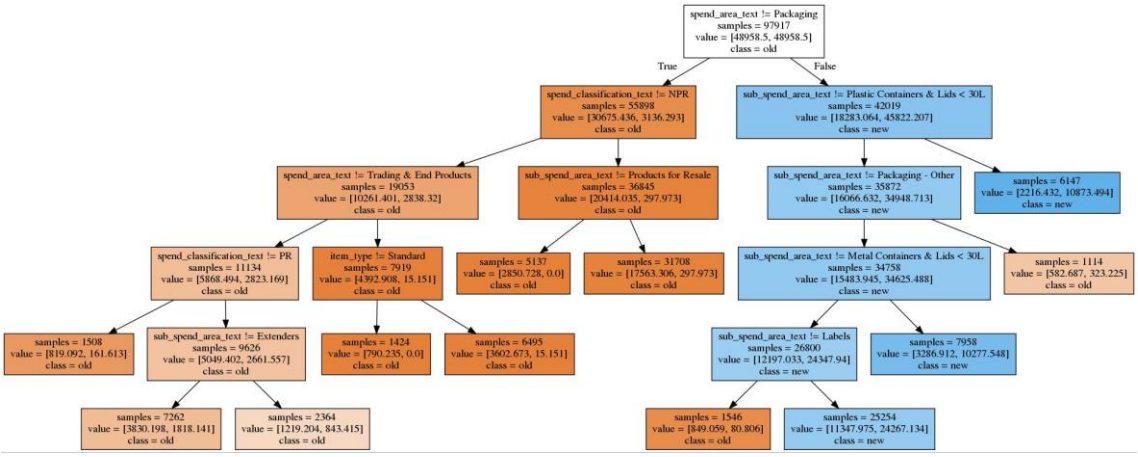

Figure 14: Decision tree for the top-4 sequences

According to the induced decision tree, the change between the two classes was introduced by the following main reasons (see Table 9).

Ranking	Feature importance	Case attribute	Attribute value	Previous percentage	New percentage
1	0.860021	Spend area text	Packaging	37.3%	93.6%
2	0.032969	Spend classification text	NPR	41.7%	0.6%
3	0.032156	Sub spend area text	Labels	23.2%	49.6%
4	0.030331	Spend area text	Trading & End Products	9.0%	0.0%
5	0.022178	Sub spend area text	Plastic Containers & Lids < 30L	4.5%	22.2%

Table 9: Feature importance in the classifier for top-4 sequences

From this analysis, we conclude that *spend area text* being *Packaging* is the main reason for the new behavior, since it has a feature importance of 0.860021. In practical terms, this means that *Packaging* items can be handled differently in the process (*Receive Order Confirmation* can occur after *Create Purchase Order Item* and before *Record Goods Receipt*).

The old sequence is now comprised by the previous four sequences (97 917 cases). The new flow, fifth sequence, has 8 835 cases and is the following:

Create Purchase Order Item → Vendor creates invoice → Record Invoice Receipt → Record Goods Receipt → Remove Payment Block → Clear Invoice

The decision tree that was obtained from the classification of case IDs into these two classes (i.e. sequences) is presented in Figure 15.

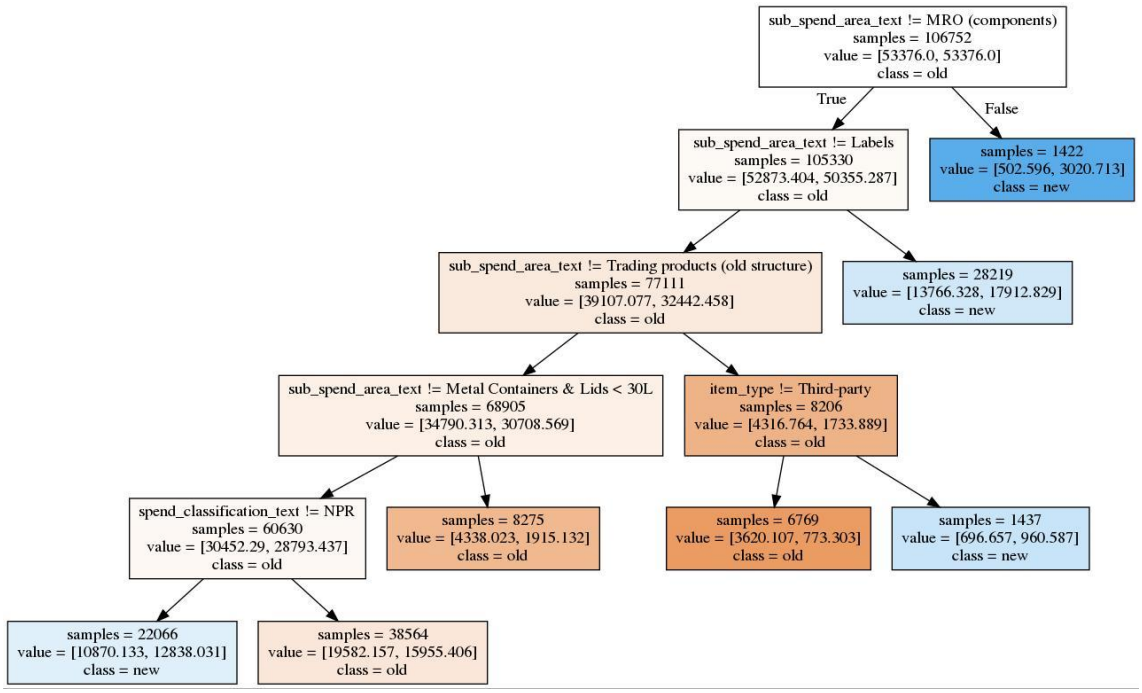

Figure 15: Decision tree for the top-5 sequences

According to the induced decision tree, the change between the two classes was introduced by the following main reasons (see Table 10).

Ranking	Feature importance	Case attribute	Attribute value	Previous percentage	New percentage
1	0.326435	Sub spend area text	MRO (components)	0.9%	5.7%
2	0.193304	Sub spend area text	Labels	25.8%	33.6%
3	0.137531	Item type	Third-party	2.4%	3.8%
4	0.129116	Spend area text	Trading & End Products	8.1%	3.2%
5	0.128192	Sub spend area text	Metal Containers & Lids < 30L	8.1%	3.6%

Table 10: Feature importance in the classifier for top-5 sequences

From this analysis, we conclude that sub *spend area text* being *MRO (components)* is the main reason for the new behavior, since it has a feature importance of 0.326435. In practical terms, this means that *MRO (components)* items can be handled differently in the process (*Record Invoice Receipt* can occur before *Record Goods Receipt*, i.e. in the parallel branching discussed in the flow analysis).

The old sequence is now comprised by the previous five sequences (106 752 cases). The new flow, sixth sequence, has 8 673 cases and is the following:

Create Purchase Requisition Item → Create Purchase Order Item → Vendor creates invoice → Record Goods Receipt → Record Invoice Receipt → Clear Invoice

The decision tree that was obtained from the classification of case IDs into these two classes (i.e. sequences) is presented in Figure 16.

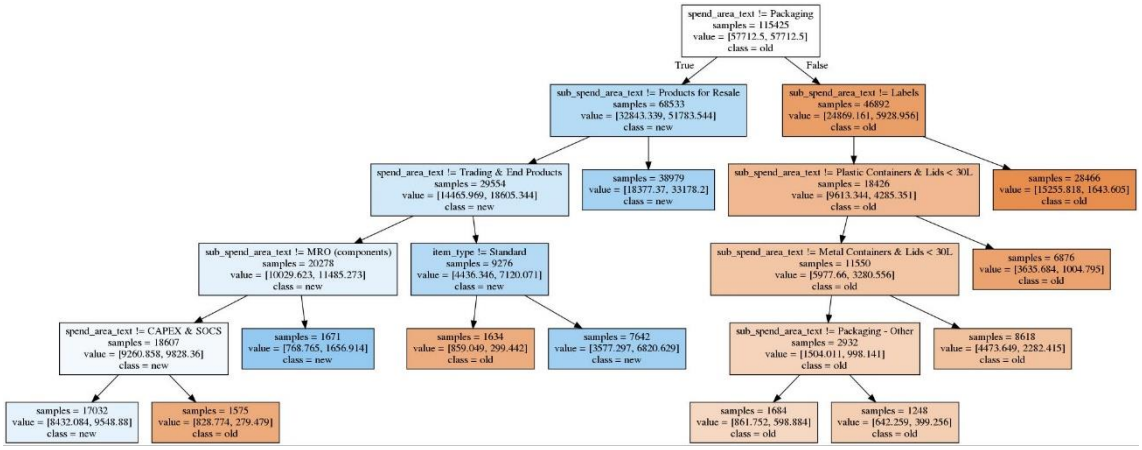

Figure 16: Decision tree for the top-6 sequences

According to the induced decision tree, the change between the two classes was introduced by the following main reasons (see Table 11).

Ranking	Feature importance	Case attribute	Attribute value	Previous percentage	New percentage
1	0.816024	Spend area text	Packaging	43.1%	10.3%
2	0.069807	Sub spend area text	Labels	26.4%	2.8%
3	0.033836	Item type	Standard	97.3%	96.9%
4	0.027132	Sub spend area text	Products for Resale	31.8%	57.5%
5	0.016680	Spend area text	CAPEX & SOCS	2.8%	3.4%

Table 11: Feature importance in the classifier for top-6 sequences

From this analysis, we conclude that *spend area text* being *Packaging* is the main reason for the new behavior, since it has a feature importance of 0.816024. In practical terms, this means that *Packaging* items can be handled differently in the process (*Create Purchase Requisition Item* can occur before *Create Purchase Order Item*).

The old sequence is now comprised by the previous six sequences (115 425 cases). The new flow, seventh sequence, has 7 694 cases and is the following:

Create Purchase Order Item → Record Goods Receipt → Vendor creates invoice → Record Invoice Receipt → Remove Payment Block → Clear Invoice

The decision tree that was obtained from the classification of case IDs into these two classes (i.e. sequences) is presented in Figure 17.

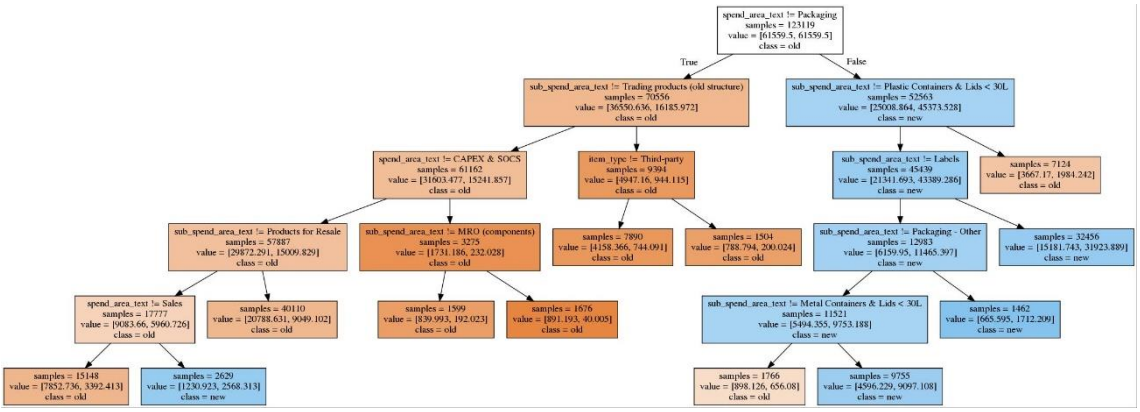

Figure 17: Decision tree for the top-7 sequences

According to the induced decision tree, the change between the two classes was introduced by the following main reasons (see Table 12).

Ranking	Feature importance	Case attribute	Attribute value	Previous percentage	New percentage
1	0.716421	Spend area text	Packaging	40.6%	73.7%
2	0.110318	Sub spend area text	Plastic Containers & Lids < 30L	6.0%	3.2%
3	0.082894	Spend area text	Sales	35.8%	18.9%
4	0.029719	Spend area text	Trading & End Products	8.0%	1.5%
5	0.018322	Spend area text	CAPEX & SOCS	2.8%	0.4%

Table 12: Feature importance in the classifier for top-7 sequences

From this analysis, we conclude that *spend area text* being *Packaging* is the main reason for the new behavior, since it has a feature importance of 0.716421. In practical terms, this means that *Packaging* items can be handled differently in the process (Record Goods Receipt can occur before Vendor creates invoice and *Remove Payment Block* can occur before *Clear Invoice*).

The old sequence is now comprised by the previous seven sequences (123 119 cases). The new flow, eighth sequence, has 4 807 cases and is the following:

Create Purchase Order Item → Delete Purchase Order Item

The decision tree that was obtained from the classification of case IDs into these two classes (i.e. sequences) is presented in Figure 18.

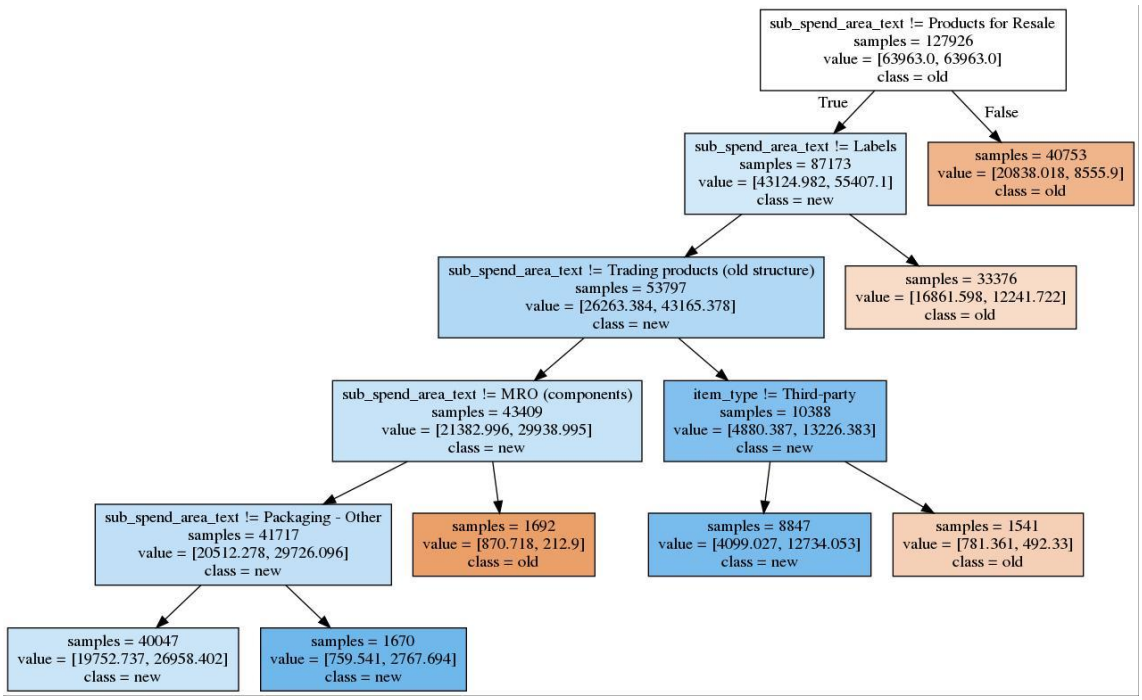

Figure 18: Decision tree for the top-8 sequences

According to the induced decision tree, the change between the two classes was introduced by the following main reasons (see Table 13).

Ranking	Feature importance	Case attribute	Attribute value	Previous percentage	New percentage
1	0.511958	Sub spend area text	Products for Resale	32.6%	13.4%
2	0.254870	Sub spend area text	Labels	26.4%	19.1%
3	0.089025	Sub spend area text	Trading products (old structure)	7.6%	20.7%
4	0.050925	Sub spend area text	MRO (components)	1.4%	0.3%
5	0.049808	Item type	Third-party	2.4%	2.1%

Table 13: Feature importance in the classifier for top-8 sequences

From this analysis, we conclude that sub *spend area text* being *Products for Resale* is the main reason for the new behavior, it has a feature importance of 0.511958. In practical terms, this means that *Products for Resale* items can be handled differently in the process (*Delete Purchase Order Item* can occur after *Create Purchase Order Item*).

As there was not any significant change to the overall subprocess flow after considering eight sequences, we decided not to go further with this analysis.

In conclusion, our analysis approach to the BPI Challenge 2019 allowed us to extract the control-flow model (using a transition counting technique) as well as to understand the reason behind the differences in each new sequence flow (using decision trees). Our analysis supports our finding that a BPI Challenge

is best analyzed combining both process mining and data mining. After understanding the flow and systematizing our findings, we were able to devise a BPMN model for the *3-way matching, invoice before goods receipt* subprocess.

The results of the data mining analysis (here using decision trees) could be a point for further investigation to the subprocess flow as it shows why the behavior of the subprocess differs from case to case. For example, *Packaging* items were the main reason why the process flow differed, so we could perform a control-flow analysis for *Packaging* and *non- Packaging* items.

Chapter 5
Conclusion

A lot of insight can be acquired from the analysis of past BPI Challenges, in terms of several possible techniques that can be used, in different perspectives as well within the same perspective.

We have noticed that real-world event logs often include not only event information, but also various attributes associated with each process instance. One of the most important conclusions that can be drawn is that the analysis of these event logs, containing both event sequences and case attributes, requires an analysis that goes beyond the application of classical process mining techniques.

In addition to the discovery of the process flow, it is possible to try to understand this flow through an analysis of the case attributes. The analysis of these case attributes can be done by resorting to data mining techniques. One of the most used classifiers in this context – and we saw this in our survey of BPI challenges – is decision trees.

As a result of this work, we proposed an analysis approach for real-world event logs that is based on the discovery of the control flow complemented by an analysis of all the decision points in that flow. This approach was described in Chapter 3 and then applied in Chapter 4 to the BPI challenge 2019.

5.1. Main contributions

With this work, we have achieved the following main results:

- We have done a review of all BPI challenges and all submissions in each BPI challenge.
- We have done a comparative analysis, in which we collected the techniques and tools used in each BPI challenge, as well as those that were most commonly used across all BPI challenges.
- We have proposed an approach for the discovery of control flow complemented by a data mining component based on decision trees that allows us to study the deviations and decision points in this flow.

- We have presented the application of this approach on a case study based on the recently published BPI challenge 2019.

Analyzing an event log can become a more multi-disciplinary task than it appeared at first. Given the specific data, it may be necessary to use different techniques to answer to the challenge, in particular the business questions being presented.

5.2. Future work

There are several opportunities for future work. For example, the proposed approach can be expanded in order to include other data mining techniques to perform the same study regarding decision points. Some of the options include Random Forests and other classifiers based on trees whose implementation is widely available.

Another possibility is to use unsupervised learning for studying the process in clusters or groups of similar instances. For example, the BPI challenge 2019 asked for not one but several process models according to *item category*. We could use for example clustering to group the instances and study each group separately, as it has been done in previous work [92].

References

[1] W. van der Aalst, *Process mining: Data science in action*. Springer, 2016.

[2] R. P. J. C. Bose and W. M. P. van der Aalst, “Analysis of Patient Treatment Procedures: The BPI Challenge Case Study,” in *BPI Challenge*, 2011.

[3] J. Li, R. P. J. C. Bose, and W. M. P. van der Aalst, “Mining context-dependent and interactive business process maps using execution patterns,” in *Lecture Notes in Business Information Processing 66*, 2011.

[4] R. P. Jagadeesh Chandra Bose and W. van der Aalst, “Trace Alignment in Process Mining: Opportunities for Process Diagnostics,” in *Business Process Management. BPM 2010. Lecture Notes in Computer Science, vol 6336.*, 2010, vol. 6336 LNCS, pp. 227–242.

[5] F. Caron, J. Vanthienen, J. De Weerdt, and B. Baesens, “Beyond X-Raying a Care-Flow: Adopting Different Focuses on Care-Flow Mining,” in *BPI Challenge*, 2011.

[6] A. J. M. M. Weijters, W. M. P. van der Aalst, and A. K. Alves De Medeiros, *Process mining with the HeuristicsMiner algorithm*. Eindhoven: BETA publicatie : working papers, vol. 166, Technische Universiteit Eindhoven, 2006.

[7] W. M. P. van der Aalst and M. Song, “Mining Social Networks: Uncovering Interaction Patterns in Business Processes,” Springer, Berlin, Heidelberg, 2004, pp. 244–260.

[8] G. Varvaressos, “Semantic Process Mining,” in *BPI Challenge*, 2011.

[9] A. D. Bautista, L. Wangikar, and S. M. K. Akbar, “Process Mining-Driven Optimization of a Consumer Loan Approvals Process,” in *BPI Challenge*, 2012.

[10] R. P. J. C. Bose and W. M. P. van der Aalst, “Process mining applied to the BPI challenge 2012: Divide and conquer while discerning resources,” in *Lecture Notes in Business Information Processing*, 2013.

[11] W. M. P. van der Aalst, H. T. de Beer, and B. F. van Dongen, “Process Mining and Verification of Properties: An Approach Based on Temporal Logic,” in *On the Move to Meaningful Internet Systems*

2005: CoopIS, DOA, and ODBASE, Springer, 2005, pp. 130–147.

[12] A. Adriansyah and J. C. A. M. Buijs, “Mining Process Performance from Event Logs The BPI Challenge 2012 Case Study,” in *BPI Challenge*, 2012.

[13] W. van der Aalst, T. Weijters, and L. Maruster, “Workflow mining: discovering process models from event logs,” *IEEE Trans. Knowl. Data Eng.*, vol. 16, pp. 1128–1142, Sep. 2004.

[14] J. M. E. M. van der Werf, B. F. van Dongen, C. A. J. Hurkens, and A. Serebrenik, “Process Discovery using Integer Linear Programming,” in *Applications and Theory of Petri Nets*, 2008.

[15] J. C. A. M. Buijs, B. F. van Dongen, and W. M. P. van der Aalst, “On the Role of Fitness, Precision, Generalization and Simplicity in Process Discovery,” in *On the Move to Meaningful Internet Systems (OTM 2012)*, Springer, Berlin, Heidelberg, 2012, pp. 305–322.

[16] T. Molka, W. Gilani, and X.-J. Zeng, “Dotted Chart and Control-Flow Analysis for a Loan Application Process,” in *BPI Challenge*, 2012.

[17] R. S. Mans, M. H. Schonenberg, M. Song, W. M. P. van der Aalst, and P. J. M. Bakker, “Application of Process Mining in Healthcare – A Case Study in a Dutch Hospital,” in *Biomedical Engineering Systems and Technologies*, Springer, 2008, pp. 425–438.

[18] H. M. W. Verbeek, “BPI Challenge 2012: The Transition System Case,” in *BPI Challenge*, 2012.

[19] E. Verbeek, J. Buijs, B. F. van Dongen, and W. M. P. van der Aalst, “ProM 6: The Process Mining Toolkit,” in *Business Process Management Demonstration Track*, 2010.

[20] W. M. P. van der Aalst, V. Rubin, B. F. van Dongen, E. Kindler, and C. W. Günther, “Process Mining: A Two-Step Approach using Transition Systems and Regions,” *BPM Cent. Rep. BPM-06-30*, 2006.

[21] C. J. Kang, K. Shin, E. S. Lee, J. H. Kim, and M. A. An, “Analyzing Application Process for a Personal Loan or Overdraft of Dutch Financial Institute with Process Mining Techniques,” in *BPI Challenge*, 2012.

[22] C. J. Kang *et al.*, “Process Mining-based Understanding and Analysis of Volvo IT’s Incident and Problem Management Processes,” in *BPI Challenge*, 2013.

[23] A. Bautista, S. Akbar, A. Alvarez, T. Metzger, and M. Reaves, “Process Mining in Information Technology Incident Management: A Case Study at Volvo Belgium,” in *BPI Challenge*, 2013.

[24] E. Dudok and P. van den Brand, “Mining an Incident Management Process,” in *BPI Challenge*, 2013.

[25] M. Arias and E. Rojas, “Volvo Incident and Problem Management Behavior Analysis,” in *BPI Challenge*, 2013.

[26] S. van den Broucke, J. Vanthienen, and B. Baesens, “Volvo IT Belgium VINST,” in *BPI Challenge*, 2013.

[27] F. van Geffen and R. Niks, “Accelerate DMAIC using Process Mining,” in *BPI Challenge*, 2013.

[28] J. Hansen, “Analyzing Volvo IT Belgium’s Incident and Problem Management Data Using Automated Business Process Discovery,” in *BPI Challenge*, 2013.

[29] J. Hevia and C. Saint-Pierre, “Analyzing Volvo Information with Process Mining,” in *BPI Challenge*, 2013.

[30] C. W. Günther and W. M. P. van der Aalst, “Fuzzy mining-adaptive process simplification based on multi-perspective metrics,” *Lect. Notes Comput. Sci.*, vol. 4714, 2007.

[31] J. Martens, “Professional Use of Process Mining for Analyzing Business Processes,” in *BPI Challenge*, 2013.

[32] Z. Paszkiewicz and W. Picard, “Analysis of the Volvo IT Incident and Problem Handling Processes using Process Mining and Social Network Analysis,” in *BPI Challenge*, 2013.

[33] A. Jalali, “Supporting Social Network Analysis Using Chord Diagram in Process Mining,” in *BIR 2016: Perspectives in Business Informatics Research*, Springer, 2016, pp. 16–32.

[34] S. Radhakrishnan and G. Anantha, “Process Improvement Focused Analysis of VINST IT Support Logs,” in *BPI Challenge*, 2013.

[35] P. van den Spiegel, L. Dieltjens, and L. Blevi, “Applied Process Mining Techniques for Incident and Problem Management,” in *BPI Challenge*, 2013.

[36] S. Taylor, *The Official Introduction to the ITIL Service Lifecycle*. London: TSO, 2007.

[37] P. Buhler *et al.*, “Service Desk and Incident Impact Patterns Following ITIL Change Implementation,” in *BPI Challenge*, 2014.

[38] J. Suchy and M. Suchy, “Predictive Model for supporting ITIL Business Change Management Processes,” in *BPI Challenge*, 2014.

[39] M. Dees and F. van den End, “A Predictive Model for the Impact of Changes on the Workload of Rabobank Group ICT’s Service Desk and IT Operations,” in *BPI Challenge*, 2014.

[40] S. Buffett, B. Emond, and C. Goutte, “Using Sequence Classification to Label Behavior from Sequential Event Logs,” in *BPI Challenge*, 2014.

[41] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules,” in *20th International Conference on Very Large Data Bases (VLDB ’94)*, 1994.

[42] P. van den Spiegel, L. Dieltjens, L. Blevi, J. Verdickt, P. Albertini, and T. Provinciael, “Applied data mining and process mining techniques for analyzing the impact of a change on the workload of the Service Desk and IT Operations,” in *BPI Challenge*, 2014.

[43] J. Hansen, “Analysing Rabobank Group ICT,” in *BPI Challenge*, 2014.

[44] T. Thaler, S. Knoch, N. Krivograd, P. Fettke, and P. Loos, “ITIL Process and Impact Analysis at

Rabobank ICT," in *BPI Challenge*, 2014.

[45] M. Arias, M. Arriagada, E. Rojas, C. Saint-Pierre, and M. Sepúlveda, "Rabobank: Incident and change process analysis," in *BPI Challenge*, 2014.

[46] G. Cacciola, R. Conforti, and H. Nguyen, "Rabobank: A Process Mining Case Study BPI Challenge 2014 Report," in *BPI Challenge*, 2014.

[47] B. C. P. Brandão, G. N. Lopes, and P. H. P. Richetti, "BPIC 2014: Insights from the Analysis of Rabobank Service Desk Processes," in *BPI Challenge*, 2014.

[48] M. Westergaard, C. Stahl, and H. A. Reijers, "UnconstrainedMiner : efficient discovery of generalized declarative process models," 2013.

[49] S. W. Hong, J. Y. Hwang, D. B. Kim, H. S. Choi, S. J. Choi, and S. H. Hong, "Linking and Analyzing Rabobank Group ICT's ITIL Subprocesses based on the Synergetic Capabilities of Process Mining and Data Mining," in *BPI Challenge*, 2014.

[50] T. H. C. van der Heijden, "Process Mining Project Methodology: Developing a General Approach to Apply Process Mining in Practice (Msc Thesis)," Eindhoven University of Technology, 2012.

[51] U. van der Ham, "Benchmarking of Five Dutch Municipalities with Process Mining Techniques Reveals Opportunities for Improvement," in *BPI Challenge*, 2015.

[52] I. Teinemaa, A. Leontjeva, and K.-O. Masing, "BPIC 2015: Diagnostics of Building Permit Application Process in Dutch Municipalities," in *BPI Challenge*, 2015.

[53] P. van den Spiegel and L. Blevi, "Discovery and analysis of the Dutch permitting process BPI Challenge 2015," in *BPI Challenge*, 2015.

[54] S. Buffett and B. Emond, "Using Sequential Pattern Mining and Social Network Analysis to Identify Similarities, Differences and Evolving Behaviour in Event Logs," in *BPI Challenge*, 2015.

[55] P. M. Dixit, B. F. A. Hompes, N. Tax, and S. J. van Zelst, "Handling of Building Permit Applications in The Netherlands: A Multi-Dimensional Analysis," in *BPI Challenge*, 2015.

[56] N. Martin, G. Janssenswillen, T. Jouck, M. Swennen, M. Hosseinpour, and F. Masoumigoudarzi, "An Exploration and Analysis of The Building Permit Application Process in Five Dutch Municipalities," in *BPI Challenge*, 2015.

[57] J. Martens and P. Verheul, "Social Performance Review of 5 Dutch Municipalities: Future Fit Cases for Outsourcing?," in *BPI Challenge*, 2015.

[58] J. Suchy and M. Suchy, "Process Mining techniques in complex Administrative Processes," in *BPI Challenge*, 2015.

[59] H. S. Choi *et al.*, "Process Mining of Five Dutch Municipalities' Building Permit Application Process: The Value Added in E-Government," in *BPI Challenge*, 2015.

[60] S. Dadashnia, J. Evermann, P. Fettke, P. Hake, N. Mehdiyev, and T. Niesen, “Identification of Distinct Usage Patterns and Prediction of Customer Behavior,” in *BPI Challenge*, 2016.

[61] U. van der Ham, “Marking up the right tree: understanding the customer process at UWV,” in *BPI Challenge*, 2016.

[62] A. Jalali, “Exploring different aspects of users behaviours in the Dutch autonomous administrative authority through process cubes,” in *BPI Challenge*, 2016.

[63] G. Janssenswillen, M. Creemers, T. Jouck, N. Martin, and M. Swennen, “Analyzing customer interactions at the Dutch Employee Insurance Agency,” in *BPI Challenge*, 2016.

[64] F. Heidari and N. Assy, “BPI Challenge 2016: Usage Analytics using Process Mining - Key Finding for the Dutch Employee Insurance Agency,” in *BPI Challenge*, 2016.

[65] A. M. B. Rodrigues *et al.*, “Stairway to value: mining a loan application process,” in *BPI Challenge*, 2017.

[66] D. Ryu, J. Lim, D. Jeong, and M. Song, “Analysis of Loan Process Using Various Process Mining Techniques The BPIC 2017,” in *BPI Challenge*, 2017.

[67] S. Dadashnia *et al.*, “Exploring the Potentials of Artificial Intelligence Techniques for Business Process Analysis,” in *BPI Challenge*, 2017.

[68] L. Blevi, J. Robbrecht, and L. Delporte, “Process mining on the loan application process of a Dutch Financial Institute,” in *BPI Challenge*, 2017.

[69] F. Berger, “Mining Event Log Data to Improve a Loan Application Process,” in *BPI Challenge*, 2017.

[70] G. Scheithauer, R. Henne, A. Kerciku, R. Waldenmaier, and U. Riedel, “Suggestions for improving a bank’s loan application process based on a process mining analysis,” in *BPI Challenge*, 2017.

[71] C. Day and S. Smith, “Insights into the Consumer Loan Application process of a Dutch bank,” in *BPI Challenge*, 2017.

[72] L. Wangikar, A. Chandra, D. Yadav, N. Biswas, R. Kumar, and S. Alam, “Driving Higher Operations Effectiveness in Banking: Process Mining and Data Science,” in *BPI Challenge*, 2017.

[73] U. van der Ham, “Improving speed and effectiveness of loan processes in a bank,” in *BPI Challenge*, 2017.

[74] E. Povalyaeva, I. Khamitov, and A. Fomenko, “BPIC 2017: Density Analysis of the Interaction With Client,” in *BPI Challenge*, 2017.

[75] P. Badakhshan, J. C. Maciel, and B. Wurm, “Process Mining in the Financial Industry: A Case Study,” in *BPI Challenge*, 2017.

[76] D. Jeong, J. Lim, and Y. Bae, “BPIC 2017: Business process mining - A Loan process application,” in *BPI Challenge*, 2017.

[77] A. Bolt, “Process Variability Analysis over Offers: Variant Detection and Suggestions.,” in *BPI Challenge*, 2017.

[78] E. Varas and C. Á. Iglesias, “Process Mining of a Loan Approval Process in a Financial Institution,” in *BPI Challenge*, 2017.

[79] K. Park, H. Lee, and H. Jung, “Analyzing Loan Process with Pocess Mining Techniques,” in *BPI Challenge*, 2017.

[80] M. F. Sani and H. Sotudeh, “Business Understanding Using Process Mining,” in *BPI Challenge*, 2017.

[81] C. H. Oh, M. G. Choi, and M. Zahedy, “Seventh International Business Process Intelligence Challenge (BPIC’17),” in *BPI Challenge*, 2017.

[82] M. T. Sant’Anna and J. A. Leite, “Using Process Mining Techniques to Support Improvement in a Financial Institution Process,” in *BPI Challenge*, 2017.

[83] H. Jang, Y. Jeong, and W. Kim, “Process Mining Applied to the BPI Challenge 2017: Analyze Netherland bank case using Disco and ProM,” in *BPI Challenge*, 2017.

[84] M. Kwon, M. Lee, and H. Kim, “Application of Process Mining in a Consumer Loan Company Process,” in *BPI Challenge*, 2017.

[85] A. Carvallo *et al.*, “Applying Process Mining for Loan Approvals in a Banking,” in *BPI Challenge*, 2017.

[86] F. Rojos *et al.*, “Process Mining: Research in Banking Operations,” in *BPI Challenge*, 2017.

[87] R. Carmona *et al.*, “Analysis of Loan Application Process Using Process Mining,” in *BPI Challenge*, 2017.

[88] S. Pauwels and T. Calders, “Detecting and Explaining Drifts in Yearly Grant Applications,” in *BPI Challenge*, 2018.

[89] L. Wangikar, S. Dhuwalia, A. Yadav, B. Dikshit, and D. Yadav, “Faster Payments to Farmers: Analysis of the Direct Payments Process of EU’s Agricultural Guarantee Fund,” in *BPI Challenge*, 2018.

[90] J. Brils, N. van den Elsen, J. de Priester, and T. Slooff, “Analysis and Prediction of Undesired Outcomes,” in *BPI Challenge*, 2018.

[91] D. R. Ferreira, *A Primer on Process Mining – Practical Skills with Python and Graphviz*. Springer, 2017.

[92] G. M. Veiga and D. R. Ferreira, “Understanding Spaghetti Models with Sequence Clustering for ProM,” in *5th International Workshop on Business Process Intelligence (BPI 2009), in Business Process Management Workshops, LNBIP 43*, 2010, pp. 92–103.

Appendix

In this appendix, we provide an analysis of additional process mining perspectives beyond the control-flow perspective regarding the BPI challenge 2019 event log.

Organizational perspective

Although the BPI challenge 2019 had no business question addressing the organizational perspective, we decided to perform it to show the potential of this analysis.

There are 628 different resources in the event log. There 607 human resources, 20 automatic resources, and a special user named NONE; meaning the resource was not recorded in the event log.

	2-way matching	Consignment	3-way matching
Human	18 (87.76%)	148 (98.11%)	601 (64.34%)
Machine	1 (0.37%)	7 (1.89%)	20 (10.02%)
NONE	1 (11.87%)	-	1 (25.64%)

Table 14: Number of distinct resources working in each process

Interestingly, only 6 resources out of all resources do not work in 3-way matching process, they are: user_426, user_602, user_603, user_604, user_605, and user_606.

The low number of resources in *2-way matching* is expectable as this process covers only ~1% of all cases and ~1% of all events. There is little resource allocation to this process, probably meaning that the company allocates the minimum possible resources to handle the 2-way matching process.

In contrast, in the *consignment* process there are a lot of resources when compared to *2-way matching*. This process is responsible for ~5% of all cases and 2% of all events. Although the percentage of cases

is five times larger (*consignment vs 2-way matching*), the percentage of events is very similar; only 1% increase. Unless the consignment process is more complex than the *2-way matching*, probably a reduction in the number of resources allocated to this process might be a good idea. These resources could be moved to the overwhelming *3-way matching process*.

The high number of resources in *3-way matching* process is acceptable as this process covers 93% of all cases and 97% of all events in the event log. Those numbers mean that almost everyone works in the 3-way matching process.

In order to understand the interactions and collaboration between resources, we performed a social network analysis based on the handover of work and working together techniques. We have seen in Table 2 that Disco[1] and ProM[2] are the most used tools in the BPI challenges, so we decided to use only ProM to perform the social network analysis.

ProM offers several plug-ins implementing the social network analysis, among them are the ones we use. Disco, in contrast offers limited support for social network analysis, as it can only be used to analyze the handover of work with a few tweaks/fine tunings in its control-flow discovery algorithm.

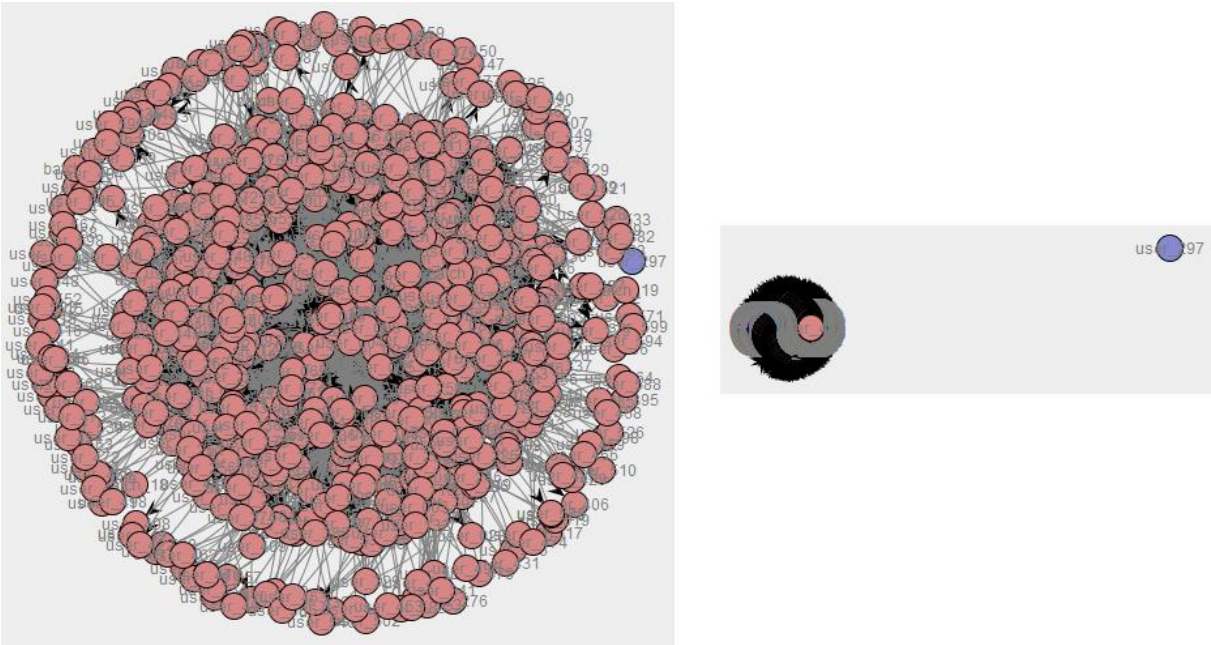

Figure 19: Graph (left) and clusters (right) for handover of work

Figure 19 (left part) shows that all resources handover work to any other resource. The inner resources handover more work than the outer resources. This type of graph layout points out that the process is performed in a flexible or ad-hoc manner; more precisely the next person to handle a task within a process instance is performed by whoever, person or machine, is available. The right part shows that there are two clusters of users in the organization, *user_297* and all the other resources. This is an odd cluster distribution and it might be interesting for the organization to perform extra analysis to understand

[1] https://fluxicon.com/disco/
[2] http://promtools.org/

why that resource is in its own cluster. Without a domain expert knowledge, it is extremely hard to understand why that resource is left alone.

After analyzing how directly the work is transferred between resources, we decided to study how groups of users often work together using the social network analysis based on the working together technique.

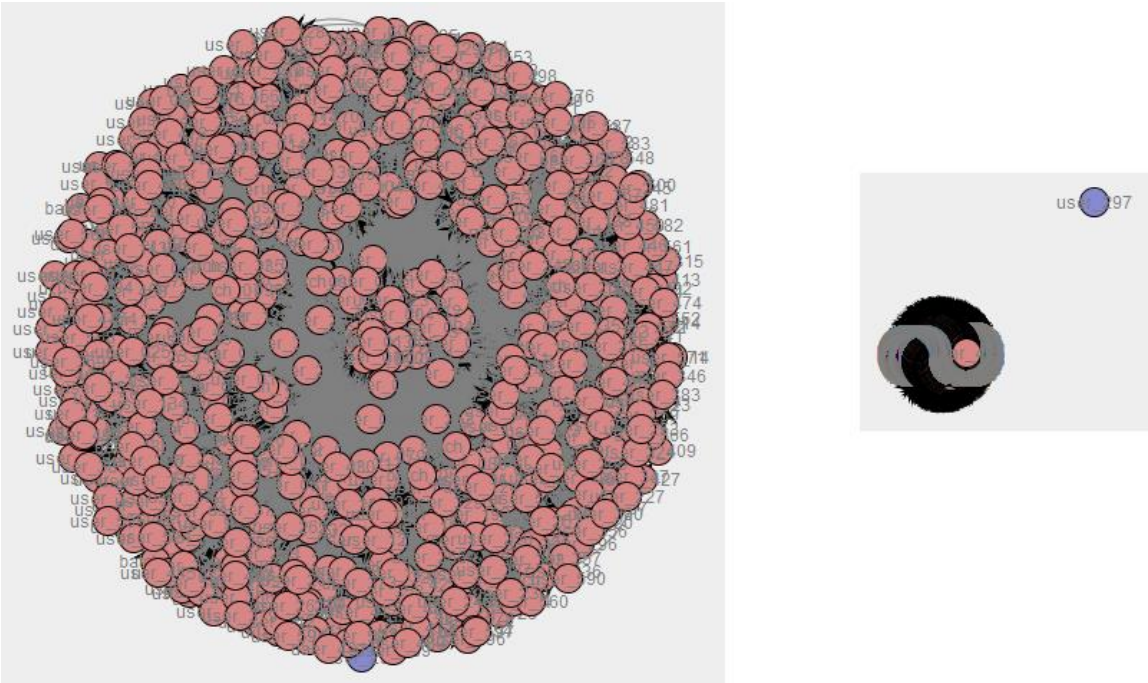

Figure 20: Graph (left) and clusters (right) for working together

In Figure 20 (left part), we can see that all resources work together. This support the result that any resources hands over work to any other resource. The right part shows that there are again two clusters of users in the organization, *user_297* and all the other resources. Again, this should be further investigated.

Performance perspective

Time is the main concern in a performance analysis. The event log provided in the BPI challenge 2019 has the task completion timestamp only, so the type of analysis we can make is limited since we do not have the actual time resources took performing a task. Nonetheless, we can still provide additional insight to the process performance. It should be noted that the lack of start-time in the event log means that all tasks are considered atomic, i.e. tasks have zero duration.

Disco and ProM provide an easy way to obtain global statistics to the overall process, so based on this fact we decided to use one of them. As Disco provides an easier and more friendly interface, we decided to use it for performance analysis.

Figure 21 shows the median time it takes between two tasks. The median time between two tasks is shown along the arc between tasks. For instance, the median time *Vendor creates invoice → Create*

Purchase Order Item is 11.5 days. It should be noted that we do not have the start-time of an event, so that is why Disco indicates “instant” in the task node.

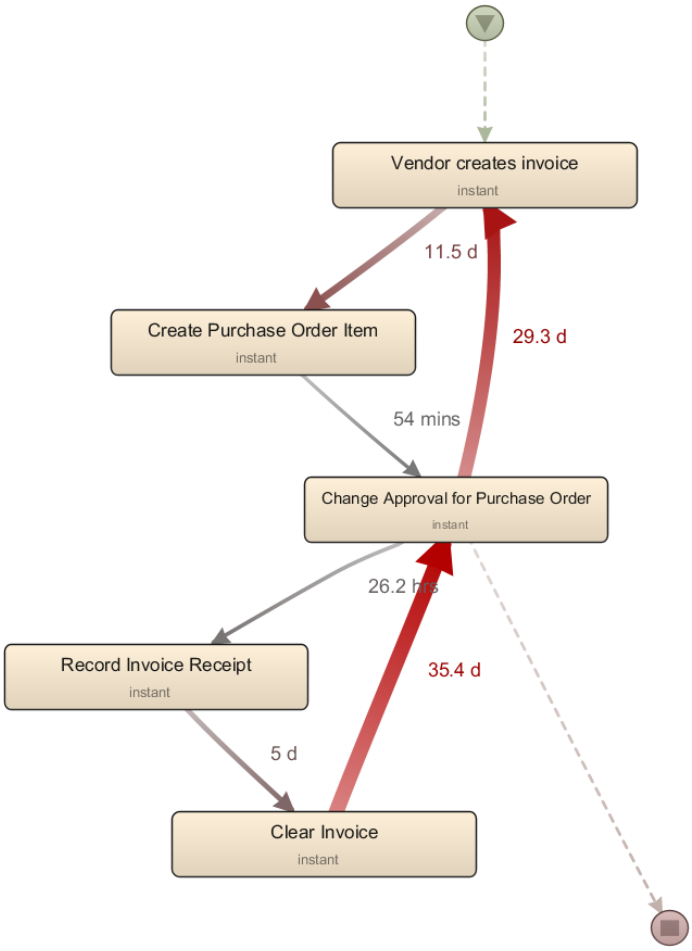

Figure 21: Median time for the 2-way matching process

From *Figure 21* we can conclude that the longest waiting time for 2-way matching process is between *Clear Invoice → Change Approval for Purchase Order*; 35.4 days. This waiting time as well as the one between *Change Approval for Purchase Order → Vendor creates invoice*, 29.3 days, are the main bottlenecks. The time between *Create Purchase Order Item → Change Approval for Purchase Order* is the shortest. The organization might perform additional analysis to understand the reason for those bottlenecks and how can they be reduced. The median case duration is 23.7 days.

In Figure 22 the overall median time for *Consignment* is depicted. We can see that the longest waiting time for Consignment process is between *Create Purchase Order Item → Record Goods Receipt*, 28.6 days. This waiting time is the main bottleneck of the process. The time between *Receive Order Confirmation → Record Goods Receipt* is a minor bottleneck, 16.5 days. A case has a median duration of 19.9 days, so improving those two bottlenecks would improve it substantially.

Finally, the 3-way matching process median time is depicted in Figure 23. The process has two bottlenecks, the longest between *Record Invoice Receipt → Clear Invoice* (65.3 days) and the shortest between *Vendor creates invoice → Create Purchase Requisition Item* (29 days). A case has a median time duration of 66.2 days. So, if the bottlenecks where to be improved, this time would reduce a lot.

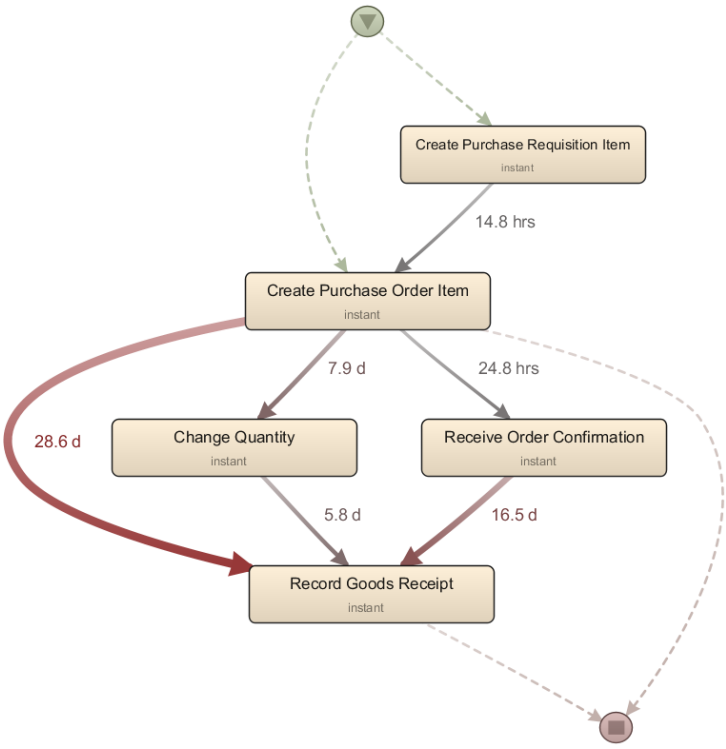

Figure 22: Median time for the consignment process

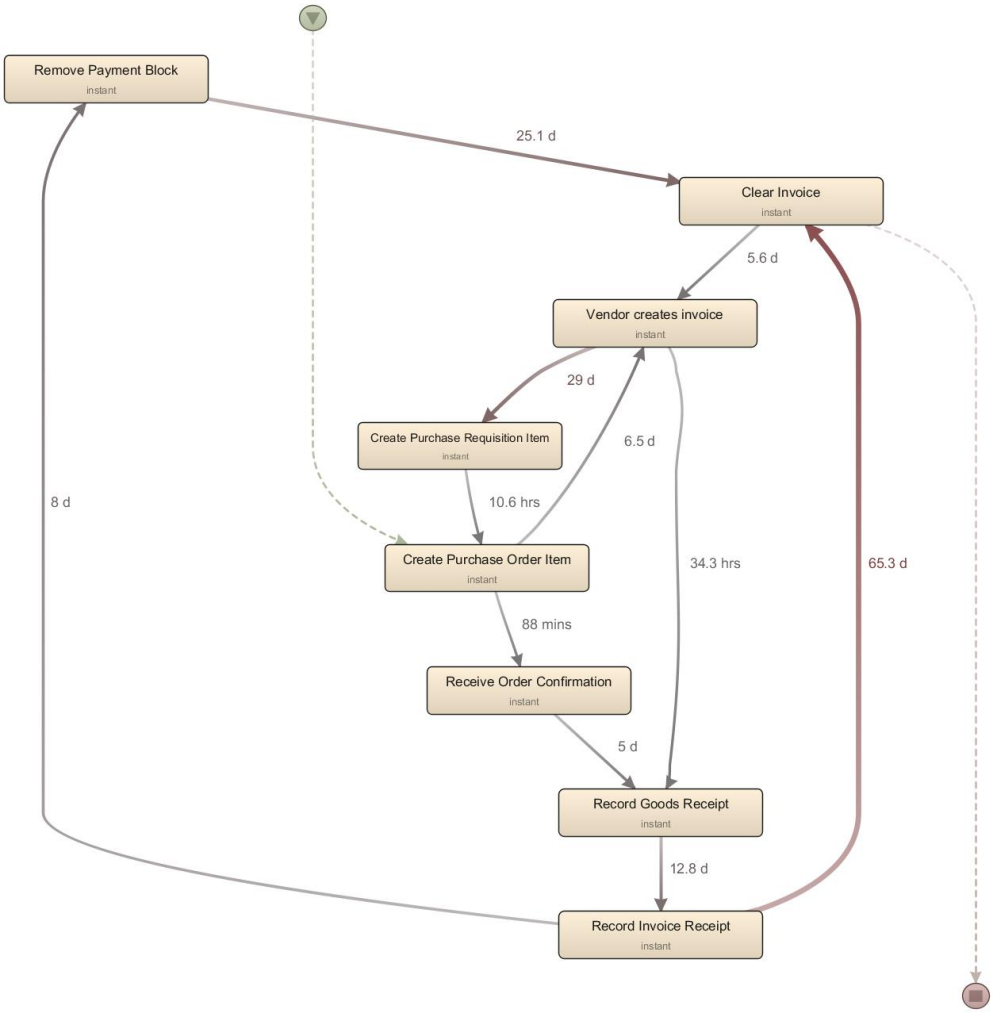

Figure 23: Median time for the 3-way matching process

In conclusion, the performance perspective analysis can be carried out at a control-flow level. However, deeper analysis can only be performed when domain knowledge is added to the process mining analysis. Business process mining is an iterative endeavor, where analysis results are confronted with domain knowledge and a new round of analysis is started. However, often there is no feedback to strengthen the analysis even further in the BPI challenge.