A Case Study on the Business Benefits of Automated Process Discovery

Matej Puchovsky, Claudio Di Ciccio, and Jan Mendling

Vienna University of Economics and Business
matej.puchovsky@gmail.com, {claudio.di.ciccio,jan.mendling}@wu.ac.at

Abstract. Automated process discovery represents the defining capability of process mining. By exploiting transactional data from information systems, it aims to extract valuable process knowledge. Through process mining, an important link between two disciplines – data mining and business process management – has been established. However, while methods of both data mining and process management are well-established in practice, the potential of process mining for evaluation of business operations has only been recently recognised outside academia. Our quantitative analysis of real-life event log data investigates both the performance and social dimensions of a selected core business process of an Austrian IT service company. It shows that organisations can substantially benefit from adopting automated process discovery methods to visualise, understand and evaluate their processes. This is of particular relevance in today's world of data-driven decision making.

1 Introduction

In order to sustain competitive advantage and superior performance in rapidly changing environments, companies have intensified their efforts towards structuring their processes in a better and smarter way. *Business Process Management* (BPM) is considered an effective way of managing complex corporate activities. However, a consistent shift from mere process modelling and simulation towards monitoring of process execution and data exploitation can be observed nowadays [8]. Event logs, i.e., data mapping traces of process execution in modern information systems, can be used to define new process models and derive optimisation possibilities. The techniques of *automated process discovery* aim at extracting process knowledge from event logs and represent the initial steps of exploring capabilities of *process mining*.

Studying real-life enterprise data by applying process mining methods can deliver valuable insights into the actual execution of business operations and their performance. This is of particular relevance in today's age of industries automating their processes via workflow systems [3] with growing support of process execution by various enterprise information systems. By building process definitions, models and exploring execution variations, automated process discovery has the potential to fill the information gap between business process departments and domain experts in enterprise settings. In order to assess the

Pre-print copy of the manuscript published by CEUR-WS.org (available at ceur-ws.org)

business benefits of process mining, we cooperated with an Austrian IT service provider to conduct an industrial case study. The company aimed at analysing one of their core business processes using process execution data from an enterprise resource planning system (ERP). Since our industrial partner has had no previous experience with process mining, the case study focuses on a preliminary assessment of data exploitation through a process discovery initiative.

The remainder of the paper is as follows. In Section 2, we introduce the research area of process mining with its fundamental terminology and related work in the field. Section 3 describes the studied process together with the structure of the examined event log. Besides, we present the approach for our case study. The results of the empirical analysis are covered in Section 4, which is divided into the *process* and *social* views as well as a subsection covering the execution variants. Section 5 elaborates on the business impacts of the research results. Finally, Section 6 summarises our findings and gives suggestions towards future research.

2 Background

Van der Aalst [1] defines process mining as the link between *data* science and *process* science, making it possible to depict and analyse data in dynamic structures. By analysing event logs, i.e., records documenting the process executions tracked by an information system, process mining aims at retrieving process knowledge – usually in a form of process models. Depending on the setting, the extracted models may (i) define and document completely new processes, (ii) serve as a starting point for process improvements, or (iii) be used to align the recorded executions with the predefined process models. The idea to discover processes from the data of workflow management systems was introduced in [7]. Many techniques have been proposed since then: Pure algorithmic ones [6], heuristic [21], fuzzy [11], genetic (e.g., [16]), etc. ProM [2] is currently one of the most used plug-in based software environment incorporating the implementation of many process mining techniques. Process mining is nowadays a well-established field in research and examples of applications in practical scenarios can be found in various environments such as healthcare services [15], [14], financial audits [22] or public sector [5]. The organisational perspective of process mining focuses on investigating the involvement of human resources in processes. Schönig et al. [19] propose a discovery framework postulating background knowledge about the organisational structure of the company, its roles and units as a vital input for analysing the organisational perspective of processes.

3 Case study

The empirical study of the process mining methods was conducted in cooperation with an Austrian IT service provider with a diversified service portfolio. The main aim of the analysis was to evaluate the applicability of process mining for process discovery and to provide solid fundaments for future process improvements. Our industrial partner has a well-developed business process management department responsible for all phases of the business process management lifecycle, as defined by Dumas et al. [10]. All process models and supporting documents are saved in a centralised process management database, which is also used for storing all quality management reports.

During our investigation, we analysed the process of offering a new service (henceforth, offering process), which belongs to the core business operations in the value-chain of our industrial partner. The process describes all the necessary steps that need to be performed in order to create an offering proposal with detailed documents on a specific service, which is a subject of the offer, including pricing information for the customer. Upon approval, the offer is sent to the customer, who decides whether it meets their expectations or not. In case of an acceptance, a new contract is concluded. If the customer rejects the offer, they can either return it for adjustments, when a new version of the offering documents is created, or decline any further modifications. Up until now, almost all information regarding the process has been collected during numerous iterative sessions of interviews with the domain experts. The industrial partner selected this process due to the complex structures of its process model, which has already been simplified, as well as an extensive support through an IT system and suspected loops in process execution that are believed to negatively impact process duration and often cause lack of transparency. The primarily goal is therefore to understand how the process *actually* works, based on the available execution data.

3.1 Approach

Our case study aims to give answers to the following business questions:

- How does the offering process perform based on the available log data? Here, the focus is set primarily on process duration, acceptance rates of the offers, and number of iterations in the recorded cases. Moreover, we investigate what variables might have and impact on the process duration. We thus aim to exploit both the "traditional" log attributes such as *timestamps* or *activities* and "non-standard" attributes with additional process information.
- Are there any iteration loops recorded in the log, and if yes, how many? How many different variations of process execution can be identified and do they vary significantly?
- How many human resources are involved in the process execution and what roles can be identified in the event log records?

The analysis was conducted primarily by means of the process mining tool minit[1]. Throughout the paper, we differentiate between the *process* and *social* view, the former dealing with the performance aspects of the process, the latter explaining its social network. The event log data were processed within both

[1] http://www.minitlabs.com

Table 1: English translations for the German activity labels from the event log

German	English
Erstellung	Offer creation
Fertigstellung	Offer completion
Fachliche Pruefung	Specialist check
Freigabe	Approval
Versand	Shipment
Angebotsentscheidung Kunde	Customer decision
Attribute vervollstaendigen	Amend attributes
AV Person festlegen	Appoint responsible manager
Uebernahme AV	Takeover
SLA-Relevanz beurteilen	Assess SLA relevance
Abstimmung SLA Ersteller	Matching SLA creator
Presales – Bearbeiter dispatchen	Dispatch performer
Presales – Requirement Analyse	requirement analysis
Presales – Solution Design	Solution design
Presales – Ausarbeitung Angebotsinhalte	Drafting offering content
Presales – 4-Augen Pruefung	Confidentiality control

frequency (i.e., absolute numbers of executions) and performance dimensions (i.e., duration metrics). Through a comparison of the process execution variants, i.e., activity sequences shared by more process instances, we aim to identify similarities between cases, but also to uncover execution anomalies recorded in the event log. The analysis of the connections between human resources and their activities in the workflows are used to cover the social perspective of the examined process.

3.2 Description of the event log data

The event log used in this study was created by the IT system owner via a query in the EPR system our industrial partner. The extracted cases were restricted to the workflows started in 2015. Due to the fact that no formal end activity is defined in the process, the log possibly contains several incomplete process instances. We therefore have to resort to specific attributes to determine the status of the analysed process instances. Our industrial partner provided us with a single CSV data table with 12,577 entries (activities), corresponding to 1,265 distinct process instances. Due to privacy concerns of the company, no other platforms were allowed. Therefore, the fact that we were only able to work with one data file was a major limitation for our research.

The recorded events are enriched with the following 11 attributes: *CaseID, Version, Customer, Segment, WF-Start, WF-Status, Activity, Start, End, Duration, Performer*. Overall, 258 employees were involved in the process execution. The activity names were recorded in German, therefore, we have complemented these with English translations (see Table 1) and also employ the labels in English throughout the paper.

The log attribute *workflow status* (WFS) plays an important role for our analysis, even though it is not a typical attribute such as activity name or timestamp. The values recorded under this attribute provide information about the outcome of the offering process or the status of the offer itself. Workflow statuses are automatically recorded at the beginning of every activity and can be changed after the activity is executed. All previous activities are then marked with the latest WFS. This attribute can also be used to determine terminal activities, which possess one of the three workflow statuses describing the outcome of an offering process – *accepted*, *aborted* or *rejected*.

However, a case can be assigned more than one workflow status, signalising an important relationship with another attribute – *version*. A good example of such case is an offer that requires adjustments after being sent to a customer, who rejected the original version. For this purpose, a new version of the offering documents is created, resulting in a new iteration of the workflow – the activities in the first iteration are recorded with *new version*. If the offer is accepted, the activities executed in the second loop are recorded with *offer accepted*. We can conclude that cases with *new version* contain at least one loop and that the version attribute. A closer analysis of both attributes is included in the section dealing with process performance.

4 Results

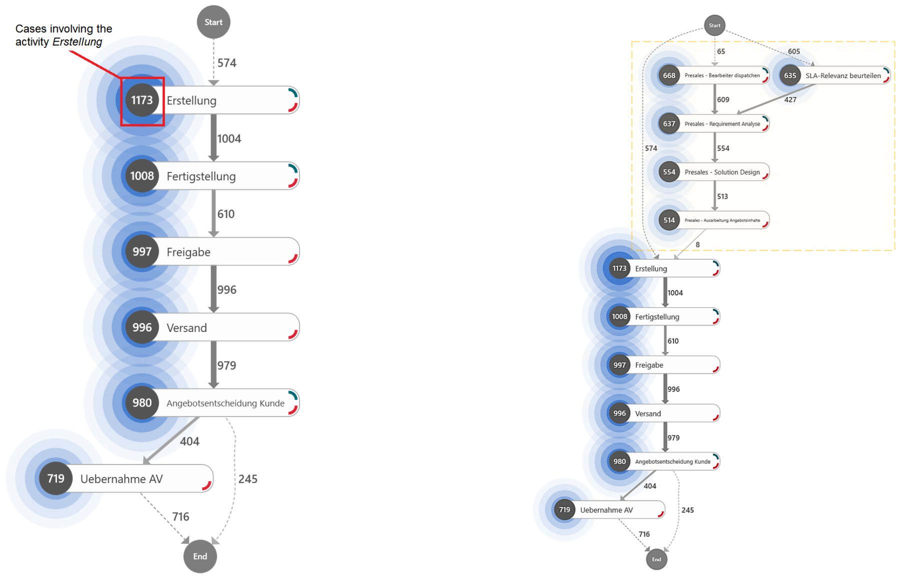
The following section describes the experimental analysis of the event log data.

4.1 Process view

In order to be able to build a process map from the event log data, the *activity names* and *timestamps* are essential.The process map is visualised as a directed graph consisting of n nodes and e edges. Each node represents an event or activity with an individual name and, based on the chosen dimension, the information about the execution *frequency* or *duration*.

Our initial point of interest is the process map depicting the sequence of activities under the frequency dimension, i.e. how often was a particular activity executed or in how many cases can an activity be found. In order to alter the abstraction level of the process map, we use different *complexity levels* of *activities* (nodes) and *paths* (edges/transitions), which determine the number of elements depicted in the model – the higher the complexity level, the more nodes/edges are visualised in the process map.

We first postulate a 0% complexity of both the activities and paths, resulting in a process model with six most frequently executed activities plus the terminal nodes *start* and *end*. In Figure 1a, the ordered activities contain the descriptive metric of *case count*, i.e., the frequency of cases in which the particular activity was performed. For example, it can be observed that the starting activity *Offer creation* was performed in 1,173 process instances of the offering workflow and is therefore accompanied by the strongest highlight around its node. Based on the recorded data, we can conclude that the model in Figure 1a represents the most common path/behaviour of the process. Increasing the percentage of activities and paths visualised in the model leads to growing complexity of the whole



(a) 0% activity, 0% path complexity level

(b) 50% activity, 0% path complexity level

Fig. 1: Process models under frequency dimension with the case frequency metric

graph. This can be observed in Figure 1b, where 50% of all activities available in the event log are depicted. We can observe that additional activity types from the pre-sales phase emerge. Their presence in the model signalises preparing non-standard offers, in which a pre-sales team has to be involved to perform the requirement analysis and deliver a solution design for the desired service. The proportion of the process instances with the pre-sales phase was slightly over 40%.

The highest possible level of activity complexity uncovers a total of 18 logged process activities. At 100% activity and path complexity levels, the generated process map becomes very cluttered (see Figure 2). The graph gives us a good perception of how cross-connected the activities in the process are. The offering process evidently can be performed in numerous variations. Similarly to the starting events, the number of possible activities ending the workflow increases at higher levels of model complexity.

4.2 Variants

A *variant* can be defined as a set of process instances that share a specific sequence of activities. This means that all cases in a certain variant have the same start and end event and that all activities between the terminal nodes are identical and performed in the same chronological order. Variants can be very helpful for recognising different performance behaviour or irregular patterns

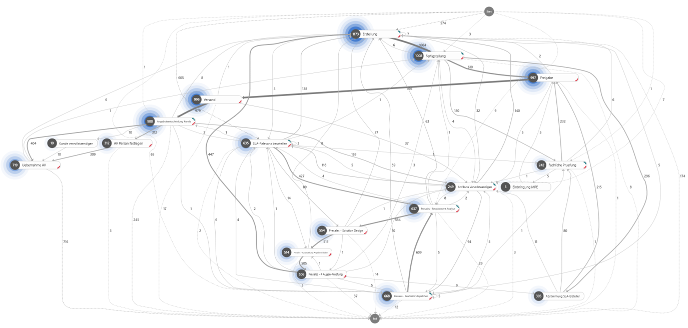


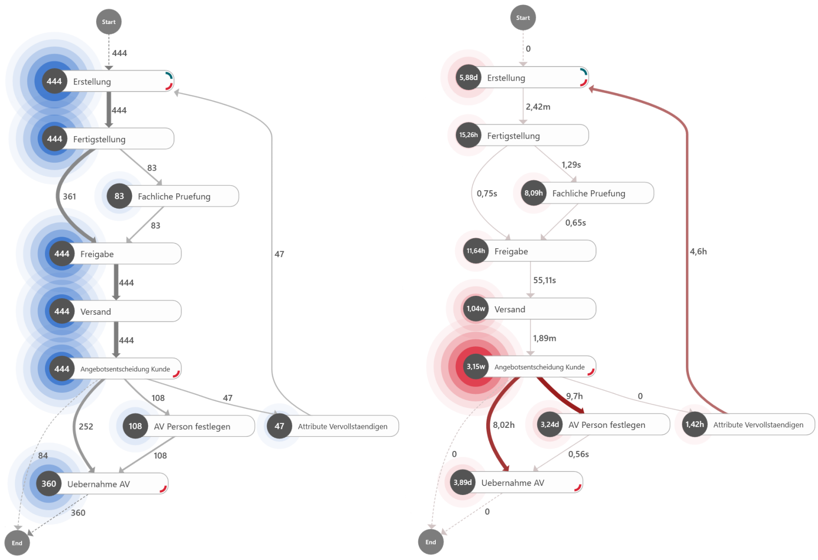
Fig. 2: Process model under frequency dimension at 100% activity and 100% path complexity level

not conforming with the a-priori process model. In general, metrics from both frequency and performance dimensions can be assessed.

The event log of the offering process depicts a total of 280 variants with event count per case varying from 1 to 49. Due to the lack of space, we only concentrate on the top 24 performance variants that cover around 70% of all available cases.

The key goal of our variant comparison is to find similarities between the 24 investigated variants and thus build variant groups explaining the execution of the offering workflow. This allows us to identify two major groups of variants based on whether they depict offers for standard or non-standard services, let them be labelled A and B.

All variants in group A share the initial activity *Offer creation* and end with either *Customer decision* or *Takeover* (see Fig. 3). There are exactly nine variants contained in group A with total case coverage of almost 40% (444 cases). Since no pre-sales activities can be found in the cases of cluster A, we can conclude that all the involved process instances cover standardised offers. The presence of the activity *Amend attributes* signalises that iteration loops were recorded in the cases involved. The second identified variant group, B, features a higher variety of the recorded activities. It contains 15 variants with event frequency varying from 2 to 15. The cases found in this group all contain at least one activity from the pre-sales phase, thus marking process instances dealing with non-standardised services. Interestingly, no loop performance was identified in the variant group B, as the recorded cases were all missing the activity *Amend attributes*, i.e. a trigger for repeated process executions. Fig. 4 shows that, for both variant groups, the mean case duration tends to increase with growing number of activities per case. This can be explained both by the presence of loops or to the resource-intensive pre-sales phase.



(a) Frequency dimension **(b)** Performance dimension

Fig. 3: Process maps of the variant group A under frequency and performance dimension

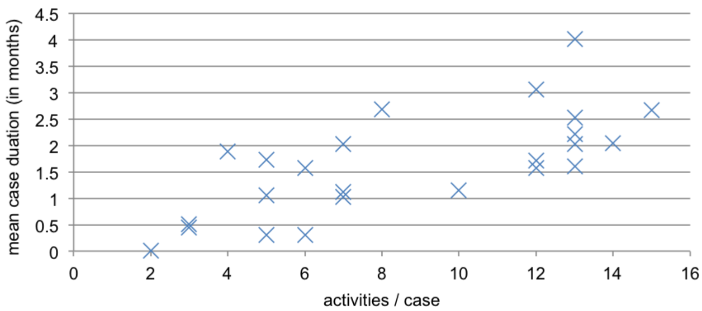


Fig. 4: Relationship between the number of activities per case and mean case duration

4.3 Performance

For the analysed process, we define three main KPIs: (a) duration, (b) number of iterations loops in the process, and (c) process outcome. The duration metrics can be assessed by making use of the start and end timestamps, which are recorded for every activity in the log. By exploiting the data recorded under the version attribute we analyse the number of iterations in the recorded process instances. Finally, the process outcome, i.e., the acceptance rate of the offers created through the process, is derived from the workflow statuses of the recorded cases.

Several activities stand out when it comes to the duration metrics – *Customer decision* accounting the mean duration of over three weeks followed by *Requirement analysis* with significantly lower mean duration of 1.67 weeks and *Offer creation* (6.03 days). These results align with the available a-priori process knowledge as they are known to require the most resources and processing time. The total duration of the analysed cases varies significantly. Therefore, we investigated the impact of three selected variables – number of activities in a case, resources involved and the number of versions a workflow possesses (since versions can be seen as iteration loops) – on the total case duration via a multivariate regression model: $\textit{case duration} = \alpha + \beta_a \cdot \textit{activities} + \beta_r \cdot \textit{resources} + \beta_v \cdot \textit{versions}$.

The coefficients of all *activities* ($\beta_a = 28.79$), *resources* ($\beta_r = 77.883$) and *versions* ($\beta_v = 276.154$) are significantly different from 0. We can therefore conclude that an increase in all selected variables results in an increased case duration with the number of versions having the highest impact. The model variables exhibit a mild degree of correlation ($r = 0.458$). However, the coefficient of determination amounts to 0.21, meaning that only 21% of the case durations can be explained by the selected variables.

The version attribute denotes the version of the offering documents throughout the whole process. The document numbering is entered manually by human resources and begins with V1.0 for new cases. The version number is then automatically assigned to all new activities in the process until the version, and thus, the document, is changed (1.x for minor changes or drafts, x.0 for major alterations). Version numbers can therefore be seen from two major points of view: (i) *business*, where altering versions corresponds to changes in the offering documents, and (ii) *process*, with versions being equivalent to the number of loops in a process instance. By inducing the version number as a metric for process iteration loops, we can conclude that 1,009 (79.76%) out of the recorded 1,265 cases were executed in only one iteration. These are followed by 192 process instances (15.18%) with two iterations. Only 64 (around 5%) cases consisted of three or more loops. These findings indicate a straight-forward execution of the process in the majority of examined cases.

We can also observe a relationship between the *version* attribute and *workflow status*, which is depicted in Figure 5, where the x-axis represents the number of distinct versions found in a case and the y-axis showing the number of workflow statuses per case. Generally speaking, a change in the workflow status triggers a change in the version number. We observe that most process instances are recorded in a 1:1 relationship, where the whole workflow holds only one version number and one workflow status – e.g., an offering proposal that has been accepted after the first iteration with the version number V1.0, or a proposal that has been rejected after the first iteration. The second largest group, 2:2, is common for process instances, where two distinct document versions were recorded (e.g., V1.0 and V2.0) together with two distinct workflow statuses. This behaviour is identified in offerings that were returned for adjustments by

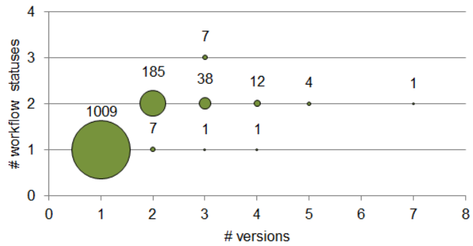


Fig. 5: Case counts for every combination of the number of recorded versions and workflow statuses

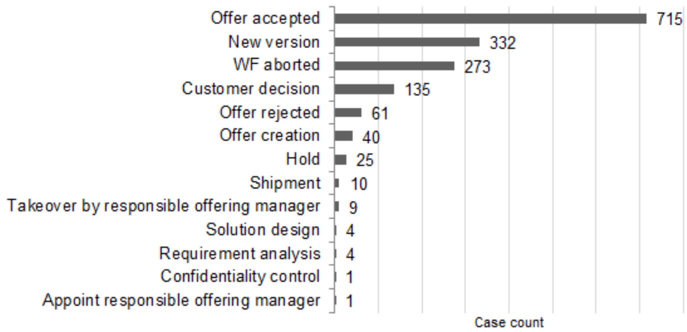


Fig. 6: Case counts for every workflow status recorded in the event log

the customer (i.e., triggering a change in the version number) and were accepted after the second iteration.

Figure 6 depicts a bar chart with all 13 workflow statuses found in the event log with the corresponding case counts. It shows the apparent dominance of *Offer accepted* with 715 cases (56.52% of all recorded). According to the data, only 61 offers were rejected by the customer. The sum of the cases in Figure 6 is 1,610, which is higher than the total case count in the event log (1,265). This difference can be explained by process instances that hold multiple workflow statuses. Yet, some of the recorded statuses do not provide explicit information about the process outcome. For instance, the offering process was *aborted* in 273 cases. Unfortunately, the data do not provide any additional information concerning the reasons for terminating the process.

Combining version numbers with the workflow status also allows for interpreting the "performance efficiency". 70.21% (502) of all accepted offering proposals were concluded after the first iteration with the version number V1.0. However, we also discovered a significant proportion of cases that entered a new loop (242 cases or 20.23%) or were aborted (221 workflows or 28.48%) after the first iteration with the version number V1.0.

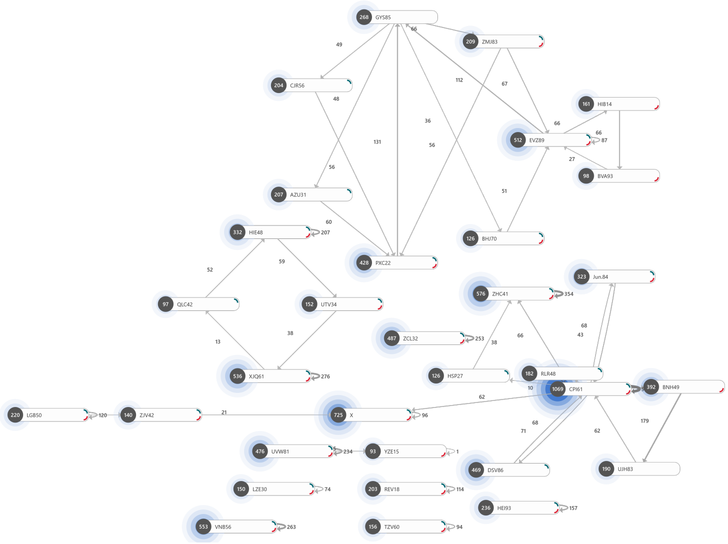


Fig. 7: Social network at 50% resource and 0% connection complexity level

4.4 Social view

Analysing the social graph of the process helps us understand the relationships between different resources involved in the process execution. Our investigation of the social network was primarily focused on examining its frequency aspects. We resort on the investigation of relationships of the performers involved in the process execution at different complexity levels. The initial complexity levels were set to 50% for the resources and 0% for the connections, resulting in the social network consisting of 32 nodes and 48 edges depicted in Figure 7. In [20], Van Steen mentions that the key to understanding the structure of a social network is a deeper analysis of subgroups found *within* the network. Since the investigated social network exhibits very high complexity with 258 resources and 396 connections, this approach is also fitting for our case study. Based on the log data, the top 15 performers (5.81%) account for 60% of all the recorded events. Here, several relationships between specific performers stick out (see Table 2). Noticeable connection counts can also be observed *within* performers who pass the workload to themselves. The latter type of relationships appears to have significantly higher total count than the connections *between* resources, despite the case occurrences staying very similar. We can therefore conclude that larger sets of subsequent activities tend to be performed by the same resource, whereas handing over the workload to other employees occurs less often.

The comparison of the event and case involvement rates, i.e., respectively the share on the total number of recorded events, and the proportion of cases in which a particular resource was involved (see Table 3), provides some assumptions towards the roles of the involved employees. It can be assumed that those with a

Connection	Cases	Total
BNH49 ⟶ UJH83	143	179
CPI61 ⟶ BNH49	109	127
PXC22 ⟶ GYS85	105	131
CPI61 ⟶ CPI61	227	541
ZHC41 ⟶ ZHC41	111	354
XJQ61 ⟶ XJQ61	109	276

Table 2: Highest connection counts *between* and *within* resources, with case occurrences and absolute connection count

Performer	Cases	Events
DSV86	28%	4%
BNH49	25%	3%
CPI61	19%	8%
EVZ89	19%	4%
JUN84	17%	3%
GYS85	17%	2%
ZCL32	16%	4%
PXC22	16%	3%
ZHC41	15%	5%

Table 3: Case and event involvement percent rates for the top 10 performers (based on case involvement)

higher share of executed events perform more activities *within* a process instance. On the other hand, the resources with lower event involvement rates, though active in many cases, are assumed to act as reviewers and approvers of the offering proposals who are only responsible for a few specific activities.

Considering human resources with regards to the performance aspects of the process, we set an initial hypothesis that an increasing count in activity instances per case leads to an increase in the number of employees involved in the process execution. However, in process instances containing loops, the activities within the new iterations are assumed to be performed by the same resources who have already been employed in the execution of previous iterations and know the case specifications. This is reflected in lower growth rate of distinct human resources per case.

The scatter plot in Figure 8 supports our hypothesis. Since the data exhibit a rather quick initial increase that then levels out, we computed a logarithmic trendline for the development of "resource population" per case with growing activity counts. With the available process knowledge, the log-trendline can be interpreted as follows: The initial quick increase in resource counts per case is linked to the difference between cases with standard and non-standard services, where the resource-consuming pre-sales phase is required. As the employees of the pre-sales team differ from those who actually create the offering documents, the differences between these two groups are higher. Subsequently, this results in a quick increase of resource counts. However, further increase in activity count per case leads to a slower growth rate in the resource count. These findings support our hypothesis that with multiple iterations in the process, the repeated activities are executed by the same human resources who have already been involved in previous iterations and the "intake" rate of new employees levels out. The level-log regression equation, $resources = -0.7005 + 2.6545 \ln(activities)$, shows that an increase in the activity count per case by one percent, we expect the resource count to increase by (2.6545/100) human resources. The coefficient of determination of 0.7372 indicates a good fit of the regression model with a significant result ($p < 0.001$).

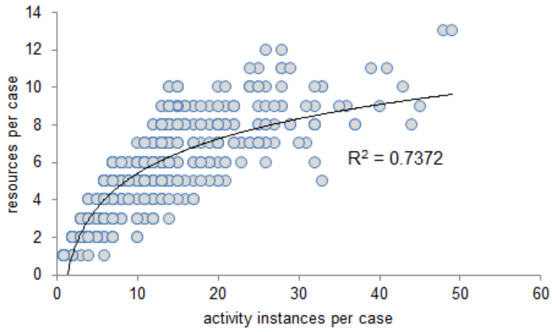


Fig. 8: Scatter plot visualising the relationship between the number of activity instances per case and number of human resources involved in process execution

5 Discussion

Our study evidenced that applying methods of automated process discovery can yield useful insights into both the performance and the social structure of business processes. By analysing event log data of an ERP system, we detected a significant number of differing execution variants among the recorded cases of the investigated process. We have shown that duration can be influenced by the number of activities per case, resources and number of loops. The presence of a pre-sales phase in offering workflows covering non-standard services often causes a longer case duration and higher numbers of employees involved in the process. Therefore, we propose the portfolio with standard services of our industrial partner to be extended by new services that have previously successfully met customer expectations. This could contribute to reducing the number of cases with the pre-sales phase and the execution time of the process.

From the offering proposals recorded in the investigated log, around 56.5% were accepted, a figure that our industrial partner aims to improve in the future. On the other hand, over 70% of the contracts were concluded after only one process iteration, which is a positive result. Considering the presence of loops in the process, the investigated cases were performed in a relatively straight-forward way with almost 80% of cases finished without any loops. Therefore, the initial suspicion that the majority of cases included several iteration loops was not confirmed. Still, it is proposed to introduce a comprehensive list of all customer change requests once the offering documents are returned for adjustments (and thus triggering new loop). That way, future performance of the process in the first iteration can be improved.

Based on the events analysis, the case counts and the social network diagram, the roles of the recorded resources were identified. Interestingly, the very high number of employees involved in the process execution was in fact unknown to our industrial partner. Even though 258 employees were involved in the examined cases, most of the activities were performed by only few individuals. Therefore, it is advised to assess whether to distribute the workload more evenly among the available employees or to reduce the resource counts for this particular process to only the key personnel.

Our research was limited by the restriction on tools that we were allowed to use by the industrial partner and by the lack of information about the specifications of the IT system used as the data source. Nonetheless, it can be concluded that these results represent a solid basis for future decisions with regards to process optimisation.

6 Conclusion

In this paper, we demonstrated how process mining can be used to extract valuable knowledge about business processes from transactional data. Our study provides evidence that by exploiting all attributes of the event log, it is possible to obtain extensive knowledge about the dynamics, performance and the resource structure of the examined process.

Our investigation was hampered by a relatively high number of events for which the resource was not specified. Such events were also associated to missing end-timestamps. Therefore, we chose not to include those events in the analysis of both the process variants and the social network, as they would have yielded biased results otherwise. The lack of reported information often affects real-world logs. It is therefore in the future plans to integrate the techniques of Rogge-Solti et al. [17, 18] to replace missing values with estimates, so as not to disregard events which may otherwise be relevant because of other information they bring. Considering the social perspective, we plan to acquire further knowledge about the organisational structure of the enterprises in future case studies as well as the role allocation of the available individuals involved in the process execution. This is of particular importance when verifying whether the designated responsibilities set for the execution of specific activities by the enterprise are not violated.

In the light of the promising results achieved, it is in our plans to collaborate further with our industrial partner to deepen the investigation by means of other process mining techniques, in the spectrum both of process discovery and conformance checking. Furthermore, the high variability of recorded process instance executions could also suggest that clarifying results might derive from declarative process discovery [13, 9, 12]. The declarative approach indeed aims at understanding the behavioural constraints among activities that are never violated during the process enactment, rather than specifying all variants as branches of an imperative model. As a consequence, it is naturally suited to flexible workflows [4].

References

1. van der Aalst, W.: Process Mining: Data Science in Action. Springer (2016)
2. van der Aalst, W.M.P., van Dongen, B.F., Günther, C.W., Rozinat, A., Verbeek, E., Weijters, T.: Prom: The process mining toolkit. In: BPM (Demos). CEUR Workshop Proceedings, vol. 489. CEUR-WS.org (2009)

3. van der Aalst, W.M.P., van Hee, K.: Workflow Management: Models, Methods, and Systems. MIT Press, Cambridge, MA, USA (2002)
4. van der Aalst, W.M.P., Pesic, M., Schonenberg, H.: Declarative workflows: Balancing between flexibility and support. Computer Science - R&D 23(2), 99–113 (2009)
5. van der Aalst, W.M.P., Reijers, H.A., Weijters, A.J.M.M., van Dongen, B.F., de Medeiros, A.K.A., Song, M., Verbeek, H.M.W.E.: Business process mining: An industrial application. Inf. Syst. 32(5), 713–732 (2007)
6. van der Aalst, W.M.P., Weijters, T., Maruster, L.: Workflow mining: Discovering process models from event logs. IEEE Trans. Knowl. Data Eng. 16(9), 1128–1142 (2004)
7. Agrawal, R., Gunopulos, D., Leymann, F.: Mining process models from workflow logs. In: EDBT. Lecture Notes in Computer Science, vol. 1377, pp. 469–483. Springer (1998)
8. Castellanos, M., Alves de Medeiros, A.K., Mendling, J., Weber, B., Weijters, A.J.M.M.: Business process intelligence. In: Cardoso, J. (ed.) Handbook of Research on Business Process Modeling, chap. XXI, pp. 467–491. IGI Global (2009)
9. Di Ciccio, C., Mecella, M.: On the discovery of declarative control flows for artful processes. ACM Trans. Management Inf. Syst. 5(4), 24:1–24:37 (2015)
10. Dumas, M., Rosa, M.L., Mendling, J., Reijers, H.A.: Fundamentals of Business Process Management. Springer (2013)
11. Günther, C.W., van der Aalst, W.M.P.: Fuzzy mining - adaptive process simplification based on multi-perspective metrics. In: BPM. Lecture Notes in Computer Science, vol. 4714, pp. 328–343. Springer (2007)
12. Kala, T., Maggi, F.M., Di Ciccio, C., Di Francescomarino, C.: Apriori and sequence analysis for discovering declarative process models. In: EDOC. pp. 1–9. IEEE Computer Society (2016)
13. Maggi, F.M., Mooij, A.J., van der Aalst, W.M.P.: User-guided discovery of declarative process models. In: CIDM. pp. 192–199. IEEE (2011)
14. Mans, R.S., Schonenberg, H., Song, M., van der Aalst, W.M.P., Bakker, P.J.M.: Application of process mining in healthcare - A case study in a dutch hospital. In: BIOSTEC (Selected Papers). Communications in Computer and Information Science, vol. 25, pp. 425–438. Springer (2008)
15. Mans, R.: Process mining in healthcare (2015)
16. de Medeiros, A.K.A., Weijters, A.J.M.M., van der Aalst, W.M.P.: Genetic process mining: an experimental evaluation. Data Min. Knowl. Discov. 14(2), 245–304 (2007)
17. Rogge-Solti, A., Mans, R., van der Aalst, W.M.P., Weske, M.: Improving documentation by repairing event logs. In: PoEM. Lecture Notes in Business Information Processing, vol. 165, pp. 129–144. Springer (2013)
18. Rogge-Solti, A., Weske, M.: Prediction of business process durations using non-markovian stochastic petri nets. Inf. Syst. 54, 1–14 (2015)
19. Schönig, S., Cabanillas, C., Jablonski, S., Mendling, J.: A framework for efficiently mining the organisational perspective of business processes. Decision Support Systems 89, 87–97 (2016)
20. Van Steen, M.: Graph Theory and Complex Networks: An Introduction (2010)
21. Weijters, A.J.M.M., van der Aalst, W.M.P.: Rediscovering workflow models from event-based data using little thumb. Integrated Computer-Aided Engineering 10(2), 151–162 (2003)
22. Werner, M., Gehrke, N.: Multilevel process mining for financial audits. IEEE Transactions on Services Computing 8(6), 820–832 (Nov 2015)

References

Puchovsky, Matej, Claudio Di Ciccio, and Jan Mendling (2016). “A Case Study on the Business Benefits of Automated Process Discovery”. In: *SIMPDA*. Ed. by Paolo Ceravolo, Christian Guetl, and Stefanie Rinderle-Ma. Vol. 1757. CEUR Workshop Proceedings. CEUR-WS.org, pp. 35–49. URL: http://ceur-ws.org/Vol-1757/paper3.pdf.

BibTeX


```
@InProceedings{   Puchovsky.etal/SIMPDA2016:BenefitsAutomatedProcessDiscovery,
  author        = {Puchovsky, Matej and Di Ciccio, Claudio and Mendling,
                  Jan},
  title         = {A Case Study on the Business Benefits of Automated Process
                  Discovery},
  booktitle     = {SIMPDA},
  year          = {2016},
  pages         = {35--49},
  crossref      = {SIMPDA2016},
  keywords      = {Process mining; process analytics; business data
                  analysis},
  url           = {http://ceur-ws.org/Vol-1757/paper3.pdf}
}
@Proceedings{     SIMPDA2016,
  title         = {Proceedings of the 6th International Symposium on
                  Data-driven Process Discovery and Analysis {(SIMPDA} 2016),
                  Graz, Austria, December 15-16, 2016},
  year          = {2016},
  editor        = {Paolo Ceravolo and Christian Guetl and Stefanie
                  Rinderle{-}Ma},
  volume        = {1757},
  series        = {{CEUR} Workshop Proceedings},
  publisher     = {CEUR-WS.org},
  urn           = {urn:nbn:de:0074-1757-5}
}
```