Business Process Comparison: A Methodology and Case Study

Alifah Syamsiyah[1], Alfredo Bolt[1], Long Cheng[1], Bart F.A. Hompes[1], R.P. Jagadeesh Chandra Bose[2], Boudewijn F. van Dongen[1], and Wil M.P. van der Aalst[1]

[1] Eindhoven University of Technology, The Netherlands
{a.syamsiyah, a.bolt, l.cheng, b.f.a.hompes, b.f.v.dongen, w.m.p.v.d.aalst}@tue.nl
[2] Xerox, India
jagadeesh.prabhakara@conduent.com

Abstract. Business processes often exhibit a high degree of variability. Process variants may manifest due to the differences in the nature of clients, heterogeneity in the type of cases, etc. Through the use of process mining techniques, one can benefit from historical event data to extract non-trivial knowledge for improving business process performance. Although some research has been performed on supporting *process comparison* within the process mining context, applying process comparison in practice is far from trivial. Considering all comparable attributes, for example, leads to an exponential number of possible comparisons. In this paper we introduce a novel methodology for applying process comparison in practice. We successfully applied the methodology in a case study within Xerox Services, where a forms handling process was analyzed and actionable insights were obtained by comparing different process variants using event data.

Keywords: Process Comparison · Process Mining · Business Analytics.

1 Introduction

Modern information systems and devices collect and store large amounts of event data. For instance, ERP systems record business transaction events, and high-tech systems such as X-ray machines record an abundance of events [10]. Such historical *event data* can be used to extract non-trivial knowledge and interesting insights that can be used for further analysis. Increasingly process mining techniques are used to analyze such data [20]. Process mining covers three types of analysis [19]: *process discovery* automatically extracts a process model from an event log; *conformance checking* measures how well the behavior recorded in an event log fits a given process model and vice versa; and *process enhancement* is concerned with extending or improving an existing a-priori process model using event data.

In process mining, process characteristics such as *waiting times, throughput times,* and *utilization rates* are typically of interest, and can be obtained from real-life event data. Many processes exhibit a high degree of variability. There may be major differences between processes and their variants, due to an abundance of factors such as temporal changes, the geographical location of the process execution, the involved resources and the overall context in which a process is executed [1, 6]. In such scenarios, our research question is: how can we conduct a comparative analysis of different processes and their variants in real businesses? Based on the results of the analysis, we should be able to find the differences between multiple processes and also find root causes for inefficiencies such as delays and long waiting times for the interpretation of process behaviors. Moreover, domain experts should also able to identify the precise events that correspond to unusual behavior, and consequently devise concrete measures to improve their business processes [17].

In this paper, we present a methodology for business process comparison. We do so by presenting an overall methodology and an instantiation thereof in the context of a large service delivery organization: Xerox Services. This organization caters similar processes across several clients, hence process variants may manifest due to the differences in the nature of clients, heterogeneity in the type of cases, etc. Moreover, the organization's operational Key Performance Indicators (KPIs) across these variants may widely vary. We show that, using our method, we gain insights into the differences between variants and we leverage these insights on non-performing variants by means of process comparison.

The highlighted contributions of this paper are as follows:

- Present a methodology for process comparison which focuses on the analysis of multiple processes. This methodology considers multiple perspectives, such as control flow, organizational, data, performance, etc.
- Validate the methodology in a case study using real-life data.

The remainder of this paper is organized as follows. In Section 2, we discuss related work in process comparison and process mining methodologies. Then, we explain the proposed process comparison methodology in Section 3 and apply the methodology in a case study in Section 4. Section 5 concludes the paper.

2 Related Work

In recent years, the value of process mining techniques has been demonstrated in *case studies* across different domains such as healthcare [11, 21, 25], industry [12, 13, 15], insurance [17], and finance [7, 8]. However, few *methodologies* have been proposed to carry out process mining projects in a structured manner. In [2], the *Process Diagnostics Method* (PDM) is proposed to quickly obtain a broad overview of the process at hand, without the need for any domain knowledge. As such, it can be used to steer a process mining project by providing initial insights and analysis opportunities. For example, the method has been adopted for the analysis of healthcare processes in [14]. In [19], the *L* life-cycle model* is proposed as an approach for mining processes. L* covers many techniques, and describes the life-cycles of a typical process mining project aiming to improve processes. Since PDM focuses on providing a broad overview using a limited set of process mining techniques and because L* is aimed at the analysis of structured processes, the authors of [23] proposed *PM^2: a Process Mining Project Methodology*. PM^2 is designed to support projects aiming to improve process performance or compliance, and focuses on iterative analysis. Its applicability was shown by a case study conducted on data provided by IBM. Like L*, PM^2 covers a wide array of process mining techniques. Contrary to L*, however, PM^2 is suitable for the analysis of both structured and unstructured processes.

A common pitfall of the discussed process mining methodologies is that the focus is on the analysis of a single process, as mentioned in [23]. As such, *process comparison* remains an interesting but insufficiently researched topic. The comparison of processes based on event logs has been the focus of several papers [1, 4, 5, 18]. However, most process comparison approaches take into consideration only the control-flow aspect (i.e., presence, routing and frequency of activities), while ignoring other dimensions.

Given the increased interest in process comparison from perspectives other than just control flow, and the lack of methodological support for applying process comparison in a process mining project in practice, we propose The Process Comparison Methodology (PCM). In this work, different from existing process mining methodologies, we introduce a novel methodology by considering multiple aspects, such as the organizational aspect (i.e. the involved resources, roles, and groups), the data aspect (attribute values), the performance aspect, etc. We validate our methodology in a case study using real-life data provided by Xerox Services. To the best of our knowledge, this is the first work that methodologically considers business process comparison from multiple perspectives.

3 The Process Comparison Methodology (PCM)

When comparing multiple processes, it is common that those processes have mutual attributes for categorization. When process comparison methods are applied to highlight the differences between similar categorized processes, the results are more detailed and representative than when comparing dissimilar or unrelated processes. Based on these underlying assumptions, in this section, we introduce a methodology for process comparison which considers many perspectives. Our methodology comprises of five main phases as depicted in Figure 1.

First, the data pre-processing phase transforms raw data to standardized event log formats such that existing process mining techniques can be applied. Next to the event logs, the so-called α-attributes are selected. These attributes are case-level attributes that identify the variants of interest. Next, in the scoping analysis phase, the interesting cases to be used for answering the analysis questions are identified. In the third phase, comparable sub-logs are generated by aggregating similar cases. Fine-grained analysis of the generated sub-logs

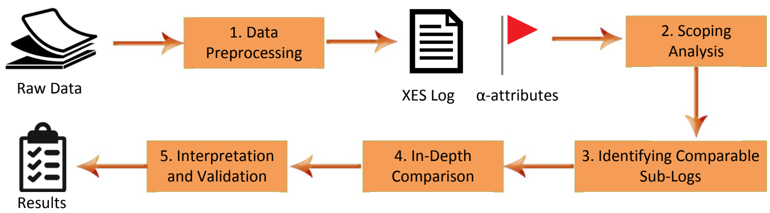

Fig. 1. Process Comparison Methodology (PCM)

takes place in the in-depth comparison phase. Finally, the discovered results are delivered to the process owners.

Phase 1. In the data pre-processing phase, raw data is translated to standardized event log formats such that existing process mining techniques can be applied. We have two main objectives for the data pre-processing phase: (1) refine event data collected by information systems; and (2) create an event log and identify a set of case attributes α to be used in the comparison process.

Typically, raw event data are collected by different information systems at different levels of granularity. To conduct a meaningful analysis, we combine all collected event data and merge them into a single collection of events. Here, standard data cleaning techniques can be used if the raw data contains noise. From this event collection, an event log is devised. To get an event log from an event collection, a notion of cases is introduced. A case refers to a time-ordered sequence of events relating to the some underlying concept, for example a purchase order or a single run of a machine, (i.e. events need to be correlated to form traces of events). We follow the XES standard [24] as the format for the generated event log, to make existing process mining techniques (implemented in tools such as ProM[1]) accessible in the following phases of our methodology. Finally, next to the case notion, attributes of interest are selected as the so-called α-attributes. In the further comparison, we consider the α-attributes to denote the process variant.

Phase 2. Once the event log and the α-attributes are defined, we scope our analysis. The goal of the scoping phase is to limit the number of comparisons to be executed later. Typically, scoping is done based on the α-attributes, for example by selecting the most frequent values of these attributes. However, in general, the scoping decision must follow the business questions and the goal of doing process comparison. As a result of scoping, a collection of sub-logs is generated, again in the XES format.

Phase 3. The next phase in the analysis is the identification of comparable sub-logs. Each of the sub-logs obtained during scoping refers to a variant of the process under investigation. However, these variants are not always directly

[1] See `http://processmining.org` and `http://promtools.org`

comparable. They may, for example, consist of disjoint sets of activities. Therefore, in this phase, we select comparable variants (i.e. variants that have enough commonalities).

The identification of comparable sub-logs can be done in several ways. For example, we can use domain knowledge to manually select sub-logs to be compared. Alternatively, if domain knowledge is not available, *clustering* techniques can be used to group sub-logs based on a quantifiable *similarity* notion [16].

Phase 4. After sets of comparable sub-logs are identified, we treat each set as the starting point for the in-depth comparison phase. In this process, the sub-logs in each set will be pairwise compared and the output of this phase will be a collection of observed and interesting differences between the input sub-logs.

For the in-depth comparison, the pairwise analysis of the sub-logs should often not be limited to control flow only. Instead, other aspects of processes, such as performance characteristics, resource utilization and compliance aspects should be considered. Most importantly, the *influence* of these aspects on each other should be investigated. For example, cases in which different resources were involved could have significantly different durations, which might be an actionable insight.

It should be noted that only the *relevant* and *impactful* differences are of interest to the process owner. For example, a difference in case duration of several seconds may be irrelevant in processes where the average case duration is in the order of days, while in processes that generally last minutes this difference can be significant.

Phase 5. After completing the in-depth comparison for each cluster and having identified relevant and impactful differences, the relevant results will be be reported to the process owner. We identify two activities for this phase:

1. *Presentation and Interpretation.* After the process mining analysis has been performed, we obtain facts about the process. Most of the time, these facts are raw and disconnected with each other. Therefore, to provide meaningful information at the business level, an additional presentation and interpretation step is needed. The significance of the results depends on how well the analysis and interpretation step is executed.
2. *Validation.* The results from the in-depth comparison have to be validated with the process owner and participants in the process.

In the remainder of this paper, we show how this high-level methodology can be executed in a concrete case study within Xerox. We use publicly available tools and techniques on proprietary data and we closely involved the Xerox stakeholders in the analysis.

4 Xerox Case Study

This section discusses the application of PCM on a case study conducted within Xerox Services. The study involved a real-life data set with millions of events. First, we explain the data set in terms of its structure and its origin, and give a

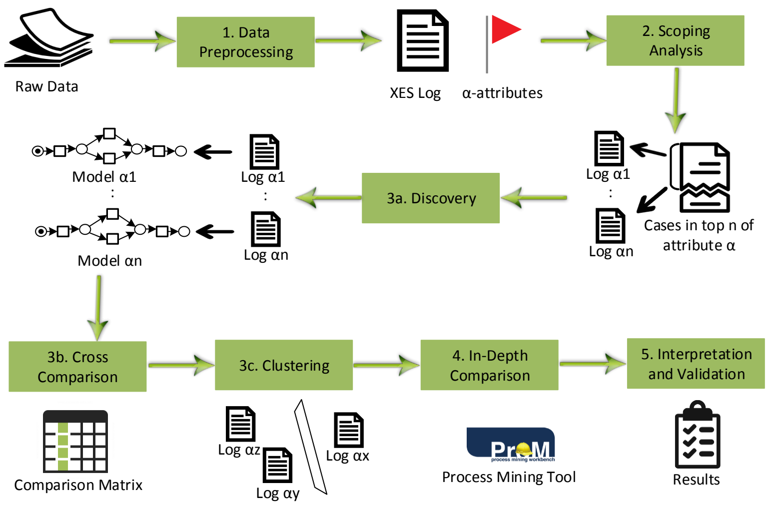

Fig. 2. Process Comparison Methodology applied to a Xerox Dataset

description of the process contained in it. Then, we present the application of our proposed methodology in detail. As demonstrated in Figure 2, the instantiation of each phase in our application corresponds to the five phases in Figure 1, respectively. However, for the case study, the phase *Identifying Comparable Sub-Logs* is refined into three smaller phases here: *Discovery*, *Cross Comparison*, and *Clustering*. This refinement choice is one of many ways to identify comparable sub-logs.

In our implementation, we used both ProM (for steps 1 and 4 in Figure 2) and RapidProM (for steps 2, 3a, 3b, and 3c). The used RapidProM workflow is depicted in Figure 3 and available at `https://goo.gl/BCq1u0`.

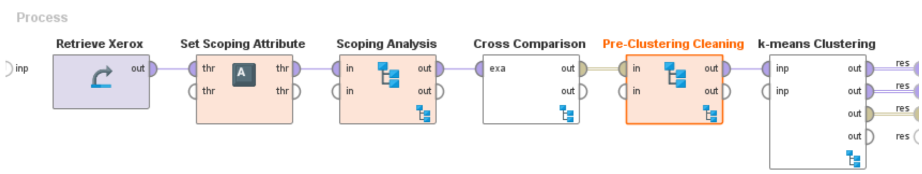

Fig. 3. The RapidProM workflow used for scoping analysis and identifying comparable sub-logs

4.1 Data Set

We analyzed event logs pertaining to the transaction processing business unit within Xerox Services. More specifically, we analyzed the process pertaining to the data entry back-office operations of insurance claim forms. Forms submitted to the insurance providers need to be digitized before the claims can be processed. *Business Process Outsourcing* (BPO) organizations assist the insurance providers in this process. Forms received by the BPO organization are classified and sorted depending on the type of form (e.g. HCFA, UB04, Dental, etc.). More fine-grained classifications further refining each type are possible (e.g. HCFA standard, HCFA careplus, etc.), thereby defining a taxonomy. Different classes in the taxonomy are divided into so-called batches, where each batch caters to one type of insurance claim form (e.g. HCFA standard).

A transaction refers to the data entry operations of one instance of an insurance claim form. The organization handles data entry operations of millions of such instances of insurance claim forms. In this paper, we only consider the transactions of one month pertaining to one client, but the approach can be applied to even larger data sets. Furthermore, different attributes concerning the execution of events such as involved resourced and properties of the forms are recorded as well. The complete dataset used here contains information on hundred transactions comprising 20 million events divided across 94 batches. The organization is interested in analyzing the processes followed across different batches and wants to obtain insights on their executions. In this paper, we focus on the analysis of three batches, where two are similar but not identical and the third batch is different from the other two.

4.2 Data Preprocessing

We transformed the raw event data obtained as CSV file to a standard XES log with the *Convert CSV to XES* plugin in ProM. To make this transformation meaningful and successful, we have done the following three main pre-processing steps. (1) We enriched the set of attributes based on anticipated questions. Since we are interested in analyzing different batches (see subsection 4.1), we set the attribute `BATCHNAME` as the α attribute to be used in comparison process. (2) We refined data into event level. Each activity in the input log includes two timestamps, indicating its start and end point, therefore we divide each activity into two events based on that. (3) We removed uninteresting/uncompleted cases from the log. Based on statistics on the start and end activities for all cases, we removed those case that have a start or end activity that does not appear frequently enough. Through this process, we removed 318,002 cases, and the output XES log contains 936,720 cases and 31,660,750 events.

4.3 Scoping Analysis

We implemented our scoping analysis using RapidProM (as depicted in Figure 4) to select the interesting batches (batches that are infrequent will not be considered in our analysis). For the generated XES log in the preprocessing phase,

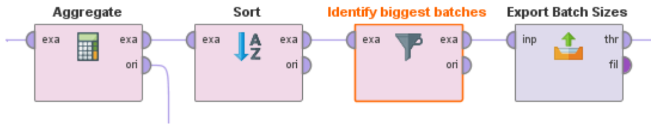

Fig. 4. Scoping analysis to select the most frequent batches

we first aggregated all the events based on their `BATCHNAME` values. Then, we filtered out the popular batches based on their occurrence frequency. There are 94 different batches in our log. We selected the 10 most frequent ones. Their corresponding batch identifiers are 1, 4, 2, 11, 7, 18, 3, 58, 23, and 30 respectively, each having between 424,560 and 8,684,476 cases. We divided the XES log into 10 sub-logs according to the chosen batch names, and conducted our process analysis using these sub-logs.

4.4 Identifying Comparable Sub-Logs

Given a collection of sub-logs from the previous phase, the next step is to identify subsets such that sub-logs within each subset share similar behavior (i.e. they are comparable to each other). In the next phase, the sub-logs within one such subset will be compared to obtain more refined comparison results. In this section, we explain the different steps of the techniques we used to identify comparable sub-logs. In Figure 2, these steps refers to phases 3a, 3b, and 3c.

Discovery. Based on our goals, we compared the extracted batches based on the analysis of their high-level process models. These models can be retrieved by current process mining techniques. Various process discovery algorithms such as the Alpha Algorithm [20], ILP Miner [22] and Inductive Miner [9] have been proposed in the past years. Considering the amount of events in our logs as well as the quality of discovered processes (e.g., soundness and fitness), we have chosen the Inductive Miner. Besides the fact that the Inductive Miner is the state-of-the-art process discovery algorithm, other techniques are inclined to produce models that are unable to replay the log well, create erroneous models, or have excessive run times for event logs of this size. The output of this phase is a collection of process models per sub-log.

Cross Comparison. In [3], Buijs coined the term *cross comparison* and presents the so-called comparison table, which has been evaluated in a case study using event data from five municipalities. A comparison table consists of three types of metrics, namely *process model metrics, event log metrics,* and *comparison metrics. Process model metrics* are metrics calculated using only the process model, such as total number of nodes in the process model, cyclicity, or concurrency in the process model. *Event log metrics* are metrics calculated based on event log, such as the total number of traces and events, average trace duration, etc.

Table 1. Example cross-comparison table showing the cross comparison between logs and models in different batches.

	$Model_1$	$Model_2$	$Model_3$	...
Log_1	0.64	0.37	0.25	...
Log_2	0.26	0.68	0.6	...
Log_3	0.25	0.69	0.61	...
...	...	...	...	...

Comparison metrics are used to compare modeled and observed behavior and include metrics such as fitness, precision, generality, and simplicity [19].

In this phase, we apply the fitness comparison metric to the sub-logs and their corresponding discovered models. We choose fitness rather than the other metrics due to the need of Xerox Services to have process models which allow for most of the observed behavior. Table 1 shows an excerpt of the cross comparison using fitness metric between logs and models in different batches. Each row represents a log of a particular batch n (Log_n), and each column represents a discovered model from a particular batch m ($Model_m$). Each cell contains the fitness value after replaying a log into a process model.

Clustering. Based on the cross-conformance checking results, we grouped the sub-logs (i.e. batches) into clusters using k-means clustering. We chose this clustering algorithm because of the information from domain expert that a batch belongs to a single cluster and thus cluster overlap is not possible. Concretely, we used the rows of the cross-conformance matrix (Table 1) as observations to perform a *k-means* clustering. In our experiments, we used k = 3 clusters, and we identify the clusters as follows: (1) cluster 0: batches 3, 4, 18, 23, (2) cluster 1: batches 1, 2, 7, 11, 52, (3) cluster 2: batch 30.

The resulting clusters contain groups of similar (i.e. comparable) batches. Comparative analysis can be performed within any of the clusters. Note that cluster 2 contains only one batch (batch 30). This can be caused by the fact that batch 30 is very different from all other batches.

4.5 In-Depth Comparison

Once clusters of comparable batches have been identified, we can proceed to compare the batches in each cluster. To illustrate this, we apply two process comparison techniques to the batches contained in cluster 0. The first technique (as introduced in [1]) detects statistically significant differences between two sub-logs in terms of control-flow and overall performance. The results of this technique identify those parts of the process where differences occur. However, they do not explain why such differences manifest. The second technique (as introduced in [6]) tackles this issue by analyzing, for each sub-log, the effect that different contexts (e.g. involved resources, data attributes, control-flow, etc.) have on process performance, and whether that effect is statistically significant. Using these two techniques on the batches contained in cluster 0, we could obtained valuable insights.

We first applied the process comparison technique [1] to the four sub-logs of cluster 0 to get a ranked list of pairs of sub-logs. After sorting the list based on the percentage of control-flow differences between the pairs of sub-logs, we found that: batches 3 vs. 18 (38.04% control-flow difference), batches 4 vs. 23 (42.03%), batches 3 vs. 23 (72.16%), batches 18 vs. 23 (73.40%), batches 3 vs. 4 (78.43%), and batches 4 vs. 18 (78.64%). This means that batches 4 and 18 are the most dissimilar pair within cluster 0, and batches 3 and 18 are the most similar. In order to illustrate the in-depth comparison phase, in the remainder of this section we will analyze the differences between batches 4 and 18 and between batches 3 and 18.

In Figure 5, we provide an example of the control-flow differences found between batches 4 and 18. The dark-blue colored states are executed only in batch 18, and never in batch 4. These states are related to *optical character recognition* (OCR) in forms. Moreover, an example of the performance differences found between batches 4 and 18 is also shown in Figure 6. We can see that the duration of the activity *Entry* is statistically significantly higher in batch 18 than in batch 4. This activity refers to manual entry of form content.

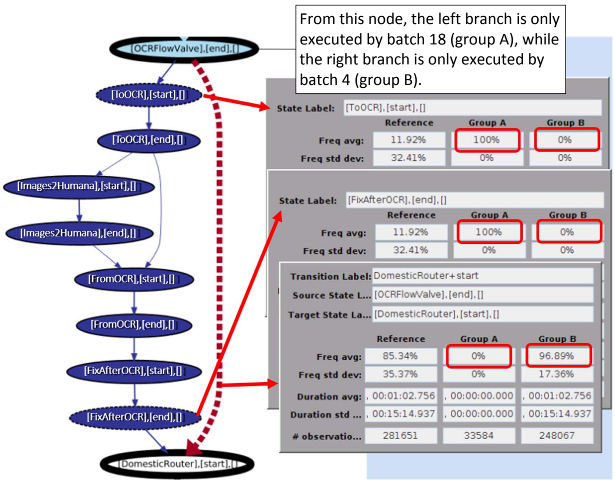

Fig. 5. Control-flow differences between batch 18 (group A) and batch 4 (group B). The activities *ToOCR, Images2Humana, FromOCR, FixAfterOCR* are executed only in batch 18.

To see whether there are any significant differences manifested in the behavior of the similar batches 3 and 18, we also conducted another comparison using the same technique. From Figure 7, we can see a significant difference in the frequency of execution of the process fragment, corresponding to the transformation of data from XML to the X12 format, and the transmission and acknowledgment

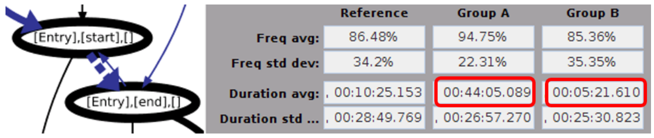

Fig. 6. Performance differences between batch 18 (group A) and batch 4 (group B). The average duration of the *Entry* activity is 44 mins for batch 18 and 5 mins for batch 4.

of that data. This fragment is almost always executed in batch 18, it is executed only in approximately 93% of the cases in batch 3. Similarly, the *Cleanup* activity is executed in only 5% of the cases in batch 18 against 12% in batch 3. From a performance point of view, we see that there is a significant difference in the average duration of cases until the execution of the *Cleanup* activity (22 days vs. 10 days). Note that besides this difference, for both batches, the standard deviation of the duration until *Cleanup* is very high relative to the average duration.

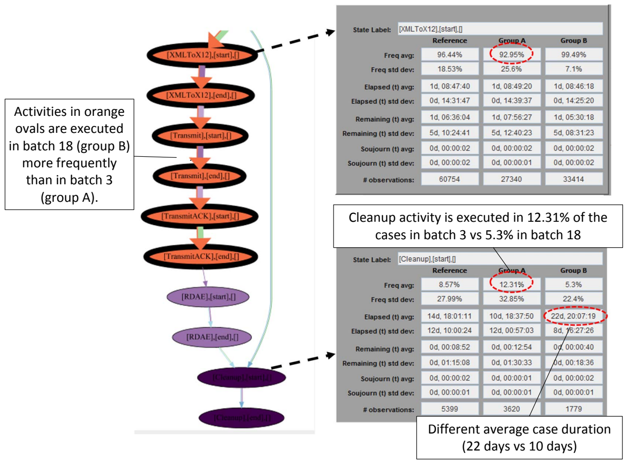

Fig. 7. Example of differences found between batch 3 (group A) and batch 18 (group B).

We analyzed the observed differences between the three batches in more detail using the context-aware performance analysis technique from [6]. This analysis revealed that, for batch 18, significant differences exist between the resources that execute the *Entry* activity (in terms of the waiting time for the activity). This observation is shown in Figure 8. The waiting times range from several hours to multiple days, and hence might be worth looking into. As explained, the standard deviation for the case duration until the *Cleanup* activity between

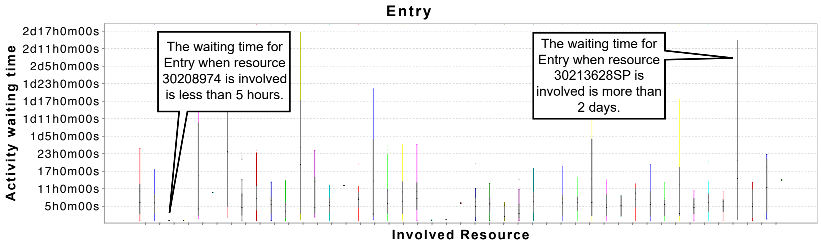

Fig. 8. The resources involved in the *Entry* activity in batch 18 lead to different waiting times.

batches 18 and 3 is quite high relative to the average duration. This observation was analyzed in more detail as well. We found that the duration until *Cleanup* showed big differences between the days in which the cleanup activity happened. In some dates, the duration until *Cleanup* took several days while in other dates, it took multiple weeks. This is illustrated in Figure 9.

4.6 Delivering Results

Our results discussed above have been presented to and confirmed by a domain expert. (1) The control-flow differences in Figure 5 are attributed to the fact that the two batches deal with different types of forms. Batch 18 deals with UB-04 forms, a claim form used by hospitals, nursing facilities, in-patient, and other facility providers. These forms are filled by healthcare providers and they can contain handwriting (e.g. disease codes, diagnosis, etc.), so OCR is needed. In contrast, batch 4 deals with claim correspondence forms (i.e. reply forms from the provider). These forms are typically digital. Hence there is no need for OCR. (2) The performance difference in Figure 6 is attributed to the fact that the forms related to batch 4 (i.e. correspondence forms) are usually smaller than the forms related to batch 18 (i.e. UB-04 forms), and have little content to be

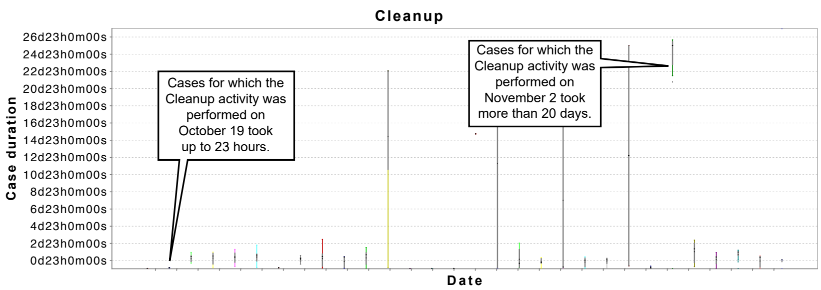

Fig. 9. The duration until the *Cleanup* activity in cases in batch 3 varies highly between days.

entered manually. Hence, the average duration of *Entry* activity in batch 4 is lower. Although these differences between batch 18 and 4 are insightful, they are not very surprising. Similarly, the differences in duration in the manual entry of smaller vs. larger forms in terms of page and image count is to be expected as well. However, the differences in waiting times for different resources are surprising and need to be investigated in order to avoid delays.

The differences between batches 3 and 18 have also provided interesting actionable insights. Both batches 3 and 18 correspond to a similar type of form (UB-04) and are expected to have very similar behavior. The remarkable differences in the frequencies in the process fragment are statistically significant and moreover unexpected by the domain expert, and hence need further investigation. The observed differences in duration until the *Cleanup* activity can be explained by the fact that, in the analyzed process, a lot of (sub) batch processing is involved, and as such, cases sometimes need to wait for other cases in order to be processed.

5 Conclusion

In this paper we have introduced a novel methodology for process comparison within the process mining context, which aims at efficiently examining the differences between multiple business processes and process variants. The proposed methodology, called The Process Comparison Methodology (PCM), considers multiple perspectives during comparison, such as control flow, organizational, data, and performance.

PCM consists of five main phases. First, the *data pre-processing* phase transforms raw data to standardized event log formats. Secondly, the *scoping analysis* phase creates sub-logs based on some case attributes values. Next, the interesting sub-logs to be used for answering analysis questions are identified. Then, in the *identifying comparable sub-logs* phase, similar sub-logs are aggregated to generate comparable sub-logs. In the *in-depth comparison* phase, fine-grained analysis is conducted within comparable sub-logs. Finally, the results are interpreted and validated in the *interpretation and validation* phase and the discovered insights and actions are delivered to the process owners.

The practical relevance of PCM is shown in a case study using real-life data provided by Xerox Services. The process pertains to the data entry back-office operations of insurance claim forms. The organization is interested in analyzing the processes followed across different batches. As there are 94 batches it was unfeasible to compare each pair in detail. Through the application of our methodology, however, very meaningful results were obtained, confirmed by a domain expert, and transformed into actionable insights such as studying the root causes and contextual circumstances for the aberrant instances.

In the future, we would like to investigate more techniques related to the comparison of business processes in order to further refine our methodology. Moreover, we would also like to study more relevant business questions through our collaboration with Xerox Services.

Bibliography

[1] A. Bolt, M. de Leoni, and W.M.P. van der Aalst. A Visual Approach to Spot Statistically-Significant Differences in Event Logs Based on Process Metrics. In *CAiSE*, volume 9694, pages 151–166. Springer, 2016.

[2] M. Bozkaya, J. Gabriels, and J.M.E.M. van der Werf. Process Diagnostics: A Method Based on Process Mining. In A. Kusiak and S. Lee, editors, *eKNOW 2009*, pages 22–27. IEEE Computer Society, 2009.

[3] J.C.A.M. Buijs. *Flexible Evolutionary Algorithms for Mining Structured Process Models*. PhD thesis, TU Eindhoven, 2014 (cit. on p. 179), 2014.

[4] J.C.A.M. Buijs and H.A. Reijers. Comparing Business Process Variants Using Models and Event Logs. In *BMMDS/EMMSAD*, pages 154–168, 2014.

[5] C. Cordes, T. Vogelgesang, and H. Appelrath. A Generic Approach for Calculating and Visualizing Differences Between Process Models in Multidimensional Process Mining. In *BPM Workshops*, volume 202, pages 383–394. Springer, 2015.

[6] Bart F. A. Hompes, Joos C. A. M. Buijs, and Wil M. P. van der Aalst. A generic framework for context-aware process performance analysis. In *On the Move to Meaningful Internet Systems: OTM 2016 Conferences: Confederated International Conferences: CoopIS, C&TC, and ODBASE 2016, Rhodes, Greece, October 24-28, 2016, Proceedings*, pages 300–317. Springer International Publishing, 2016.

[7] M. Jans, M. Alles, and M.A. Vasarhelyi. Process Mining of Event Logs in Internal Auditing: A Case Study. In *ISAIS*, 2012.

[8] M.J. Jans, M. Alles, and M.A. Vasarhelyi. Process Mining of Event Logs in Auditing: Opportunities and Challenges. *Available at SSRN 2488737*, 2010.

[9] S.J.J. Leemans, D. Fahland, and W.M.P van der Aalst. Discovering Block-Structured Process Models from Event Logs Containing Infrequent Behaviour. In *BPM*, pages 66–78. Springer, 2013.

[10] S.J.J Leemans, D. Fahland, and W.M.P. van der Aalst. Scalable Process Discovery with Guarantees. In *BPMDS/EMMSAD*, pages 85–101, 2015.

[11] R.S. Mans, 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 *BIOSTEC*, pages 425–438, 2008.

[12] Z. Paszkiewicz. Process Mining Techniques in Conformance Testing of Inventory Processes: An Industrial Application. In *BIS Workshop*, pages 302–313, 2013.

[13] M. Puchovsky, C. Di Ciccio, and J. Mendling. A Case Study on the Business Benefits of Automated Process Discovery. In *SIMPDA*, pages 35–49, 2016.

[14] A. Rebuge and D.R. Ferreira. Business Process Analysis in Healthcare Environments: A Methodology Based on Process Mining. *Inf. Syst.*, 37(2):99–116, 2012.

[15] A. Rozinat, I.S.M. de Jong, C.W. Günther, and W.M.P. van der Aalst. Process Mining Applied to the Test Process of Wafer Scanners in ASML. *IEEE Trans. Systems, Man, and Cybernetics, Part C*, 39(4):474–479, 2009.

[16] M. Song, C.W. Günther, and Wil M. P. van der Aalst. *Trace Clustering in Process Mining*, pages 109–120. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[17] S. Suriadi, M.T. Wynn, C. Ouyang, A.H.M. ter Hofstede, and N.J. van Dijk. Understanding Process Behaviours in a Large Insurance Company in Australia: A Case Study. In *CAiSE*, pages 449–464. Springer, 2013.

[18] N.R.T.P. van Beest, M. Dumas, L. García-Bañuelos, and M. La Rosa. Log Delta Analysis: Interpretable Differencing of Business Process Event Logs. In *BPM*, pages 386–405, 2015.

[19] W.M.P. van der Aalst. *Process Mining - Data Science in Action.* Springer, 2016.

[20] W.M.P. van der Aalst, A.J.M.M. Weijters, and L. Maruster. Workflow Mining: Discovering Process Models from Event Logs. *IEEE*, 16(9):1128–1142, 2004.

[21] S. van der Spoel, M. van Keulen, and C. Amrit. Process Prediction in Noisy Data Sets: A Case Study in a Dutch Hospital. In *SIMPDA*, pages 60–83, 2012.

[22] J.M.E.M. van der Werf, B.F. van Dongen, C.A.J. Hurkens, and A. Serebrenik. Process Discovery using Integer Linear Programming. *Fundam. Inform.*, 94(3-4):387–412, 2009.

[23] M.L. van Eck, X. Lu, S.J.J. Leemans, and W.M.P. van der Aalst. PM^2 : A Process Mining Project Methodology. In *CAiSE*, pages 297–313. Springer, 2015.

[24] H.M.W. Verbeek, J.C.A.M. Buijs, B.F. van Dongen, and W.M.P. van Der Aalst. Xes, Xesame, and ProM 6. In *CAiSE Forum*, pages 60–75. Springer, 2010.

[25] Z. Zhou, Y. Wang, and L. Li. Process Mining Based Modeling and Analysis of Workflows in Clinical Care - A Case Study in a Chicago Outpatient Clinic. In *ICNSC*, pages 590–595. IEEE, 2014.