Generating Performance Improvement Suggestions by using Cross-Organizational Process Mining

Onur Yilmaz and Pinar Karagoz

{onur.yilmaz, karagoz}@ceng.metu.edu.tr
Computer Engineering Department, Middle East Technical University, Turkey

Abstract Process mining is a relatively young and developing research area with the main idea of discovering, monitoring and improving processes by extracting information from event logs. With the increase of cloud computing and shared infrastructures, event logs of multiple organizations are available for analysis where cross-organizational process mining stands with the opportunity for organizations learning from each other. The approach proposed in this study mines process models of organizations and calculates performance indicators; followed by clustering of organizations based on performance indicators and finally spots mismatches between the process models to generate recommendations. This approach is implemented as an extensible and configurable plug-in set in ProM framework and tested by synthetic and real life logs where successful and suitable results are achieved with defined evaluation metrics. Generated recommendation results indicate that the use of this approach can help users to focus on the parts of process models with potential performance improvement, which are difficult to spot manually and visually.

Keywords: Process Mining, Cross-organizational Process Mining, Performance Indicators, Clustering, Process Performance Improvement

1 Introduction

Process mining is a relatively young and developing research area with the roots in computational intelligence, data mining; and process modeling and analysis [5]. Main idea in this research area is to discover, monitor and improve processes by extracting information from event logs. Traditional process mining approaches work on a single organization; however, with the increase of cloud computing and shared infrastructures, event logs of multiple organizations are currently available for analysis where cross-organizational process mining stands out. In the cross-organizational process mining area, recent studies focus on commonality and collaboration between organizations, especially on how similar the process models and behaviors of organizations under cross comparison are [11] and challenges based on partitioning of tasks and process models between organizations [2]. This study is based on the environment where processes are executed in several organizations and cross-organizational process mining is applied with the idea of unsupervised learning where predictor variables related to performances of organizations are used. In this environment, underlying assumption of the appraoch is that the correlation between performance values and mismatches hints at a causal relationship.

The approach proposed in this study is a four-stage solution and it starts with mining the process models of organizations; followed by performance indicator analysis and then mismatch pattern analysis. Finally in the suggestion generation stage, learning opportunities are created for each organization. With this approach it is aimed to help business process management users to focus on the potentially important parts of their business maps. Proposed methodology is implemented in ProM framework [29] as a set of plug-ins corresponding for each stage and packaged under the name of *CrossOrgProcMin* and tested on a synthetic and real-life event logs. Performance of methodology is assessed with a set of defined evaluation metrics for each stage and resulting recommendations are presented to show how this approach helps users to focus on learning opportunities between organizations with a performance improvement potential.

The rest of the paper is organized as follows: In Section 2, related studies in process mining area are presented. In Section 3, background information for the relevant topics is explained. In Section 4, methodology proposed in this study is presented with detail. In Section 5, methodology of this study is applied on datasets and results are discussed. In Section 6, summary of this study is presented with the final remarks and pointers for future work.

2 Related Work

In this section, studies related to the presented work are summarized. Firstly, studies in the process mining area are explained and then studies from cross-organizational process mining, which is the main topic of this research, are introduced. Following these, studies related to similarity in process mining are presented.

Within the process mining framework, there are various different process mining algorithms proposed which have the same aim of discovering underlying processes. Considering the underlying approaches, algorithms can be grouped as α-algorithms [7,26], inductive approaches [22,21], hierarchical clustering [19], genetic approaches [6,17], and heuristic approaches [18]. Considering the scope of this study; process discovery operations are undertaken with inductive methods which is a robust, repeatable and mature set of approaches.

Cross-organizational mining is based on cross-correlation of workflows and the realized activities in different organization to compare in an objective approach. In the study of Buijs et al. [11], process models and behaviors of organizations are cross-compared with the idea of supporting each other and representing differences. In the studies of van der Aalst [1,2], configurable process models are proposed with the ideas of *exploiting commonality* and *collaboration* for the organizations sharing the same infrastructure and doing the similar work. In this study, usage of cross-organizational process mining is based on *exploiting commonality* where organizations can learn from each other.

Similarity in process mining have various approaches which focus on metrics [14], analytical comparison [12,31], ontology analysis [27], delta analysis [16,15,20] and mismatch patterns [13]. In this research, combination of metric and mismatch pattern approaches are used to identify variations between process models of different organizations that execute the same tasks.

3 Background

In this section, process discovery methods and mismatch patterns are presented within the scope of this work. In the process mining field, various process discovery algorithms are proposed to address different challenges in process discovery and using different notations. In this study, since the focus is learning lessons from cross-organizational mining, we used Inductive Process Mining [23] for process discovery, which is simple, highly applicable and configurable. In the literature, its derivatives which handles infrequent behaviors [24]; incomplete logs [25]; and model optimization [30] are also available. *Inductive Miner Infrequent (IMi)* [24] extension is used in this study which is capable of filtering the infrequent behavior and results with lower fitness, higher precision and equal generalization.

In cross-organizational process mining environment, there is a need to align processes of different organizations. In the study of Dijkman [13], a collection of patterns to describe frequent mismatches between the similar process models are presented. Within the scope of this study, the related mismatch patterns are defined in study [13] as follows:

Skipped Activity An activity exists in one process but no equivalent activity is found in the other process.

Refined Activity An activity exists in one process but, as an equivalent, a collection of activities are existing in the other process to achieve the same task.

Activities at Different Moments in Processes Set of activities are undertaken with different orders in different processes.

Different Conditions for Occurrence Set of dependencies are same for two processes; however, occurrence condition is different.

Different Dependencies Dependency set of activities differ in different organizations.

Additional Dependencies This pattern is a special case of different dependencies where one set of activities includes the other and results with additional dependencies.

As mentioned in the study [13], their approach does not create a comprehensive list to resolve all mismatches but includes the most common mismatch patterns spotted during case studies. In addition, from their definitions and examples it can be easily seen that these patterns are not orthogonal. Moreover, there are no algorithms provided to spot these mismatches in [13] or consequent studies, and thus implementation of spotting mismatch patterns are performed within the scope of this study.

4 Methodology

In this section, the methodology proposed in this study is presented. Firstly, approach overview is described from a high-level perspective. Then, each stage in the methodology is presented together with their importance in the study, mathematical representations and definitions; and black-box diagrams. Finally, implementation details of this methodology in ProM framework is explained in detail with a software architecture overview.

4.1 Approach Overview

The approach proposed in this study consists of four main stages visualized in Figure 1. Firstly, in *Process Model Mining*, process models are extracted from event logs for each organization with a user specified noise threshold. Secondly, in *Performance Indicator Analysis*, event logs are replayed on process models and performance indicators are calculated for each organization then using these indicators, organizations are clustered based on how well they are operating. Thirdly, in *Mismatch Pattern Analysis*, differences between process models of organizations are extracted with well-established mismatch patterns. Finally, in *Recommendation Generation*, using the performance indicator clusterings and differences between process models; a set of recommendations for each organization is generated.

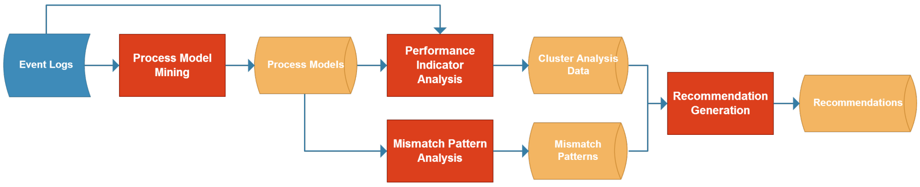


Figure 1: Overview of Methodology

4.2 Process Model Mining

Process model mining in the proposed approach has the aim of creating reproducible and generalized process models from event logs. Considering the fact that the process models may not be defined beforehand or outdated to reflect latest state of the process, they are mined from event logs. However, if there are process models that represent the event logs, this stage can be skipped. In order to mine process models, implementation of the *Inductive Miner Infrequent (IMi)*, which is proposed in [24] as an extension to *Inductive Miner* to handle noise in the event logs, is used in this study. In order to set a filtering threshold, a user-provided value between 0 to 1 is added as input to the method in addition to event logs.

4.3 Performance Indicator Analysis

Performance indicator analysis stage focuses on calculating and analyzing the performance values using the event logs and mined process models. This stage consists of mainly two steps as *a*) alignment and calculation of performance indicators; and *b*) clustering of organizations based on their performance values. In order to evaluate the performance of an organization based on their process models and past activities; there is a number of indicators in time dimension, cost dimension and utilization [3]. However, in this study, process related performance values are considered since differences in the process models are studied in the next stages. To this aim, the following performance indicators are calculated:

Average Time Between Activities This is a simple but powerful performance metric for organizations since it can yield the average time to complete one task based on a starting point. From the performance perspective, organizations want to minimize average time between activities to increase their throughput [4]. This notion can be defined as follows:

Definition 1. *Average time between activity A and B in organization i is* $AvgTime^{i}_{A\rightarrow B} = \frac{\sum_{Case\ c \in EventLog_i} TimeBetween_c(A,B)}{|Occurences_{Event\ Log_i}(A,B)|}$ *where*

1. $TimeBetween_c(A, B) = EndTime_c(B) - StartTime_c(A)$
2. $StartTime_c(A)$ *is start time of activity A in case c,*
3. $EndTime_c(B)$ *is end time of activity B in case c,*
4. $|Occurences_{EventLog_i}(A, B)|$ *is number of occurrences of activity A followed by B in Event* Log_i.

Standard Deviation of Time Between Activities Time between activities in real life is not stable and they deviate due to various reasons such as the user responsible of tasks, size and the content of tasks or seasonality [3]. On the other hand, organizations want to be confident about their processes and therefore they want to minimize the deviation in the time between activities. Minimized deviation in time helps organizations to plan, act and re-organize the activities in the processes with high accuracy [4]. With the same approach above, the following formulation can be defined:

Definition 2. *Standard deviation time between activity A and B in organization i is* $StdDevTime^{i}_{A\rightarrow B} = \sqrt{\frac{\sum_{Case\ c \in EventLog_i} [TimeBetween_c(A,B) - AvgTime^{i}_{A\rightarrow B}]^2}{Occurences_{Event\ Log_i}(A,B)}}$

Replay and Performance Indicator Calculation Replay of event logs on process models is based on the idea of *alignment* which is formalized in [4] and the basic assumption in this concept is that process models and event logs have the same activity labels. For each organization, the steps of alignment and creating transitions are performed with the corresponding event logs and process models; and the resulting process performance summaries are used for further analysis. Resulting data can be defined as follows:

Definition 3. *Performance Summary data for any organization i is $PerfSum_i = \{AvgTimeSum_i \cup StdDevTimeSum_i\}$ where*

1. $AvgTimeSum_i = \{AvgTime^i_{A \to B} | A, B \in Event\ Log_i\}$
2. $StdDevTimeSum_i = \{StdDevTime^i_{A \to B} | A, B \in Event\ Log_i\}$

Performance Indicator Clustering Clustering is based on the idea of collecting the set of observations into clusters so that observations within the same cluster are similar whereas the observations from different clusters are dissimilar. In this study, clustering is used to gather organizations based on their performance indicator data. In this research, random initialization based *k-means++* approach from the study of Arthur and Vassilvitskii [8] is used to cluster organizations. Since the number of clusters are not known priori, k-means clustering is applied starting k from 1 to the number of organizations. For each clustering with different number of clusters, *Sum of Squared Error (SSE)* values are plotted and user is asked to select the appropriate cluster size. For the selected cluster size, clustering related information is used to generate recommendations in the further steps. Resulting cluster analysis data is formulated as follows:

Definition 4. *Cluster Analysis Data is a tuple $(k, Assignments, Cluster\ Centroids)$ where*

1. *k is the number of clusters,*
2. *Assignments is a set of tuple $(Organization_i, Cluster_j)$ where i is identifier for organization and $j \leq k$ is identifier for cluster,*
3. *Cluster Centroids is a set of tuple $(Cluster_j, Type, A_{start}, A_{end}, Value)$ where*
   - *(a) Type is performance indicator type which is Average or StandardDev,*
   - *(b) A_{start} and A_{end} are starting and ending points of performance indicator,*
   - *(c) $Value$ is the actual value of performance indicator,*
   - *(d) Cluster $Centroids_j$ is a function that returns a set of Centroid which is a tuple $(Type, A_{start}, A_{end}, Value)$ for $Cluster_j$.*

4.4 Mismatch Pattern Analysis

In order to learn from other organizations, it is necessary to spot the differences between process models of different organizations. In this phase, differences between process models will be revealed by the mismatch patterns which are defined by Dijkman [13]. Since performance indicators are calculated based on a starting and ending point in the process model, the same approach is applied to locate mismatch patterns. In other words, differences of process models are located through a starting activity to an ending activity. With this aim, each mismatch pattern and its analyzers are defined by extending the following definitions. For each organization, mismatch pattern analyzers are pipelined and mismatch patterns are stored for further analysis.

Definition 5. *Mismatch Pattern is a tuple* $Mismatch\,Patten = (O_1, O_2, ExtensionData, A_{start}, A_{end})$ *where*

1. O_1 *is the first organization and*
2. O_2 *is the second organization in between the pattern occurs,*
3. *ExtensionData is a set of tuples where mismatch related information is recorded,*
4. A_{start} *and* A_{end} *are starting and ending points to check mismatch patterns.*

Definition 6. *Mismatch Pattern Analyzer is a function* $MismatchPatternAnalyzer(O_1, O_2, A_{start}, A_{end})$ *and it returns a set of Mismatch Pattern for the organization* O_1 *compared to* O_2 *for the activities between* A_{start} *and* A_{end}*.*

4.5 Genarating Suggestions/Recommendations for Performance Improvement

Recommendation generation stage in the methodology is the final and core stage where all information retrieved from the event logs until now are utilized. In this study, idea of recommendation is based on providing a set of mismatch patterns for each organization so that they can enhance their processes. These mismatch patterns are generated by comparing the process models of other organizations, particularly those that are performing better in terms of their performance indicator values. Recommendation idea and recommendation generation function is defined as following:

Definition 7. *Recommendation is a tuple* $Recommendation = (O, A_{start}, A_{end}, Mismatch\ Patterns)$ *where*

1. O *is identifier for organization,*
2. A_{start} *and* A_{end} *are starting and ending activities in between the recommendations are checked,*
3. $Mismatch\ Patterns$ *is collection of mismatch patterns.*

Definition 8. *Recommendation generation is a function that is* $RecGen(O, C, P)$ *and it returns a set of Recommendation where*

1. O *is identifier for organization,*
2. C *is Cluster Analysis Data which is result of cluster analysis stage,*
3. P *is Performance Threshold which is a real number larger than or equal to 0 and it is calculated over the same type of performance indicators of different organizations in Cluster Analysis Data.*

Algorithm of recommendation generation function is based on the idea of checking other clusters for a significant change in performance indicators, where significance is defined by the threshold provided by user. Only mismatches which are located between the activities that causes high level of difference in performance indicators are analyzed. This approach is formalized in Algorithm 1.

Algorithm 1: Recommendation Generation

```
Input: $O$ organization, $C$ Cluster Analysis Data, $P$ performance difference threshold
Output: $Recommendations$ a set of recommendations
$Recommendations \leftarrow \{\}$
$i \leftarrow C(Assignments(O))$
for $Centroid \in C(ClusterCentroids_i)$ do
    for $Centroid' \in C(ClusterCentroids_j)\ i \neq j$ do
        if $Centroid(A_{start}) = Centroid'(A_{start})$ & $Centroid(A_{end}) = Centroid'(A_{end})$ then
            if $(|Centroid(Value) - Centroid'(Value)| \div Centroid(Value)) \geq P$ then
                $A_{start} \leftarrow Centroid(A_{start})$
                $A_{end} \leftarrow Centroid(A_{end})$
                $MismatchPatterns \leftarrow \{\}$
                for $O' \in C(Assignments(j))$ do
                    $MismatchPatterns \leftarrow MismatchPatternAnalysis($O,O',A$_{start}$,$A_{end}$)
                $Recommendations \leftarrow Recommendation($O,A$_{start}$,$A_{end}$, $MismatchPatterns$)
return $Recommendations$
```

4.6 Implementation in ProM Framework

Methodology of this study is implemented in ProM [29], which is an extensible framework that supports a wide variety of process mining techniques in form of plugins. Approach of this study is implemented with its each stage as a standalone plugin that enables extensions for further studies. Developed set of plugins are packaged with the name of *CrossOrgProcMin*[1] and published open-source[2] being available in the latest version of ProM release.

[1] http://www.promtools.org/prom6/packages/CrossOrgProcMin

[2] http://github.com/onuryilmaz/cross-org-proc-min

5 Experimental Analysis Results and Discussions

In this section, methodology presented in this study is applied on several data sets and results are presented. Firstly, evaluation metrics are defined for each stage of methodology to assess the performance of approach. Following this, methodology is applied on two data sets and results are explained with discussions.

Approach in this study is an aggregation of various methods and they are significantly different from each other in their mathematical background. Therefore, instead of a global evaluation metric for the complete methodology, each stage is evaluated within its evaluation metrics. In *process model mining*, performance of process mining stage is measured by *fitness* and *appropriateness* which are defined in [28]. In *performance indicator analysis*, *alignment costs* [4] are compared with process model mining metrics for replay phase. In clustering phase, *within-SSE* analysis is undertaken to decide on the number of clusters. For *mismatch pattern analysis*, number of mismatch patterns found are compared with the *graph-edit similarity* [14] of process models. In *recommendation generation*, different threshold values are tried to check how many mismatch patterns are generated for organizations and how they could be used for focused analysis.

5.1 Loan Application Process

Loan Application Process dataset is synthetically created and consists of four variants of a simple loan application in a financial institute. These event logs are used for testing different approaches of discovering a configurable process model from a collection of event logs [10]. In this dataset there are a total of 475 cases and 2440 events with a fairly even distribution between variants and these variants are used as organizational logs and the methodology presented in this study is be applied.

In *Process Model Mining* stage, process models resulted with perfect fitness and high appropriateness as it is expected from a synthetically generated dataset without noise. In *Performance Indicator Analysis* stage, firstly event logs are replayed over process models and performance indicators are calculated and then organizations are clustered based on their performance indicators. In order to avoid overfitting, with two clusters, Variant #1, #2, and #4 are grouped into one cluster where only Variant #3 is left to other cluster. In *Mismatch Pattern Analysis* stage, number of mismatch patterns are analyzed with the *graph-edit similarity* between each two organization. As the similarity between process models decreases our method spots more mismatch patterns and it ensures that the developed mismatch pattern analyzers work as expected for this dataset. In *Recommendation Generation* stage, for different threshold values, number of performance indicators that are performing better for the selected organization and spotted mismatch patterns are plotted in Figure 2. In order to construct the data in Figure 2, every organization is selected one-by-one with different threshold values. For each analysis, number of performance indicators and average number of mismatch patterns causing them are plotted. In addition, total number of mismatch patterns without clustering is added as an upper bound. With the help

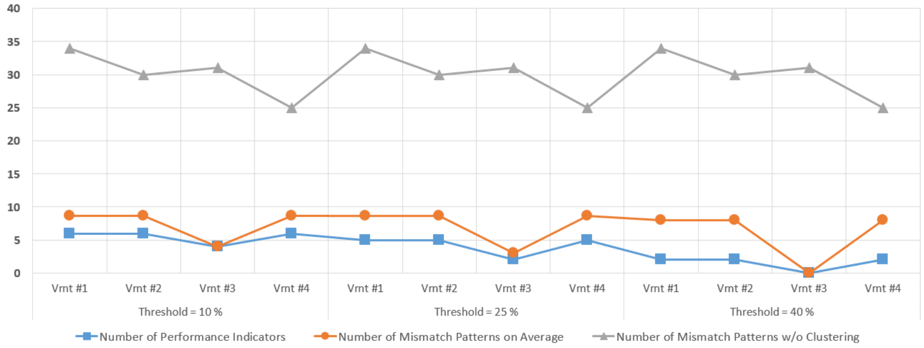


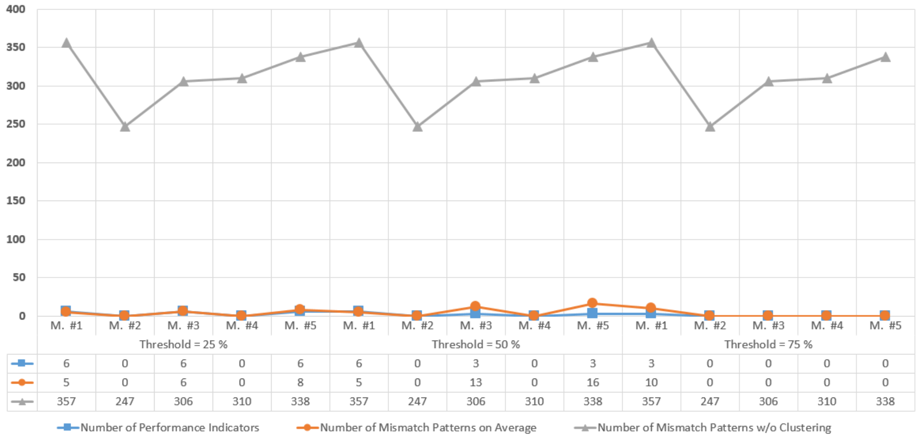
Figure 2: Recommendation Generation analysis for Loan Application Process dataset

of this upper bound, responsiveness and degree of helping the user to focus on the performance improvement can be analyzed. As can be seen, for each threshold value, average number of mismatch patterns *with performance indicator clustering* are very low compared to *without clustering*. In other words, when user wants to improve its performance with any threshold, there is significantly less number of mismatch patterns on average to check. This shows the methodology proposed in this study can help users to focus on differences between organizations given this dataset.

5.2 Environmental Permit Application Process

Environmental Permit Application Process dataset originates from the "Configurable Services for Local Governments (CoSeLoG)" project [1] which investigates the similarities and dissimilarities between several processes of different municipalities in Netherlands. Dataset contains records of receiving phase for the building permit application process in 5 municipalities, which are comparable since activity labels in the different event logs refer to the same activities performed in five municipalities. In this dataset [9], there are 1214 cases and 2142 events with a variable distribution between event logs of municipalities and municipalities are used as organizational logs.

In *Process Model Mining* stage, with 10 % of noise threshold, high fitness values are achieved; however, some of the process models like Municipality #4 and #5 resulted with low appropriateness values. In *Performance Indicator Analysis* stage, after calculating the performance indicators, municipalities are clustered and three clusters are created: Municipality #1 is located in the first cluster; Municipality #2 and #4 are located in the second cluster; and Municipality #3 and #5 are grouped in to the last cluster. In *Mismatch Pattern Analysis* stage, it can be stated that as the similarity between process models of municipalities increases, number of mismatch patterns decreases for most of the cases. When further analyzed, it can be seen that Municipalities #4 and #5, which have significantly more complex process models compared to others, fail in spotting mismatch patterns under *graph-edit similarity*. In *Recommendation Generation* stage, for different threshold values, number of performance indicators that are performing better for the selected organization and spotted mismatch patterns are plotted in Figure 3 for the thresholds of 25 %, 50 % and 75 % since these are the breaking points. For instance, cluster of Municipality #1 performs worse in 6 indicators with the difference of 25 % and on average 5 mismatch patterns are listed for each performance indicator. When it is compared to the total mismatch patterns of Municipality #1, which is 357, proposed approach helps significantly to the user for focusing performance improvement.



	Threshold = 25 %					Threshold = 50 %					Threshold = 75 %				
	M. #1	M. #2	M. #3	M. #4	M. #5	M. #1	M. #2	M. #3	M. #4	M. #5	M. #1	M. #2	M. #3	M. #4	M. #5
■	6	0	6	0	6	6	0	3	0	3	3	0	0	0	0
●	5	0	6	0	8	5	0	13	0	16	10	0	0	0	0
▲	357	247	306	310	338	357	247	306	310	338	357	247	306	310	338

Figure 3: Recommendation Generation analysis for Environmental Permit Application Process dataset (3 Clusters)

5.3 Discussions

When the evaluation of the stages for *Loan Application Process* and *Environmental Permit Application Process* datasets are gathered together, the following results can be expressed:

- Process mining stage of the proposed methodology can mine the process models with high fitness appropriateness levels.
- For the successfully mined models with high fitness values, replay and performance indicator calculation stage works seamlessly as expected. With this step, average and standard deviation time between each activity can be measured for each organization. Number of these metrics are quadratic to the number of activities in each organization's process model and difficult to analyze with a cross comparison.

– Internal measure of clusters indicates that the organizations can be clustered according to their performance indicators which yields a collective approach of organizations for their subprocesses. In other words, organizations are divided into clusters which shows that they can be grouped based on how well they are executing.
– Mismatch analysis spots the differences between process models in coherence with structural similarity of them. This indicates that the idea of using mismatch patterns to reveal differences between process models is a feasible approach since its results are comparable to the similarity metrics of process models in the literature.
– Recommendation generation aims to gather all generated information in this study to help focusing on the potentially important mismatch patterns for performance improvement. When the number of mismatch patterns with and without performance clusterings are checked, it shows that in a small dataset performance clustering lists 3 times less number of differences in *Loan Application Example* dataset. When it is impossible to locate mismatch patterns manually like in *Environmental Permit Application Process*, performance clustering spots 100 times less number of differences. This difference helps user to focus on the differences with a potential performance improvement which is one of the aims in this study.
– Although each step of methodology can be counted as successful based on their evaluation metrics, mismatch patterns recommended at the end of methodology can yield important observations as well as being irrelevant and infeasible. Since this decision is based on the business environment of organizations, evaluation of the quality of recommendations for business usefulness requires domain expertise. However, an example recommendation can be presented to provide an insight. In the analysis of *Loan Application Process*, Variant #3 performs worse 27 % on average time and 12 % on standard deviation time between activities "Calculate Capacity" and "Accept". When the mismatch patterns for these performance indicators are checked the following ones can be mentioned:
   - "Check Credit" is a *Refined Activity* of with "Check System (50 %)"; "Check Paper Archive (42 %)"; "Send Credit Check Request (32 %)"; "Process Credit Check Reply (31 %)" where the corresponding similarity values provided in parentheses.
   - "Calculate Capacity" is a *Different Moments in Processes* which have different previous activities in clusters.

   When these example mismatch patterns are checked, removing "Check Credit" activity and putting other activities instead of it might be the cause of performance improvement. With the same approach, putting "Calculate Capacity" on different orders in processes can effect the average and variance of time between activities. These mismatch patterns are also visualized on process model of Variant #3 and a variant from other cluster in Figure 4. In the process models, refined activities of "Check Credit" and different positions of "Calculate Capacity" are indicated.

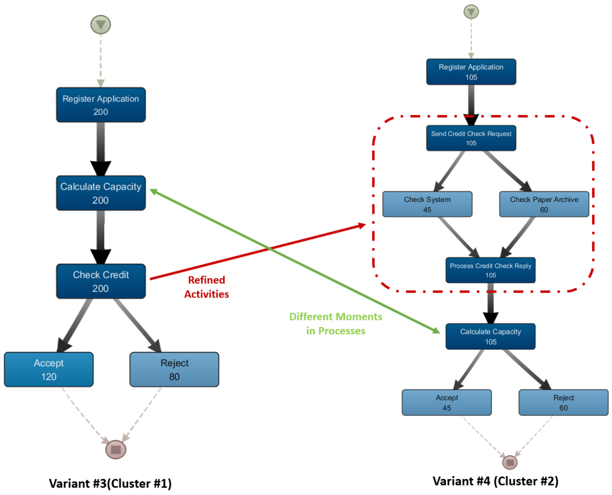


Figure 4: Visualization of example recommendation for Loan Application Process dataset

6 Conclusion and Future Work

In this study, a new approach is proposed and tested for generating recommendations using cross-organizational process mining for process performance improvement. Cross-organizational process mining is applied with the idea of unsupervised learning where predictor variables related to performances of organizations are used in an environment where processes are executed on several organizations. Results show that it is possible to use cross-organizational process mining and mismatch patterns for performance improvement recommendations. In this study, proposed methodology is developed as extensible and configurable set of plugins in ProM framework [29] and published as open-source. This makes the methodology open to include new process mining methods, mismatch patterns and clustering approaches as well as testing with different datasets.

For the approach proposed in this study, the following issues can be listed as pointers to future work:

- In the process mining stage, instead of *Inductive Miner*, new techniques can be used which can mine complex process models with higher appropriateness levels while keeping the current high fitness values.
- In the performance indicator analysis stage, new indicators can be defined based on the business environment, event log attributes and user needs. For instance, personnel and resource allocation indicators can be included as well as cost dimension.
- For mismatch pattern analysis, new and business oriented mismatch patterns can be included in the analysis. In addition analyzers can fail when there are loops in the process models in current implementations, therefore more

robust implementations for process models with loops can be developed in the future.

- For the generated recommendations, quality for business environment is not assessed within the scope of this study. However, when any feedback from a domain expert or BPM people is provided, the learning approach can be converted to semi-supervised learning from unsupervised learning.

References

1. van der Aalst, W.M.P.: Business process configuration in the cloud: how to support and analyze multi-tenant processes? In: Web Services (ECOWS), 2011 Ninth IEEE European Conference on. pp. 3–10. IEEE (2011)
2. van der Aalst, W.M.P.: Intra-and inter-organizational process mining: Discovering processes within and between organizations. In: The Practice of Enterprise Modeling, pp. 1–11. Springer (2011)
3. van der Aalst, W.M.P.: Process mining: discovery, conformance and enhancement of business processes. Springer Science & Business Media (2011)
4. van der Aalst, W.M.P., Adriansyah, A., van Dongen, B.: Replaying history on process models for conformance checking and performance analysis. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 2(2), 182–192 (2012)
5. van der Aalst, W.M.P., Adriansyah, A., de Medeiros, A., et al.: Process mining manifesto. In: Business process management workshops. pp. 169–194. Springer (2012)
6. van der Aalst, W.M.P., de Medeiros, A., Weijters, A.J.M.M.: Genetic process mining. In: Applications and theory of Petri nets 2005, pp. 48–69. Springer (2005)
7. van der Aalst, W.M.P., Weijters, T., Maruster, L.: Workflow mining: Discovering process models from event logs. Knowledge and Data Engineering, IEEE Transactions on 16(9), 1128–1142 (2004)
8. Arthur, D., Vassilvitskii, S.: k-means++: the advantages of carefull seeding. In: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. pp. 1027–1035 (2007)
9. Buijs, J.C.A.M.: Environmental permit application process ('wabo'), coselog project (2014), http://dx.doi.org/10.4121/uuid:26aba40d-8b2d-435b-b5af-6d4bfbd7a270
10. Buijs, J.C.A.M.: Flexible Evolutionary Algorithms for Mining Structured Process Models. Ph.D. thesis, PhD thesis. Eindhoven, The Netherlands: Technische Universiteit Eindhoven, 2014 (cit. on p. 179) (2014)
11. Buijs, J.C.A.M., van Dongen, B.F., van der Aalst, W.M.P.: Towards cross-organizational process mining in collections of process models and their executions. In: Business Process Management Workshops. pp. 2–13. Springer (2012)
12. Buijs, J.C.A.M., Reijers, H.A.: Comparing business process variants using models and event logs. In: Enterprise, Business-Process and Information Systems Modeling, pp. 154–168. Springer (2014)
13. Dijkman, R.: Mismatch Patterns in Similar Business Processes. Beta, Research School for Operations Management and Logistics (2007)
14. Dijkman, R., Dumas, M., van Dongen, B., Käärik, R., Mendling, J.: Similarity of business process models: Metrics and evaluation. Information Systems 36(2), 498–516 (2011)

15. Esgin, E., Karagoz, P.: Sequence alignment adaptation for process diagnostics and delta analysis. In: Hybrid Artificial Intelligent Systems, pp. 191–201. Springer (2013)
16. Esgin, E., Senkul, P.: Delta analysis: a hybrid quantitative approach for measuring discrepancies between business process models. In: Hybrid Artificial Intelligent Systems, pp. 296–304. Springer (2011)
17. Esgin, E., Senkul, P., Cimenbicer, C.: A hybrid approach for process mining: using from-to chart arranged by genetic algorithms. In: Hybrid Artificial Intelligence Systems, pp. 178–186. Springer (2010)
18. Esgin, E., Senkul, R.: A hybrid approach to process mining: Finding immediate successors of a process by using from-to chart. In: Machine Learning and Applications, 2009. ICMLA'09. International Conference on. pp. 664–668. IEEE (2009)
19. Greco, G., Guzzo, A., Pontieri, L.: Mining hierarchies of models: From abstract views to concrete specifications. In: Business Process Management, pp. 32–47. Springer (2005)
20. Hallerbach, A., Bauer, T., Reichert, M.: Capturing variability in business process models: the provop approach. Journal of Software Maintenance and Evolution: Research and Practice 22(6-7), 519–546 (2010), http://dx.doi.org/10.1002/smr.491
21. Herbst, J.: Dealing with concurrency in workflow induction. In: European Concurrent Engineering Conference. SCS Europe. Citeseer (2000)
22. Herbst, J., Karagiannis, D.: Integrating machine learning and workflow management to support acquisition and adaptation of workflow models. In: Database and Expert Systems Applications, 1998. Proceedings. Ninth International Workshop on. pp. 745–752. IEEE (1998)
23. Leemans, S.J.J., Fahland, D., van der Aalst, W.M.P.: Discovering block-structured process models from event logs-a constructive approach. In: Application and Theory of Petri Nets and Concurrency, pp. 311–329. Springer (2013)
24. Leemans, S.J.J., Fahland, D., van der Aalst, W.M.P.: Discovering block-structured process models from event logs containing infrequent behaviour. In: Business Process Management Workshops. pp. 66–78. Springer (2014)
25. Leemans, S.J.J., Fahland, D., van der Aalst, W.M.P.: Discovering block-structured process models from incomplete event logs. In: Application and Theory of Petri Nets and Concurrency, pp. 91–110. Springer (2014)
26. de Medeiros, A.K.A., van Dongen, B.F., van der Aalst, W.M.P., Weijters, A.J.M.M.: Process mining: Extending the α-algorithm to mine short loops (2004)
27. Pascalau, E., Rath, C.: Managing business process variants at ebay. In: Mendling, J., Weidlich, M., Weske, M. (eds.) Business Process Modeling Notation, Lecture Notes in Business Information Processing, vol. 67, pp. 91–105. Springer Berlin Heidelberg (2010), http://dx.doi.org/10.1007/978-3-642-16298-5_9
28. Rozinat, A., van der Aalst, W.M.P.: Conformance checking of processes based on monitoring real behavior. Information Systems 33(1), 64–95 (2008)
29. Verbeek, H.M.W., Buijs, J.C.A.M., van Dongen, B.F., van der Aalst, W.M.P.: Prom 6: The process mining toolkit. Proc. of BPM Demonstration Track 615, 34–39 (2010)
30. Weidlich, M., van der Werf, J.M.: On profiles and footprints–relational semantics for petri nets. In: Application and Theory of Petri Nets, pp. 148–167. Springer (2012)
31. Weidlich, M., Mendling, J., Weske, M.: A foundational approach for managing process variability. In: Mouratidis, H., Rolland, C. (eds.) Advanced Information Systems Engineering, Lecture Notes in Computer Science, vol. 6741, pp. 267–282. Springer Berlin Heidelberg (2011), http://dx.doi.org/10.1007/978-3-642-21640-4_21