Process Simulation and Pattern Discovery through Alpha and Heuristic Algorithms

Wichian Premchaiswadi[1], Parham Porouhan[2]
Graduate School of Information Technology
Siam University
Bangkok, Thailand
wichian@siam.edu[1], parham@siam.edu[2]

Abstract—**The paper is divided into two main parts. In the first part of the study, we applied two process mining discovery techniques (i.e., alpha and heuristic algorithms) on an event log previously collected from an information system during an Academic Writing (English) training course at a private university in Thailand. The event log was initially consisted of 330 process instances (i.e., number of participants) and 3,326 events (i.e., number of actions/tasks) in total. Using alpha algorithm enabled us to reconstruct causality in form of a Petri-net graph/model. By using heuristic algorithm we could derive XOR and AND connectors in form of a C-net. The results showed 86.36% of the applicants/participants managed to achieve the Academic Writing (English) certificate successfully, while 6.36% of them failed to achieve any certificate after a maximum number of 3 attempts to repeat the training course. Surprisingly, 7.28% of the participants neither achieved an accredited certificate nor failed the course by dropping out before ending the course training process. In the second part of the study, we used performance analysis with Petri net technique (as a process mining conformance checking approach) in order to further analyze the points of non-compliant behavior (i.e., so-called bottlenecks or points of non-compliant behavior) for every case in the collected course training log. Based on the results, we could eventually detect the existing discrepancies of the event log leading to +24 missed tokens and -24 remained tokens altogether.**

Keywords—**Process Mining, Model Discovery, Alpha algorithm, Heuristic Miner algorithm, Process Simulation, ProM, Bottleneck Mining, Conformance Checker, Performance Analysis with Petri net, MXML.**

I. INTRODUCTION

ProM is a popular platform designed for implementing process mining tools in a standard environment. The ProM framework accepts and supports the input logs in forms of XES or MXML formats only. At the moment, this framework includes tools suitable for process mining process discovery, data analysis, workflow monitoring and conversion. In other words, the ProM environment has been developed based on an entirely adaptable and plug-able platform in such a way that it can be extended by more than 200 plug-ins in total [1], [2],[3].

Fig. 1 represents a holistic view of the way these plug-ins can be implemented and categorized. The plug-ins which only emphasize and focus on extracting knowledge and insight from an event log are so-called “discovery tools” as they do not apply (or compare) any prevailing information about pre-defined/master models. The plug-ins that compare and contrast the extent of compatibility and consistency (i.e., fitness or auditing) between an event log and a pre-defined model are called “conformance tools”. Lastly, the plug-ins that use both a pre-defined model and its relevant event logs to learn new information that can lead to enhancement of" the existing model are called “extension tools” [2],[3].

II. LITERATURE REVIEW

The idea of process simulation and workflow mining in business environments (based on the event logs collected from information systems) is not new [6],[7], [2],[3]. Cook and Wolf investigated the appliance of the process mining by focusing on the software engineering processes. In [6], [2],[3] they explained three methods for process discovery: one using neural networks (which later was elaborated to emergence of the social network miner graphs), one using a purely algorithmic approach (which later were elaborated into several process discovery algorithms based on the Petri-nets and BPMN models), and one Markovian approach (which later were elaborated in terms of the multiple clustering methods). The authors considered the latter two the most promising approaches. The Markovian approach used a mixture of algorithmic and statistical methods capable of dealing with noise. Note that the results presented in [6] were limited to sequential behavior. Cook and Wolf extended their work to concurrent processes in [10]. They proposed specific metrics (such as entropy, event type counts, periodicity, and causality) and used these metrics to discover models out of event streams. However, they did not provide an approach to generate explicit process models [2],[3]. In [7] Cook and Wolf provided a measure to quantify discrepancies between a process model and the

978-1-4673-9190-0/15/$31.00©2015 IEEE

actual behavior as registered using event-based data. The idea of applying process mining in the context of workflow management was first introduced in [8]. This work was based on the workflow graphs, which were previously inspired by work flow products such as IBM MQSeries workflow (formerly known as Flowmark) and InConcert [2],[3].

Consequently, in [9], a tool based on these algorithms was presented. Schimm [11], [12] developed a mining tool suitable for discovering hierarchically structured workflow processes in such a way that all splits and joins were appropriately balanced. Herbst and Karagiannis also addressed the issue of process mining with regard to the context of workflow management [13], [14] using an inductive approach. The work presented in [14], [15] was limited to sequential models. The approach described in [14] allowed for investigation of the concurrency cases by using stochastic task graphs as an intermediate representation which could generate a workflow model described in the ADONIS modeling language. In the induction step task nodes were merged and split in order to discover the underlying process. In [16], [17] a heuristic approach, using rather simple metrics, was used with the purpose of constructing the so-called the "dependency/frequency tables" as well as the "dependency/frequency graphs". The preliminary results presented in [16], [17] only provided heuristics methods and focused on more sophisticated issues/problems such as dealing with noise in more complex and complicated datasets. In [18] the EMiT tool was presented which applied an extended version of the α-algorithm in order to incorporate timing information. More information and detail about the α-algorithm were explained in [19], [2],[3].

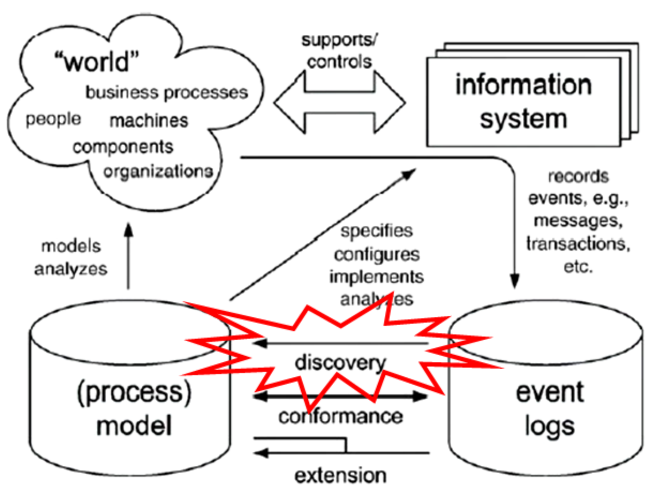

Figure 1. A holistic view of the process mining model as well as its methods, dimensions, and relationships (Sources: Process Mining official website and Process Mining: Discovery, Conformance and Enhancement of Business Processes) [4],[5],[26].

III. Case Study

In this study, we used an event log describing the students of a private university in Bangkok (Thailand) who voluntarily enrolled and undertake an optional *Academic Writing (English)* course. Those who managed to successfully pass the course were granted a certified certificate, while those who could not pass the course (after a maximum number of 3 attempts) failed the course without any certificate. In general, a total of 330 students registered and attended the *Academic Writing (English)* course during a period of time from 2 November 2555 (i.e., based on the Thai solar calendar it is equal with the year 2012) to 24 July 2557 (i.e., based on the Thai solar calendar it is equal with the year 2014).

The *Academic Writing (English)* course was initially started by enrolling the course by paying the required tuition. After completion of the enrollment process, the applicants needed to take a "Pre- Test" where the level of their knowledge and familiarity with the Academic Writing (English) topics was assessed, evaluated and categorized with respect to five different levels as the following: "Elementary level", "Pre-Intermediate level", "Intermediate level", "Upper-Intermediate level", and "Advanced level". Once the applicant was assessed and evaluated, he/she needed to attend the authentic classrooms so-called "Training Sessions". The training sessions also were divided into 2 main training programs (with different scopes of learning and duration of time to accomplish each program). An applicant whose level of knowledge (i.e., extent of familiarity with Academic Writing (English) topics) was equivalent with the "Elementary" or "Pre-Intermediate" levels, needed to undertake the "Long Training" program within 4 months. An applicant whose level of knowledge (and extent of familiarity with Academic Writing (English) topics) was equivalent with the "Upper-Intermediate" or "Advanced" levels, needed to undertake a "Short Training" program within only 6 weeks. Those applicants whose level of knowledge (and extent of familiarity with Academic Writing (English) topics) was equivalent with the "Intermediate" level, had the authority to choose any of the "Long Training" or "Short Training" programs after consulting with the qualified instructors and evaluators, or based on their willingness and readiness to choose and undertake any of the programs. Once an applicant accomplished the training sessions, he/she needed to take the "Pro-Test" in order to indentify the level of progress made compared with Pre-Test scores and results. If an applicant could pass the "Pro-Test" exam properly, then an accredited/certified/formal *Academic Writing (English) Certificate* would be granted. If an applicant could not pass the "Pro-Test" exam, then he or she should repeat the training sessions (i.e., whether short or long training program) after consulting with the qualified instructors and evaluators, or based on their willingness and readiness to choose and undertake any of the programs. Consequently, if an applicant could not pass the "Pro-Test" exam after a maximum number of 3 attempts, then he or she would fail the *Academic Writing (English)* training course without any certificate and would be able to re-register the training course after a passing a duration of 5 years, at least.

IV. Findings and Results

As described earlier in previous studies conducted by [2] and [3], there are many different algorithms for process mining which are capable of addressing problems (or case studies) in different ways, and therefore, applicable to versatile situations and scenarios. However, process mining algorithms are divided into two main approaches: (a) the algorithms that build (and generate) a model based on an event log (input-data) such as Alpha algorithm, Heuristics Miner, Fuzzy Miner and so on [1],[5]. (b) The algorithms that compare and contrast a pre-defined model with an event log (whether authentic or synthetic) with the purpose of identifying the level of compatibility and consistency (i.e., fitness) between the event log and the model such as LTL Checker approach, Performance Analysis plug-in and etc. In this study, since the original event log —collected from an information system in a private university in Thailand during an Academic Writing (English) training course—contained information about cases (i.e., applicant's/student's identification ID or registration number), originators (i.e., the staff and instructors who were in charge of the registration & payment, evaluation & assessment, and teaching & training the applicants/students), activities (i.e., the types of the actions occurred during the Academic Writing (English) course training, such as, registration & payment, initial evaluation & assessment, intensive course training, full-term course training, re-take course training, granting a certificate, or failure of the course), timestamps (i.e., the dates and times the applicants/students start or complete the assigned actions), and some additional information and demographic details (such as the applicants and number of their attempts to pass the course), we decided to apply those process mining algorithms that build (and generate) a model based on an event log (input-data) such as Alpha algorithm and Heuristics Miner (i.e., discovery methods) [26].

A. *Alpha (α) Algorithm*

The α-algorithm is a basic algorithm which is commonly used in process mining with the purpose of reconstructing causality from a set of sequences of events [1],[5]. Alpha-algorithm is illustrated in terms of Petri Nets (Place/Transition Nets). The algorithm was first put forward by "van der Aalst", a full professor at the Department of Mathematics & Computer Science of the Technische Universiteit Eindhoven (TU/e). In this paper, we used α-algorithm as a very basic process mining technique to identify the routing constructs within an event log —dealing with handling of an *Academic Writing (English)* training course — collected from an information system at a private university in Bangkok, Thailand [20], [21],[2],[3]. Due to the fact that our main emphasis was on the *Academic Writing (English)* training course as a whole, therefore, we based our analysis on the "started" and "completed" types of the process instances only, while ignoring the "in-progress" (or "pending") types of actions and activities.

Consequently, the resulting graph/model produced by Alpha algorithm —based on the *Academic Writing (English)* training course event log—is shown in Fig. 2. By contemplating on the resulting Alpha model, now we can better study the control-flow of the course training processes/tasks as well as the dependencies between amongst them. The Alpha model clearly indicates which tasks precede which other ones. However, in complete surprise, we observe that (based on the collected event log) currently the resulting Petri net ends with 5 activities/tasks (rather than only 2). Being aware of the fact the course training process eventually should end with "With Certificate (Complete)" and "Without Certificate (Complete)" (i.e., shown in Green Color Boxes and Asterisk Marks), we can see that the training process ends with 3 unwanted activities/tasks as follows: "Post-Test (Complete)", "Long-Training (Start)", and "Long-Training (Complete)" (i.e., shown in Red Color Boxes and Asterisk Marks).

B. *Heuristic Mining*

Due to the fact that the Alpha algorithm is not a robust approach to deal with the logs that contain concurrent activities and loop data [5],[2],[3], therefore, we used Heuristic Miner algorithm as a process discovery technique in order to consider the frequency of the process instances (despite of the Alpha algorithm) as well. The main objective was to produce models/graphs which are less sensitive to noise and the incompleteness of logs. Two screenshots of the resulting graph/model produced by Heuristic Miner algorithm —based on the *Academic Writing (English)* training course event log—are shown in Fig. 3. As we consider, the Heuristic Miner algorithm dealt with two fundamental metrics with respect to: (a) Significance and (b) Correlation. "Significance" dealt with the relative importance of behavior while "Correlation" dealt with how closely related two events following one another are [3],[27]. By contemplating on the resulting Heuristic model, we find out something abnormal. Although the total number of applicants taking the *Academic Writing (English)* training course was equal with 330 applicants/students, however, the total number of applicants "With Certificate" and "Without Certificate" currently is less than 330 case instances (i.e., 285 + 21 = 306). [25],[24],[26].

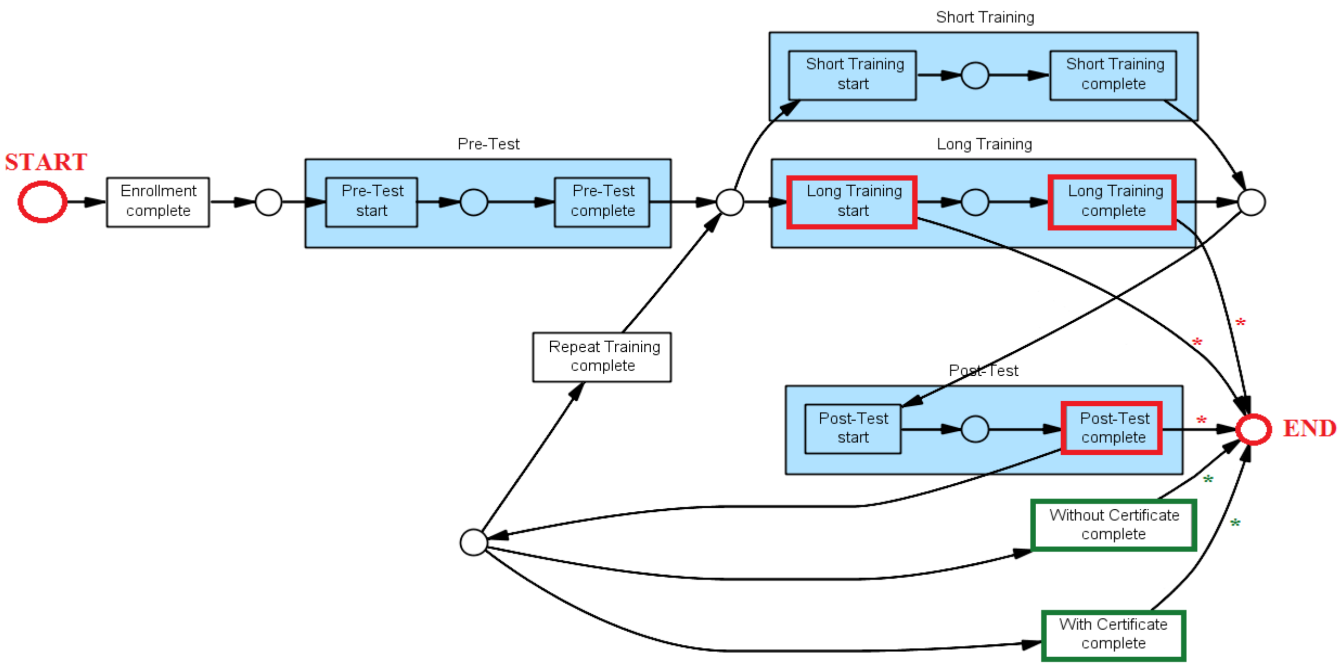

Figure 2. A screenshot of the resulting Alpha model/graph based on the event log collected during an Academic Writing (English) training course at a private university in Bangkok, Thailand.

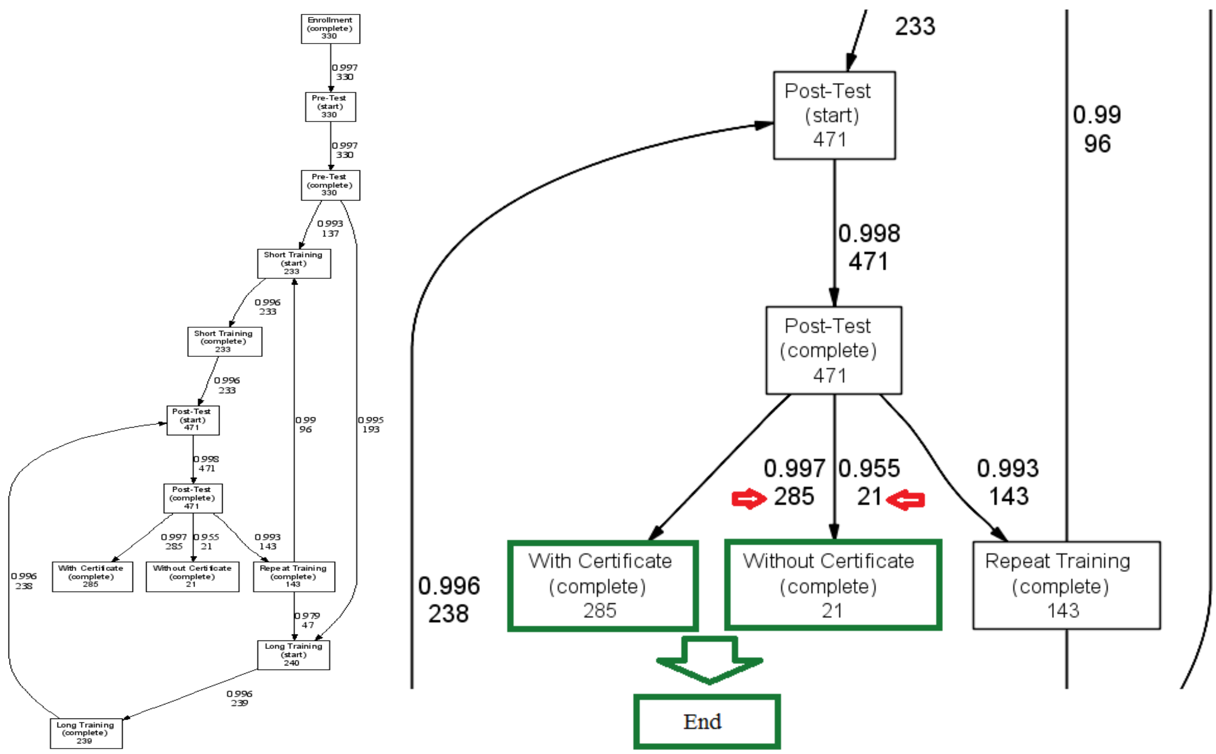

Figure 3. Two screenshots of the resulting Heuristic Miner model/graph based on the event log collected during an Academic Writing (English) training course at a private university in Bangkok, Thailand.

Furthermore, being aware of the fact that the square boxes in Fig. 3 are the tasks; the upper number on the arcs indicates the significance metric/coefficient between thetasks while the lower number on the arcs indicates the correlation metric/coefficient. The number in the event box indicates the number of times (i.e., frequency) the tasks in that box are performed [5], [22], [23],[3]. For instance, in this event log, there are 330 occurrences of the task "Enrollment (Complete)". The number on the arcs indicates the number of times the connection is used. For example, the number 137 on the " Pre-Test (complete)" to "Short Training (Start)" arc indicates that the task "Pre-Test (complete)" is 137 times followed by the task " Short Training (Start)". Similarly, the number 193 on the "Pre-Test (complete)" to

"Long Training (Start)" arc indicates that the task "Pre-Test (complete)" is 193 times followed by the task "Long Training (Start)". In addition, the dependency measure indicates dependency relation between two activities. A high value (close to 1.0) means that we are very sure that there is a dependency relation between the connected tasks [24].

By considering the resulting Heuristic miner graph/model shown in Fig. 3, we realize that the tasks/activities "Post-Test (Start and Complete)", "Pre-Test (Start and Complete)", and "Enrollment (Complete)" were identified as the most significant (i.e., top-3) tasks —during the *Academic Writing (English)* training course at the private university in Bangkok, Thailand— with absolute and relative frequencies of 942 (28.32%), 660 (19.84%), and 330 (9.92%) cases, respectively. In other words, 86.36% of the applicants/students managed to achieve the *Academic Writing (English)* certificate successfully (i.e., 285 students with certificate divided by a total of 330 students), while 6.36% of the applicants/students (i.e., 21 students with certificate divided by a total of 330 students) failed to achieve any certificate (after maximum 3 attempts to repeat the training programs). Surprisingly, 7.28% of the applicants/students (i.e., 24 cases) neither achieved a certificate nor failed the course training (i.e., they dropped out the training course before any success or failure). Therefore, 24 processes are missed (i.e., -24) before the tasks/activities "With Certificate" and "Without Certificate", and 77 processes are remained (i.e., +24) after the "With Certificate" and "Without Certificate" tasks/activities. Interestingly, 43.33% of the applicants/students (i.e., 143 students with certificate divided by a total of 330 students) had to repeat the *Academic Writing (English)* training course again. Accordingly, based on the conformance checking results, 60.60% of the applicants/students took the "Long Training" program while 39.40% of them followed the "Short Training" program during the *Academic Writing (English)* training course (see Fig. 4).

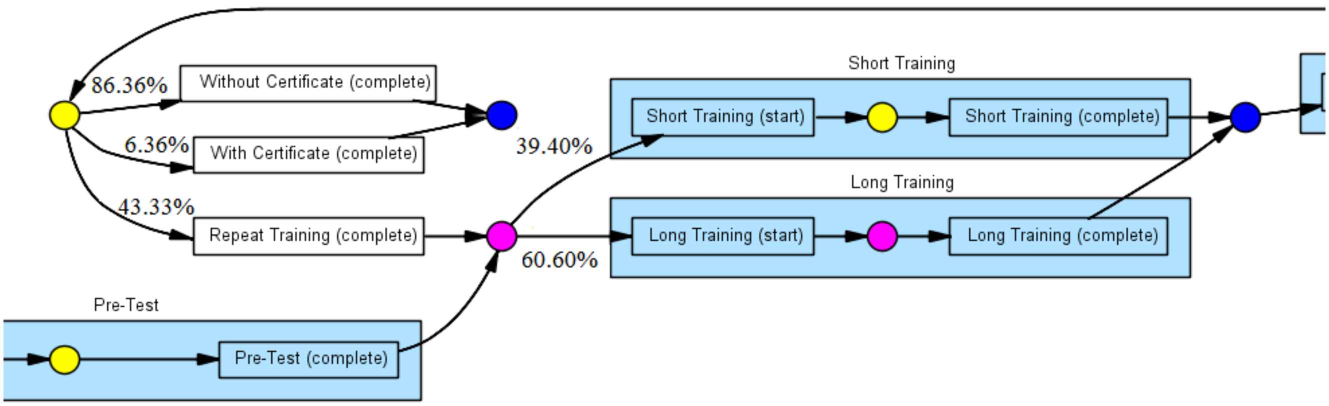

Figure 4. A screenshot of the conformance checkeing results based on the event log collected during an Academic Writing (English) training course at a private university in Bangkok, Thailand.

Fig. 5 shows three screenshots that provide detailed information about the problems encountered during the log replay. The *model* perspective diagnoses information about token counter (i.e., number of missing/left tokens), failed tasks (i.e., tasks that were not enabled), remaining tasks (i.e., tasks that remained enabled), path coverage (i.e., the tasks and arcs that were used during the log replay) and passed edges (how often every arc in the model was used during the log replay) [25],[24]. Based on the following *model* perspective, we could detect the existing discrepancies of the collected course training event log leading to +24 missed cases as well as the -24 remained cases in total. By studying the *model* perspective results, we could specify the number of the missing/left tokens as well as the remained tokens. As we consider, 22 tokens are missed (i.e., +22) before the activities "With Certificate (Complete)" and "Without Certificate (Complete)". In the same way, 24 tokens are remained (i.e., -22) after the activities "With Certificate (Complete)" and "Without Certificate (Complete)" in the "End" token of the *Academic Writing (English)* learning/teaching process. On the other hand, 1 token is missed (i.e., +1) between the activities "Long Training (Start)" and "Long Training (Complete)". One more token also is missed (i.e., +1) after the activities "Short Training (Complete)" and "Long Training (Complete)".

V. Conclusion

The main objective the study was to apply process mining techniques based on an event log previously captured, stored and collected from a private university— during an *Academic Writing (English)* course training— in Bangkok, Thailand. Using the Alpha and Heuristic algorithms (as two process mining discovery approaches), we could reconstruct (and simulate) process models compatible with the event log. The results showed that the tasks/activities "Post-Test (both Start and Complete)", "Pre-Test (both Start and Complete)", and "Enrollment (only Complete)" were identified as the most significant (i.e., top-3) tasks during the *Academic Writing (English)* learning/teaching process with absolute and relative frequencies of 942 (28.32%), 660 (19.84%), and 330 (9.92%) cases, respectively. In total, 86.36% of the applicants/students managed to achieve the *Academic Writing (English)* certificate successfully, while 6.36% of the applicants/students failed to achieve any certificate after maximum 3 attempts to repeat the training programs. Surprisingly, 7.28% of the applicants/students neither achieved a certificate nor failed the course training by dropping out the course before ending the training process. Accordingly, 60.60% of the applicants/students took the "Long Training" program while 39.40% of them followed the "Short Training" program during the *Academic Writing (English)* training course.

Using the Performance Analysis with Petri net technique (as a process mining conformance checking approach), we could provide detailed information about the problems encountered during the log replay. Consequently, we investigated the event log with respect to the number of missing/left tokens, failed tasks, remaining tasks, path coverage and passed edges. Based on the resulting *model* perspective, we could detect the existing discrepancies of the collected course training event log leading to +24 missed tokens as well as -24 remained tokens altogether. By studying the *model* perspective results, we could specify the number of the missing/left tokens as well as the remained tokens. We found out that 22 tokens were missed before the activities "With Certificate (Complete)" and "Without Certificate (Complete)". In the same way, 24 tokens were remained after the activities "With Certificate (Complete)" and "Without Certificate (Complete)" in the "End/Finish" token of the *Academic Writing (English)* learning/teaching process. On the other hand, 1 token was missed between the activities "Long Training (Start)" and "Long Training (Complete)", while 1 more token also was missed after the activities "Short Training (Complete)" and "Long Training (Complete)". Consequently, the process mining conformance checker techniques enabled us to indicate the points of non-compliant behavior for every case in the collected course training event log.

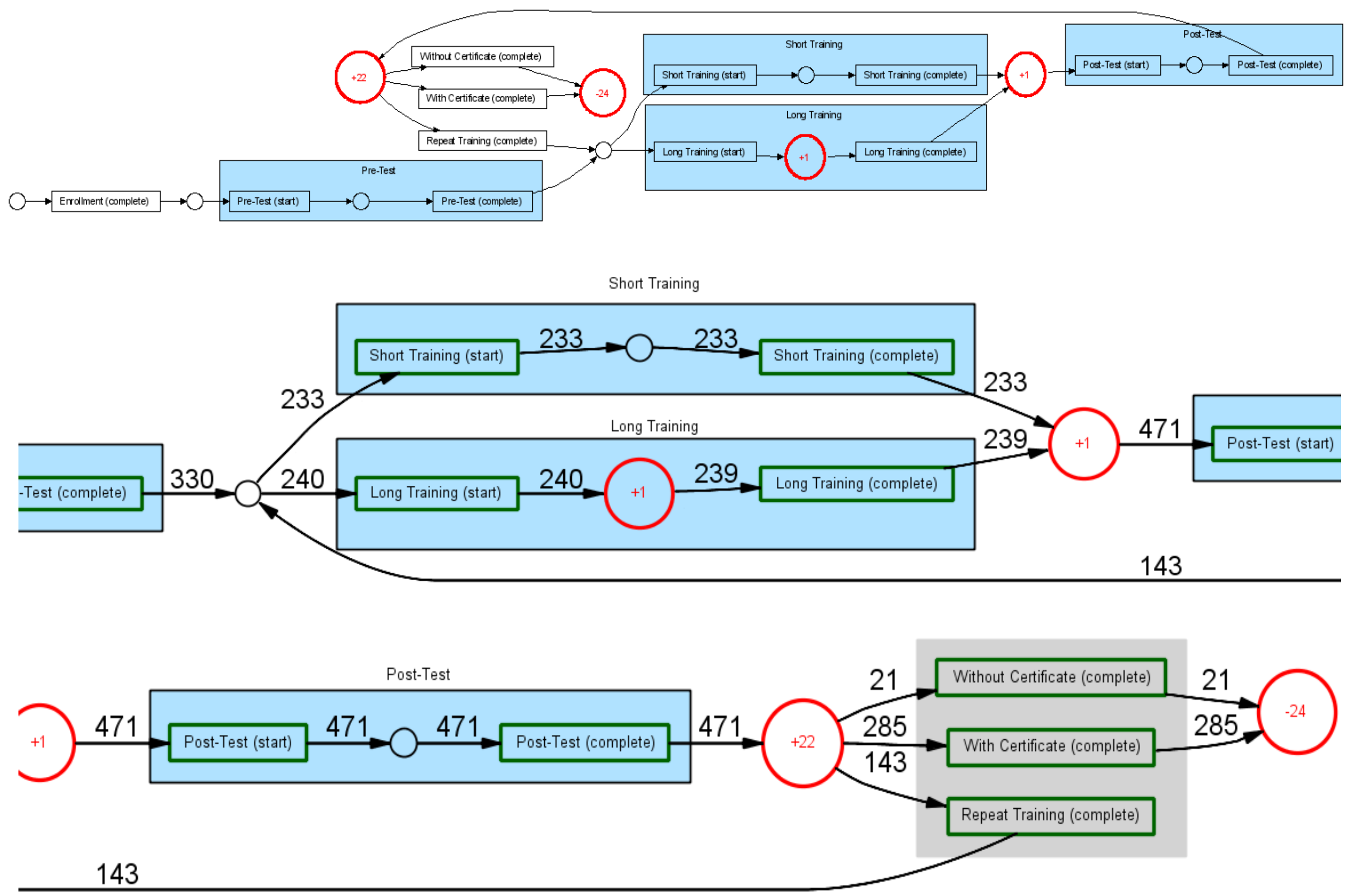

Figure 5. Three screenshots of the model perspective of the performance analysis with Petri net technique enables us to diagnose information about token counter (number of missing/left tokens), failed tasks (tasks that were not enabled), remaining tasks (tasks that remained enabled), path coverage (the tasks and arcs that were used during the log replay) and passed edges (how often every arc in the model was used during the log replay). Based on the model perspective, we could detect the existing discrepancies of the collected course training event log leading to +24 missed cases and -24 remained tokens in total [25],[24].

REFERENCES

[1] Website of Process Mining Research Tools Application, Math and Computer Science department, Eindhoven University of Technology, Eindhoven, The Netherlands, 2009. [Online]. Available: http://www.processmining.org/

[2] W. Premchaiswadi and P. Porouhan, Process modeling and bottleneck mining in online peer-review systems.. 441 s.l. : SpringerPlus, August 22, 2015, Computer Science , Vol. 4, pp. 1-18.

[3] P. Porouhan, N. Jongsawat, and W. Premchaiswadi, Process and Deviation Exploration through Alpha-Algorithm and Heuristic Miner Techniques, *Proceedings of the* 12th IEEE International Conference of ICT & Knowledge Engineering, Bangkok, Thailand, 2014, IEEE Xplore, pp. 83-89.

[4] W.M.P. van der Aalst, and K.M. van Hee. Workflow Management: Models, Methods, and Systems. MIT press, Cambridge, MA, 2002.

[5] W.M.P. Van Der Aalst. *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Springer-Verlag, Berlin, 2011.

[6] J.E. Cook and A.L. Wolf. (1998). Discovering Models of Software Processes from Event-Based Data. *ACM Transactions on Software Engineering and Methodology*. [Online]. *volume 7(3)*. pp. 215–249. Available: http://wwwis.win.tue.nl/~wvdaalst/publications/p128.pdf

[7] J.E. Cook and A.L. Wolf. (1999). Software Process Validation: Quantitatively Measuring the Correspondence of a Process to a Model. *ACM Transactions on Software Engineering and Methodology*. [Online]. *volume 8(2)*. pp. 147–176. Available: http://dl.acm.org /citation.cfm?id=304401

[8] R. Agrawal, D. Gunopulos, and F. Leymann. (1998). Mining Process Models from Workflow Logs. Presented at the Sixth International Conference on Extending Database Technology. [Online]. pages 469–483. Available: http://citeseerx.ist.psu.edu/ viewdoc/summary?doi=10.1.1.25.8660

[9] M.K. Maxeiner, K. Kuuspert, and F. Leymann. (2001). Data Mining von Workflow-Protokollen zur teilautomatisierten Konstruktion von Prozemodellen. *Informatik Aktuell Springer*. Berlin, Germany, pp. 75–84

[10] J.E. Cook and A.L. Wolf, "Event-Based Detection of Concurrency", in *Proceedings of the Sixth International Symposium on the Foundations of Software Engineering (FSE-6)*, 1998, pp. 35–45.

[11] G. Schimm. (2000). Generic linear business process modeling. *Springer*. [Online]. *volume 1921 of Lecture Notes in Computer Science*, pp. 31–39. Available: http://link.springer.com/chapter/ 10.1007%2F3-540-45394-6_4

[12] G. Schimm. (2002). Process miner—a Tool for Mining Process Schemes from Event-based Data. Presented in Proceedings of the 8th European Conference on Artificial Intelligence (JELIA). [Online]. Available:http://download.xn--geschftsprozessmanagement-pec.de/ processminer.pdf

[13] J. Herbst, "A Machine Learning Approach to Workflow Management", in *Proceedings 11th European Conference on Machine Learning*, Berlin, 2000, pp. 183–194.

[14] J. Herbst, D. Karagiannis, "Integrating machine learning and workflow management to support acquisition and adaptation of workflow models", in *Proceedings of the Ninth International Workshop on Database and Expert Systems Applications, IEEE*, Ulm, Germany, 1998, pp. 745–752.

[15] J. Herbst, D. Karagiannis, "An inductive approach to the acquisition and adaptation of workflow models", in *Proceedings of the IJCAI'99 Workshop on Intelligent Workflow and Process Management: The New Frontier for AI in Business*, Stockholm, Sweden, 1999, pp. 52–57.

[16] A.J.M.M. Weijters, and W.M.P. van der Aalst. (2001). Process mining: discovering workflow models from event-based data. Presented in Proceedings of the 13th Belgium–Netherlands Conference on Artificial Intelligence (BNAIC 2001). [Online]. Available: http://wwwis.win.tue.nl/~wvdaalst/publications/p128.pdf

[17] A.J.M.M. Weijters, and W.M.P. van der Aalst. (2001). Rediscovering workflow models from event-based data. Presented in Proceedings of the 11th Dutch-Belgian Conference on Machine Learning (Benelearn 2001). [Online]. Available: http://wwwis.win. tue.nl/ ~wvdaalst/publications/p188.pdf

[18] W.M.P. van der Aalst, and B.F. van Dongen. (2002). Discovering Workflow Performance Models from Timed Logs. Presented in International Conference on Engineering and Deployment of Cooperative Information Systems (EDCIS 2002). [Online]. Available: http://tmpmining.win.tue.nl/_media/publications/ aalst2002b.pdf

[19] W.M.P. van der Aalst, A.J.M.M. Weijters, and L. Maruster. (2004). Workflow Mining: Discovering Process Models from Event Logs. Presented in IEEE Transactions on Knowledge and Data Engineering (TKDE). [Online]. Available: http://www.processmining. org/blogs/pub2004/workflow_mining_discovering_process_models_ from_event_logs

[20] P. Porouhan, W. Premchaiswadi, and S. Weerapong, "Process Mining: Using α-Algorithm as a Tool (A Case Study of Student Registration)", *Proceedings of the IEEE ICT & Knowledge Engineering*, Bangkok, Thailand, 2012, pp. 213- 220.

[21] W. Premchaiswadi and P. Porouhan, Process modeling and decision mining in a collaborative distance learning environment, Decision Analytics 2015, 2:6 (4 August 2015).

[22] A. Karla Alves de Medeiros and A.J.M.M. (Ton) Weijters. (2008, February). ProM Framework Tutorial. Technische Universiteit Eindhoven. The Netherlands. [Online]. Available: http://tmpmining. win.tue.nl/_media/tutorial/promtutorialv2.pdf?id=tutorials&cache=c ache

[23] A.K.A. de Medeiros, B.F. van Dongen, W.M.P. van der Aalst and A.J.M.M. Weijters. (2004). Process Mining: Extending the α-algorithm to Mine Short Loops. Eindhoven University of Technology, Eindhoven, The Netherlands. [Online]. Available: http://alexandria. tue.nl/repository/books/576199.pdf

[24] Chang, Elizabeth J. and Sycara, Katia . Advances in Web Semantics I : Ontologies, Web Services and Applied Semantic Web. Berlin Heidelberg : Springer-Verlag Berlin Heidelberg, 2009. Vol. 4891.

[25] Medeiros, Ana Karla Alves de and Weijters, A.J.M.M. Ton. ProM Framework Tutorial. TU/e. [Online] February 2008. [Cited: October 20, 2015.] https://www.tmpmining.win.tue.nl/_media/tutorial/promtutorialv2.pdf.

[26] W.M.P. van der Aalst. Process Mining: Data science in Action. Coursera.org. [Online] Technische Universiteit Eindhoven and Coursera , November 12, 2014. [Cited: October 20, 2015.] MOOC . https://class.coursera.org/procmin-003/lecture/75.

[27] Günther, C., and W.M.P. van der Aalst (2007). Fuzzy mining: adaptive process simplification based on multi-perspective metrics. In G. Alonso, P. Dadam, & M. Rosemann (Eds.), International Conference on Business Process Management (BPM 2007) (Vol. 4714, pp. 328–343)., Lecture Notes in Computer Science Springer: Berlin.