Ferret Miner: A Process Mining Case Study

Martin Alvarez-Lopez
MS in Software Engineering
San Jose State University
San Jose, United States
martin.alvarezlopez@sjsu.edu

Hardy Leung
MS in Artificial Intelligence
San Jose State University
San Jose, United States
kwok-shing.leung@sjsu.edu

Carlos Hernandez
MS in Computer Engineering
San Jose State University
San Jose, United States
carlos.hernandez@sjsu.edu

Divyam Sobti
MS in Artificial Intelligence
San Jose State University
San Jose, United States
divyam.sobti@sjsu.edu

***Abstract*—The current project analyzed event data logs from the travel reimbursement process at Eindhoven University of Technology (ETU/c). The objective is to identify the bottlenecks in the process of travel declaration and improve its efficiency. Process Mining techniques such as a control-flow model and a value stream map are generated to help identify any delays in the process. Subsequently, bottlenecks were detected using a representative analysis. The implementation results recognized bottlenecks in critical tasks of the process that were optimized by assigning more resources to the area. Compared with the original, the optimized process improved its efficiency by almost 30%, significantly reducing the time processing the travel reimbursement.**

I. INTRODUCTION

Process mining consists of analysis techniques that reveal core information from processes through event data logs [1]. Error and deficiencies can be detected using this methodology. Process mining and data mining are two different but closely related data analytical systems since both belong to the business intelligence field. However, data mining concentrates on finding data relationships, whereas process mining also focuses on analyzing data from a systematic perspective.

Currently, information systems, such as Enterprise Resource Planning (ERP), Work on Management Systems (WMS), or Customer Relationship Management (CRM), are vital tools for the execution of almost any company process [2]. The essence of these information systems relies on their capabilities of logging events. An event log is a record of an activity performed with a Timestamp embedded [3]. Event logs facilitate process mining to discover information by tracking the chronology of the tasks and their execution time registered in each log.

A. Problem Description

This project performs a process mining analysis to detect patterns, delays, and bottlenecks obstructing the efficient flow of process activities. The main objective is to provide a case study assessment using a dataset with accurate information from the Eindhoven University of Technology. The research data is from the BPI Conference 2020 held in Padova, Italy[4].

The appropriate supervision of a business behavior enables the opportunity to adjust any task process generating a problem [5]. Therefore, our approach focuses on studying the problem using process mining methodology. *Process mining* is an outstanding data analysis technique that generates models representing the behavior indicated by activity logs produced by business applications after the execution of a process task.

Our approach proposes a real case study analysis using the process mining technique and utilizes a structured dataset based on systematic event logs. The system logs provide a detailed record of the process task, identification, resource, timestamp, and other data. These attributes provide insights into the successful execution of the process or any other problem encountered at any stage of the activities process flow.

Wil Van Der Aalst introduced the fundamentals of our approach in his paper titled "Process Mining" [6], where process mining guidelines are detailed. A

process mining implementation generally covers three main issues: 1) Discovery, 2) Conformance, and 3) Enhancement. Our project focuses on Discovery since it is the area that requires more effort and it is the most deterministic for the success of the analysis.

B. Motivation

Practical application publications are scarce, and our research paper aims to add support to this section. Furthermore, another motivation to complete this investigation relies on resolving a real problem using process mining techniques implemented with python PM4PY library modules. We found python PM4PY a stable open-source library available for everyone with outstanding capabilities to resolve our problem. The goal is to demonstrate the practicality of process mining analysis in practice. Our paper illustrates a case study in processing travel reimbursement for a university in Germany. The travel reimbursement issue involves several business processes for which analysis of event logs registration is worthwhile. The case being developed involves a number of students who submitted their applications, added their support documentation, and a series of tasks that need to be validated before the disbursement approval.

C. Approach Summary

The document is structured as follows: Section 1 provides details about process mining and its fundamentals; Section 2 presents an overview of our approach and related literature review; Section 3 analyzes the outcomes produced from our primary studies and research problem. Section 4 refers to the discussion. Finally, Section 5 concludes the work.

II. SURVEY

The literature review for process mining-related topics concentrates on developing innovative techniques and algorithms from a control-flow discovery perspective and its applications.

A. Process Mining

The position of process mining, considering other disciplines, is just between data mining and process modeling [7]. Such interaction is represented in figure one.

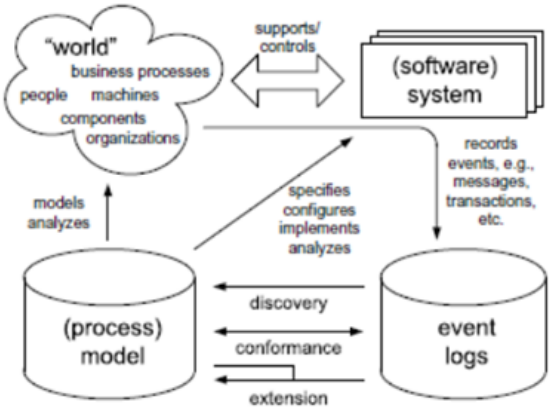

Figure 1 General outline of process mining approach

Figure 1 General outline of process mining approach Usually, information system databases do not have clearly defined structures, so a process mining analysis must search and separate the information directly from the system. Under these terms, process mining is considered a technique where the event logs possess the correct information to determine the system's behavior. Event logs, then, are guidelines that dictate the applicability of a process mining implementation [6]. As a result, the process mining approach embraces three main actions (Figure 2):

1) *Discovery*, which creates models by analyzing event logs

2) *Conformance checking*, which enables to compare of a newly created model versus a predefined model to gain knowledge

3) *Enhancement*, which deals with the implementation of solutions to overcome previously detected problems.

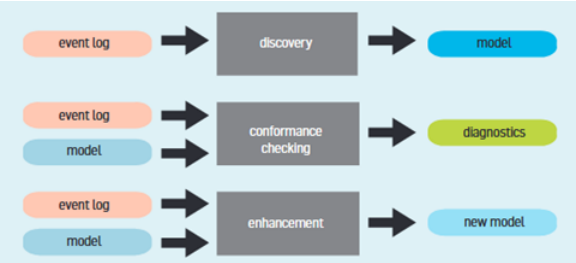

Figure 2 Process mining layout

B. Process Models

Distinct data analysis approaches have been developed to discover processes from event logs. However, the Petri Nets technique has taken relevance over the others [8]. Another approach utilizes Unified Modeling Language (UML) diagrams [9]. Furthermore, A novel design named "alpha algorithm" can depict event data logs into work-flow diagrams deriving from Petri nets. Nowadays, a popular model is the "Heuristic Miner." One main characteristic of this method is to be highly permissive to distinct statistical levels and thresholds adjusted by the user [10]. In order to deal with the massive number of events that existing information systems generate, Burattin et al. [11] developed a Heuristic Miner adaptation based on Lossy Counting and a sliding window that effectively analyzes streams of real datasets.

C. Applications

Most work-related publications are dedicated mainly to enhancing novel methodologies and algorithms with significant emphasis on the control-flow discovery area [7]. However, the public services area has been shared for process mining development research. For instance, a Genetic Miner approach was tested by Alves de Madeiros et al. [12] in Dutch municipalities. Additionally, van del Aalst et al. [13] applied an analysis of organizations in another municipality process. Rozinat et al. [14] evaluated process models in two case studies on the public sector.

Nonetheless, all these research applications focus on validating and assessing rather than enhancing the process. Implementation of process mining approaches has also been developed in the private domain. Distinct discovery approaches were deployed by Goedertier et al. [15] in the telecom field. Also, Mans et al. [16] and Refuge and Ferreira [17] created a convoluted process mining method to resolve the tracking of patients within the healthcare industry.

The literature mentioned above provides the fundamentals of process mining from a general perspective. Each approach contributes significantly to the process mining study field, and in this paper, we focus on its application.

III. TECHNICAL APPROACH

IV. EXPERIMENTAL METHODOLOGY

In this section we will be disscusiing about data, how we preprocessed the data, which model we used, how we reach the result and disscuss about the result.

A. Dataset Description

After clear understanding of the question we got to know that we had to find bottleneck in travel declaration. After reviewing the data provided to us we decided to go with the domestic and internation declaration. Both the dataset are in XML format. It contains the information regarding the process that is they contains event log. Domestic process is different from international process so the data in those file are also different. For domestic, a employee complete the trip then ask for reimbursement where as for international trip, an employee had to get primts from the supervisor and then the trip is considered to start.

Domestic declaration data set contains 10 features:-
1. id 2. org:resource 3. Concept:name 4. time:timestamp 5. org:role 6. case:id 7. case: concept:name 8. case:BudgetNumber 9. case: DeclarationNumber 10. case:Amount

The International declaration log contains 23 features:- 1. id 2. Org:resource 3. concept:name 4. time:timestamp 5. Org:role 6. case:Permit travel permit number 7. case:DeclarationNumber 8. case:Amount 9. case:RequestedAmount 10. case:Permit TaskNumber 11. case:Permit BudgetNumber 12. case:OriginalAmount 13. case:Permit ProjectNumber 14. Case:concept:name 15. case:Permit OrganizationalEntity 16. case:travel permit number 17. case:Permit RequestedBudget 18. case:id 19. case:Permit ID 20. case:Permit id 21. case:BudgetNumber 22. case:Permit ActivityNumber 23. case:AdjustedAmount

B. Preprocessing

To Pre-process on the data, first we analysis the data i.e how many null value the data contains, were there any missing values or duplicate values and checked for infrequent data. We found out there were no missing data but as we analysis the data we found out

that there were alots of duplicated data so we combine the it with fields similar to it so that we dont remove and necessary data. After combining we drop the whole columns. Some features had missing or unknow value which were also drop. But the main in preprocessing the international data was that the process was interjected with premit process which was effecting the process of we decided to remove them. At last to deal with infrequent data the regular approach is to drop those value with were certain threshold to transform data needed for models but it comes with the sideeffect tha it many effect the outcome so what we did is that we Filter and retaining top variants. In our case we kept the top 15-25% variants depending on data.

C. Models

To solve the process discovery for this data we had 2 options:- 1. Direct Flow/Follow Graph (DFG) 2. Perti nets. Both models display the timestamps on graph that is it display that mean time of how much each process toke to complete. Both models had some positive and negative points, DFG is simple and it is easy to understand where as perti net is complicated and hard to understand. The drawback for DFG is that it does cover the concurrent data that if the process can be complete A- B and A-C-B it will ethier show A-B or A-C-B not both where perti nets cover all of this. For DFG model you just provide input as event log and it will provide us with the best possible outcome, however for perti nets you have to create Inductive and heuristic perti nets. To find the bottleneck in process we used DFG models, we create 2 models one for frequency and other for performance for each data set.

D. Results

V. FUTURE WORK

A. Conformance Analysis

Once we have developed a process model either extracted through process discovery or built on some reference guidelines, it would be interesting to investigate further into conformance checking to see how well the data fit the model, which in turn would help us evaluate and improve on the model.

This can be done with token-based replay (TBR) on the Petri-net converted from the process model. For each trace, we would try to see if it can be executed on the Petri-net mathematically soundly, keeping track of additional tokens added to avoid deadlock and checking for unused tokens at completion. We define *total fitness* simply as the percentage of traces that fit the model. Another technique is alignment analysis, similar to TBR, except that events of a trace are aligned to a legal and model-fitting trace trajectory subject to certain cost functions. TBR and alignment analysis would identify non-conformance, but the latter is more informative, although it comes at the expense of runtime complexity. Behavior analysis entails the analysis of concepts present in the event log but either impossible or expensive to capture in the event formalism (e.g., the person who submits the request cannot be the same person who approves). These techniques can better identify the hows and whys of non-conformance and build better models.

Getting to 100% total fitness is not necessarily the desired goal. Instead, what we seek is a tradeoff. On the one hand, you want the model to be robust enough to elegantly explain the process. On the other hand, we do not want to over-build the model to make it so complex that it is hard to reason about. Let's use our specific problem as an example. We ran an experiment to examine such relationships. We ran the inductive miner on different values of k, where k is the number of top variants we used to build the model.

Getting to 100% total fitness is not necessarily the desired goal. Instead, what we seek is a tradeoff. On the one hand, you want the model to be robust enough to elegantly explain the process. On the other hand, we do not want to over-build the model to make it so complex that it is hard to reason about.

Let's use our specific problem as an example. We ran an experiment to examine such relationships. We ran the inductive miner on different values of k, where k is the number of top variants we used to build the model.

And we ran the conformance check to get the total fitness and wrote code to examine the model. In this figure, the red line is the total fitness, and the blue line is the model's size. We see that the model gets more complex as we include more traces, but the

fitness improves. If we would include only the top 3 variants, we can explain 90% of the traces. But if we blindly up the ante to 18 variants, we can now account for 99% of the traces, but the model is five times more significant and much harder to understand.

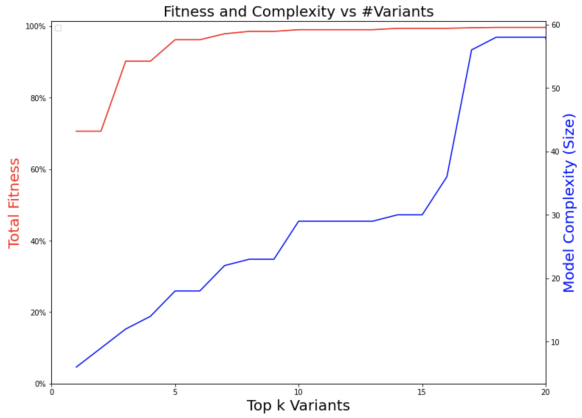

Figure 3 Fitness and Complexity vs Top k Variants

The sweat spots happen at k = 5, which can explain 96% of the traces. The heuristics map, shown in Figure 4, is relatively easy to understand. And it can adequately capture the submission, approval, rejection, resubmission, and final payment.

Figure 4 Heuristics Net when using the Top 5 Variants

We have only scratched the surface of conformance analysis, as much more can be done.

B. Machine Learning and Timely Analysis

Another area of interest is to apply machine learning techniques to perform (1) flow analysis to predict the likelihood of resubmission, rejection, payment delay, or over-budget based on other variables (time since submission, request amount, prior owner approval) and (2) timing analysis to determine mean-time-to-trace-completion. This provides essential clues to administrators and employees about the time frame of payment in the best-case scenario.

We are also interested in detecting non-compliance "as it happens" and sending alerts to administrators.

VI. CONCLUSION

REFERENCES

[1] "What is Process Mining? All You Need to Know | Quixy," Aug. 10, 2020. https://quixy.com/blog/all-about-process-mining/ (accessed Nov. 11, 2022).

[2] "tutorial:introduction | ProM Tools." https://www.promtools.org/doku.php?id=tutorial:introduction (accessed Nov. 11, 2022).

[3] "Event Log – The Process Mining Glossary | Appian." https://appian.com/process-mining/event-log.html, https://appian.com/process-mining/event-log.html (accessed Nov. 11, 2022).

[4] B. van Dongen, "BPI Challenge 2020," Mar. 2020, doi: 10.4121/UUID:52FB97D4-4588-43C9-9D04-3604D4613B51.

[5] F. Leymann and W. Altenhuber, "Managing business processes as an information resource," *IBM Syst. J.*, vol. 33, no. 2, pp. 326–348, 1994, doi: 10.1147/sj.332.0326.

[6] W. Van Der Aalst, "Process mining," *Commun. ACM*, vol. 55, no. 8, pp. 76–83, Aug. 2012, doi: 10.1145/2240236.2240257.

[7] W. van der Aalst, T. Weijters, and L. Maruster, "Workflow mining: discovering process models from event logs," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 9, pp. 1128–1142, Sep. 2004, doi: 10.1109/TKDE.2004.47.

[8] J. E. Cook and A. L. Wolf, "Discovering models of software processes from event-based data," *ACM Trans. Softw. Eng. Methodol.*, vol. 7, no. 3, pp. 215–249, Jul. 1998, doi: 10.1145/287000.287001.

[9] S. Saito, "Understanding Key Business Processes for Business Process Outsourcing Transition," in *2019 ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE)*, Montreal, QC, Canada,

May 2019, pp. 35–39. doi: 10.1109/ICGSE.2019.00021.
[10] A. J. M. M. Weijters, van der Aalst W. M. P., and A. K. Alves De Medeiros, *Process mining with the HeuristicsMiner algorithm*. Technische Universiteit Eindhoven, 2006.
[11] A. Burattin, A. Sperduti, and W. M. P. van der Aalst, "Control-flow discovery from event streams," in *2014 IEEE Congress on Evolutionary Computation (CEC)*, Beijing, China, Jul. 2014, pp. 2420–2427. doi: 10.1109/CEC.2014.6900341.
[12] A. K. A. de Medeiros, A. J. M. M. Weijters, and W. M. P. van der Aalst, "Genetic process mining: an experimental evaluation," *Data Min. Knowl. Discov.*, vol. 14, no. 2, pp. 245–304, Apr. 2007, doi: 10.1007/s10618-006-0061-7.
[13] A. Rozinat, I. S. M. de Jong, C. W. Gunther, and W. M. P. van der Aalst, "Process Mining Applied to the Test Process of Wafer Scanners in ASML," *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 39, no. 4, pp. 474–479, Jul. 2009, doi: 10.1109/TSMCC.2009.2014169.
[14] A. Rozinat, R. S. Mans, M. Song, and W. M. P. van der Aalst, "Discovering simulation models," *Inf. Syst.*, vol. 34, no. 3, pp. 305–327, May 2009, doi: 10.1016/j.is.2008.09.002.
[15] S. Goedertier, J. De Weerdt, D. Martens, J. Vanthienen, and B. Baesens, "Process discovery in event logs: An application in the telecom industry," *Appl. Soft Comput.*, vol. 11, no. 2, pp. 1697–1710, Mar. 2011, doi: 10.1016/j.asoc.2010.04.025.
[16] R. S. Mans, M. H. Schonenberg, M. Song, W. M. P. van der Aalst, and P. J. M. Bakker, "Application of Process Mining in Healthcare – A Case Study in a Dutch Hospital," in *Biomedical Engineering Systems and Technologies*, vol. 25, A. Fred, J. Filipe, and H. Gamboa, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 425–438. doi: 10.1007/978-3-540-92219-3_32.
[17] Á. Rebuge and D. R. Ferreira, "Business process analysis in healthcare environments: A methodology based on process mining," *Inf. Syst.*, vol. 37, no. 2, pp. 99–116, Apr. 2012, doi: 10.1016/j.is.2011.01.003.

TEAM CONTRIBUTIONS

We would like to thank Professor Mahima Agumbe Suresh for her teaching, encouragement, and devotion to helping us succeed. The breakdown of our team contributions is as follows:

Martin A.	Coding, mining algorithm, data preparation, writing, presentations
Carlos H.	Literature survey, report template and first draft author, presentations
Hardy L.	Coding, conformance research, report writing, presentations
Divyam S.	Coding, core algorithm, report writing, presentations