Science and Technology RMUTT Journal

Vol.5 No.2 (2015) : 118-134
www.sci.rmutt.ac.th/stj
Online ISSN 2229-1547

Process Modeling and Bottleneck Mining in MXML-based Course Training Event Logs

Parham Porouhan[1*] and Wichian Premchaiswadi[1]

[1]Graduate School of Information Technology, Siam University
Petchkasem Road, Phasi-Chareon District, Bangkok, 10160, Thailand
*E-mail: parham@siam.edu

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) in order to extract knowledge from an event log previously collected from an information system —during a project management training course at a private university in Thailand. The event log was initially consisted of 548 process instances and 5,390 events in total. Using alpha algorithm we could reconstruct causality (in form of a Petri-net) from a set of sequences of events, while through heuristic algorithm we could derive XOR and AND connectors (in form of a C-net) based on the dependency, significance and correlation metrics/coefficients. The results showed 80% of the applicants/students managed to achieve the project management certificate successfully, while 6% of them fail to achieve any certificate (after maximum number of 3 attempts to re-take the course). Surprisingly, 14% of the applicants (77 cases) neither achieved a certificate nor failed the course. Therefore, in the second part of the study, we used conformance checker and performance analysis techniques in order to further analyze the points of non-compliant behavior (i.e., bottlenecks) for every case in the log. Subsequently, we could detect and identify the number of missing tokens, as well as the activities that were not enabled, or remained enabled.

Keywords: Process Mining, Model Discovery, Alpha algorithm, Heuristic algorithm, Process Simulation, ProM, Bottleneck Mining, Conformance Checker, Performance Analysis, MXML.

Received: Oct 22, 2015
Revised: Dec 29, 2015
Accepted: Dec 29, 2015

1. 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], [30],[31].

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” [30],[31].However, before any manipulation and investigation of the data in ProM process mining environment, it is important to deal with the appropriate type of format supported by ProM platform. To do this, ProMimport can be used in order to import event logs from various systems (e.g., Staffware and FLOWer) in such a way that they can be analyzed using ProM5.2 or ProM6.4.2. Fig. 2 shows a typical standard MXML format consisted of the ProcessInstance elements which correspond to cases. One ProcessInstance element may hold multiple AuditTrailEntry elements. Each of these elements represents an event. Accordingly, each AuditTrailEntry element may contain WorkfowModelElement, EventType, Timestamp, and Originator elements. The WorkfowModelElement and EventType are mandatory elements [3], [30],[31].

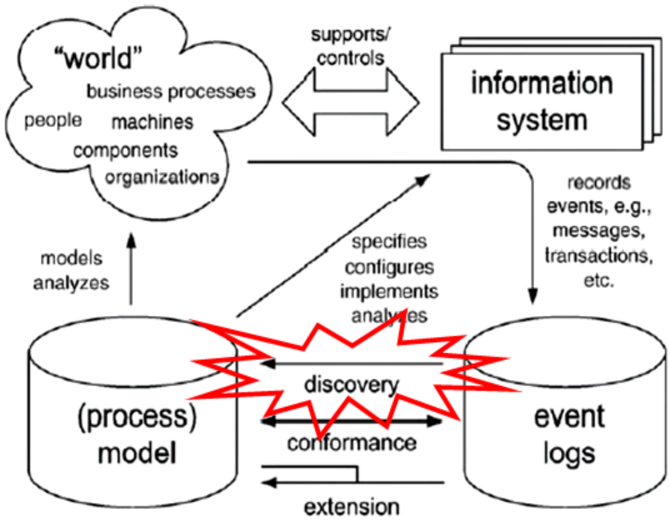

Fig. 1 A holistic view of the process mining model as well as its methods, dimensions, and relationships. (right) A typical structure of a MXML-formatted event log (Sources: Process Mining official website and Process Mining: Discovery, Conformance and Enhancement of Business Processes) [4,5].

2. 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 [4]-[6], [30],[31]. Cook and Wolf investigated the appliance of the process mining by focusing on the software engineering processes. In [4], [30],[31] 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 [4]were limited to sequential behavior. Cook and Wolf extended their work to concurrent processes in [5]. 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 [30],[31].In [6] Cook and Wolf provided a measure to quantify discrepancies between a process model and the actual behavior as registered using event-based data. The idea of applying process mining in the context of workflow management was first introduced in [7]. 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[30],[31].

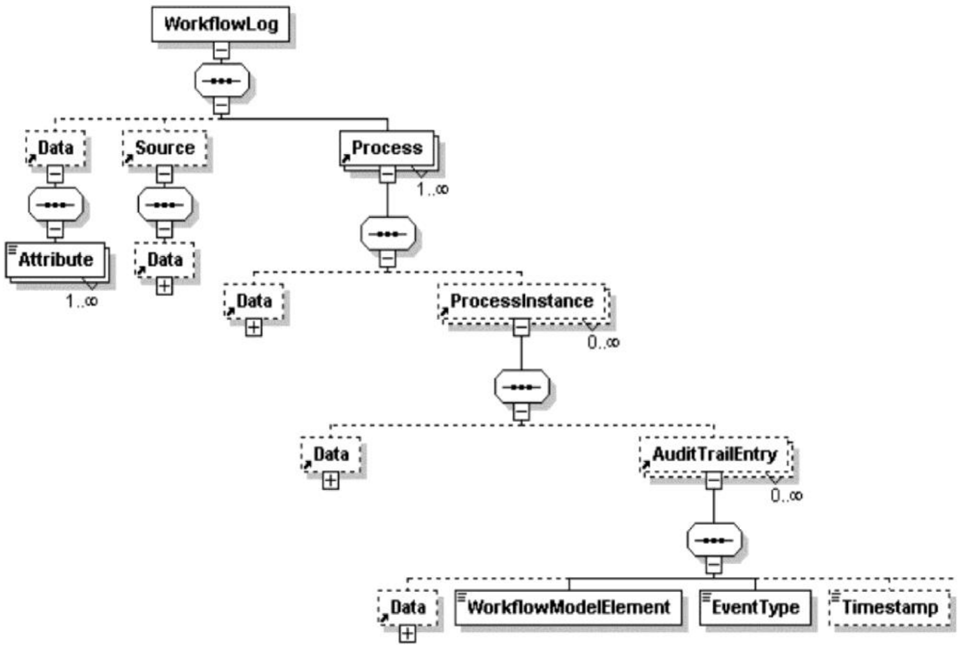

Fig. 2 A typical structure of a MXML-formatted event log (Sources: Process Mining official website and Process Mining: Discovery, Conformance and Enhancement of Business Processes) [4,5].

3. Methodology

In this study, we used an event log describing the students of a private university in Bangkok (Thailand) who voluntarily registered and made a payment in order to undertake a Project Management course leading to an issued "successfully completed the certificate" or failure of the course "without any certificate". In general, a total of 548 students registered and attended the course from 16 October 2556 (i.e., starting time of capturing the data based on the Thai solar calendar which is equal with the year 2013) to 18 August 2558 (i.e., finish time of capturing the data based on the Thai solar calendar which is equal with the year 2015). The offered Project Management course accepted students of the 4 different international programs as follows: "BBA Program" (172 students), "Hotel and Hospitality Management Program" (68 students), "Master of Business Administration Program, MBA" (284 students) and "Information Technology in Business Program, PhD of IT in Business" (24 students).

The Project Management's course training started by registering the course (and making a payment) by a student (applicant). After registration and payment, the student needs to undertake the "Initial Evaluation and Assessment Test" where the level of the student's knowledge and familiarity with the project management topics is evaluated, analyzed and categorized with regard to 5 different levels as follows: "Elementary", "Pre-Intermediate", "Intermediate", "Upper-Intermediate", and "Advanced". Once the applicant is evaluated, the student can attend the "Course

Training Classes/Sessions". However, the "Course Training Classes/Sessions" are divided into two main categories with different focus and learning aspects. The applicants whose level of knowledge (and extent of familiarity with project management topics) is evaluated as "Elementary" or "Pre-Intermediate" have to attend and accomplish the "Full-Term Course Training" within 4 months. The applicants whose level of knowledge (and extent of familiarity with project management topics) is evaluated as "Upper-Intermediate" or "Advanced" have to attend and accomplish the "Intensive-Term Course Training" within 6 weeks/1.5 month. Those applicants whose level of knowledge (and extent of familiarity with project management topics) is evaluated as "Intermediate" can attend and accomplish any of the "Full-Term Course Training" or "Intensive-Term Course Training" based on the consultancy with evaluators and trainers (instructors) or based on their willingness and readiness to select (and attend) any of the available options/choices.

Once an applicant completed the assigned "Course Training Classes/Sessions", the student has to take a "Qualification Exam" a systematic determination of the student's qualification to receive (or not receive) an accredited project management certificate, using criteria compatible with a set of standards (and performance assessment metrics). Accordingly, the main objective of the "Qualification Exam" is to reevaluate (reassess) the extent of progress/or improvement made in the level of the students' knowledge and understanding about the project management topics and issues. If a student can pass the "Qualification Exam" properly with required scores, then an accredited/formal project management certificate is granted. If a student cannot pass the "Qualification Exam", then he or she should retake a new training course based on the level of progress and advancement made his/her knowledge in regard to the project management topics and learning materials (or in some cases, repeat the same training course while not any considerable progress or advancement is made in the level of his/her learning competencies). Subsequently, if a student cannot pass the "Qualification Exam" after 3 attempts, then the student fails the course with no certificate and is exempted (or banned) from re-registering the course for a period of at least 5 years.

This study is divided into 2 main parts. In the first part of the study, we applied two process mining discovery techniques (i.e., alpha and heuristic algorithms) with the purpose of extracting knowledge from the above-mentioned project management training course event log. Accordingly, we used Alpha algorithm in order to reconstruct causality in form of a Petri-net. The Heuristic algorithm enabled us to derive XOR and AND connectors in form of a C-net in terms of the three metrics as follow: the dependency, the significance and the correlation coefficients. In the second part of the study, we applied conformance checker and performance analysis techniques (from conformance checking/audit class) in order to identify and detect the points of non-compliant behavior and bottlenecks for every case in the log with respect to the number of missing tokens and remaining activities [33],[36].

Table 1 Number of applicants/students who registered the project management training course.

	Number of Students	**Attendance Percentage**
BBA (Undergraduate)	172	31%
Hotel and Tourism (Undergraduate)	68	12.5%
MBA (Graduate, Master)	284	52%
IT in Business (Graduate, PhD)	24	4.5%

4. Findings and Results

4.1 Alpha Algorithm

The Alpha-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],[2]. 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 TechnischeUniversiteit Eindhoven (TU/e). In this paper, we used Alpha-algorithm as a very basic process mining technique to identify the routing constructs within an event log —dealing with handling of a project management training course — collected from an information system at a private university in Bangkok, Thailand [27],[29],[30],[31]. Due to the fact that our main emphasis was on the “project management 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. Accordingly, after applying the appropriate filtering on the collected event log, our event log contained only two types of processes: Start and Complete. Filtering the event log allowed us to merely focus on the started and completed tasks, rather than ongoing (or undone) processes [27],[29],[30],[31]. Knowing that the alpha-algorithm mines the control flow perspective of a process, its algorithm [3],[27], [28] is comprehensively defined in Figure 3 as follows:

Let L be an event log over T. $\alpha(L)$ is defined as follows.

1. $T_L = \{ t \in T \mid \exists_{\sigma \in L}\, t \in \sigma \}$,
2. $T_I = \{ t \in T \mid \exists_{\sigma \in L}\, t = \mathit{first}(\sigma) \}$,
3. $T_O = \{ t \in T \mid \exists_{\sigma \in L}\, t = \mathit{last}(\sigma) \}$,
4. $X_L = \{ (A,B) \mid A \subseteq T_L \wedge A \neq \emptyset \wedge B \subseteq T_L \wedge B \neq \emptyset \wedge \forall_{a \in A} \forall_{b \in B}\, a \rightarrow_L b \;\wedge\; \forall_{a1,a2 \in A}\, a_1 \#_L\, a_2 \;\wedge\; \forall_{b1,b2 \in B}\, b_1 \#_L\, b_2 \}$,
5. $Y_L = \{ (A,B) \in X_L \mid \forall_{(A',B') \in X_L}\, A \subseteq A' \wedge B \subseteq B' \Rightarrow (A,B) = (A',B') \}$,
6. $P_L = \{ p_{(A,B)} \mid (A,B) \in Y_L \} \cup \{i_L, o_L\}$,
7. $F_L = \{ (a, p_{(A,B)}) \mid (A,B) \in Y_L \wedge a \in A \} \cup \{ (p_{(A,B)}, b) \mid (A,B) \in Y_L \wedge b \in B \} \cup \{ (i_L, t) \mid t \in T_I \} \cup \{ (t, o_L) \mid t \in T_O \}$, and
8. $\alpha(L) = (P_L, T_L, F_L)$.

Fig. 3 Definition of the Alpha algorithm. Source: Process Mining Discovery, Conformance, and Enhancement of Business Processes by Wil van der Aalst [2],[33].

Consequently, the resulting graph/model produced by Alpha algorithm —based on the project management training course event log—is shown in Figure 4. By contemplating on the resulting Alpha model, we can clearly investigate the control-flow of the course training processes/tasks (at a private university in Thailand) as well as the dependencies among its tasks. The Alpha model also indicates which tasks precede which other ones. For example, in this case, the Initial Evaluation and Assessment task/process always precedes the Intensive Course training task/process. In the same way, the Initial Evaluation and Assessment task/process always precedes the Full-Term Course training task/process. Another advantage of the Alpha models is that they can reveal on concurrent tasks (and loops) taken place in an event log. As we can see in Figure 4, the course training data does not include any concurrent tasks or loops. In short, the resulting Alpha process model that summarizes the general flow followed by all cases in the project management training course event log. This information has crucial importance as it gives us feedback about how cases are actually being executed during the course training process [33].

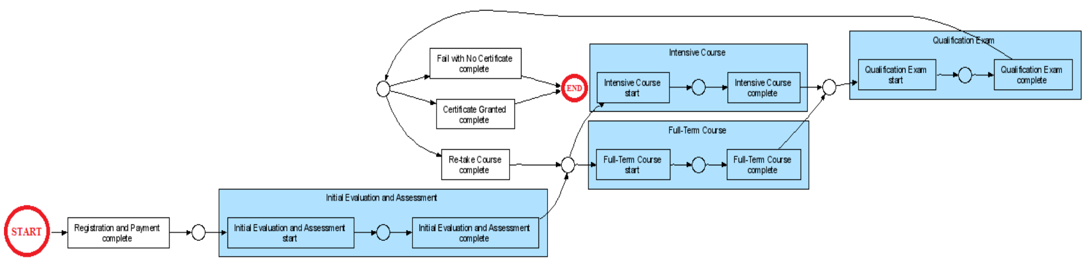

Fig.4 A screenshot of the resulting Alpha model/graph based on the event log collected during a project management training course.

4.2 Heuristic Mining

As discussed in [34], the formal approaches like the alpha algorithm presupposes that the mined log must be complete and there should not be any noise in the log. However, this is not practically possible. Also, Alpha-algorithm does not make use of any frequency information (i.e. frequency of various dependencies of the tasks in an event log) which can be quite useful in situations of noise. To say simple, Alpha algorithm is not robust to logs that contain noisy data [2],[30],[31]. Therefore, we used Heuristic Miner algorithm as a process discovery to consider the frequency of the process instances (despite of Alpha algorithm) in order produce models/graphs which are less sensitive to noise and the incompleteness of logs as shown in Fig. 5. Therefore, in the Heuristic Miner approach, the log files are examined for causal dependencies (similar to Alpha algorithm), but additionally the frequency of their repartition also is considered and taken into account (on contrary to Alpha algorithm method).

$$|a >_L b| = \sum_{\sigma \in L} L(\sigma) \times |\{1 \leq i < |\sigma| \mid \sigma(i) = a \wedge \sigma(i+1) = b\}|$$

$|a \Rightarrow_L b|$ is the value of the dependency relation between a and b:

$$|a \Rightarrow_L b| = \begin{cases} \dfrac{|a >_L b| - |b >_L a|}{|a >_L b| + |b >_L a| + 1} & \text{if } a \neq b \\ \dfrac{|a >_L a|}{|a >_L a| + 1} & \text{if } a = b \end{cases}$$

Fig. 5 Despite of the Alpha algorithm, the Heuristic Miner algorithm considers the frequency of the process instances with respect to a dependency measure calculated between two activities. Source: Process Mining Discovery, Conformance, and Enhancement of Business Processes by Wil van der Aalst [2],[33].

The resulting graphs/models produced by Heuristic Miner algorithm —based on the project management training course event log—are shown in Fig. 6. The Heuristic Miner algorithm deals with two fundamental metrics as follows: (a) Significance and (b) Correlation. “Significance” deals with the relative importance of behavior while “Correlation” deals with how closely related two events following one another are [31],[35]. By contemplating on the resulting Alpha model, we can answer some crucial questions such as how are the cases actually being executed? Or how is the distribution of all cases over the different paths through the process? Or what is the most frequent path for every process model? [33],[36].

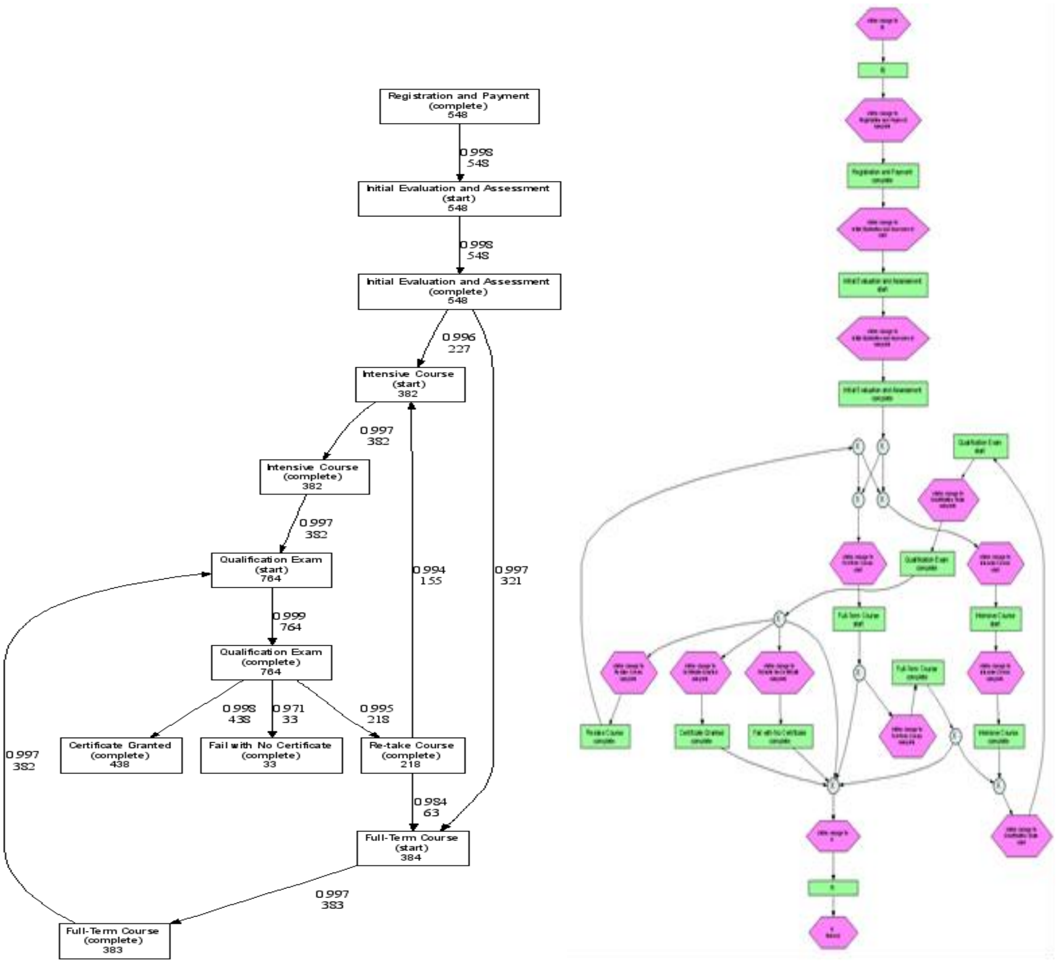

Fig. 6 Screenshots of the resulting Heuristic Miner model/graph based on the event log collected during a project management training course at a private university in Bangkok, Thailand.

In Fig. 6, the square boxes are the tasks; the upper number on the arcs indicates the significance metric/coefficient between the tasks, 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 [2],[3],[28],[31]. For instance, in this event log, there are 548 occurrences of the task "Registration and Payment". The number on the arcs indicates the number of times the connection is used. For example, numbers 548 and 227 on the "Initial Evaluation and Assessment (start)" to " Initial Evaluation and Assessment (complete)", and " Initial Evaluation and Assessment (complete)" to "Intensive Course (start)" arc indicates that the task " Initial Evaluation and Assessment (start)" is 548 times followed by the task " Initial Evaluation and Assessment (complete)", and the task "Initial Evaluation and Assessment (complete)" is 227 times followed by the task to " Intensive Course (start)". Furthermore, 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 [36].

As shown in Fig. 7, the tasks/activities "Qualification Exam", "Full-Term Course", and "Intensive Course" were identified as the most significant tasks —during the project management training course at a private university in Bangkok, Thailand— with absolute and relative frequencies of 1528 (31.54%), 767 (15.83%), and 764 (15.77%) cases, respectively. Moreover, as shows in Fig. 8, majority of the applicants/students (80%) managed to achieve the project management certificate successfully, while 6% of the applicants/students fail to achieve any certificate (after maximum 3 attempts to re-take the course). Surprisingly, 14% of the applicants/students (i.e., 77 cases) neither achieved a certificate nor failed the course training (i.e., they dropped out the training course before any success or failure). Therefore, 77 processes were missed (i.e., -77) before the tasks/activities "Certificate Granted" and "Fail with No Certificate), and 77 processes are remained (i.e., +77) after the "Certificate Granted" and "Fail with No Certificate) tasks/activities. Interestingly, 40% of the applicants/students had to re-take (or repeat) the course again. Accordingly, 53% of the applicants/students took the full-term course while 47% of them followed the intensive project management training course. Fig.9 and Fig. 10 show two perspectives provide detailed information about the problems encountered during the log replay. The *model* perspective (see Fig.9) diagnoses 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) [33],[36]. Based on the *model* perspective (Fig. 9), we could detect the existing discrepancies of the collected course training event log leading to -77 missed cases as well as the +77 remained cases in total. The *log* perspective (Fig. 10) indicates the points of non-compliant behavior for every case in the log. By contemplating on the results, we can clearly see how many traces are not compliant with the log (i.e., highlighted in orange color). As shown in Fig. 10, one of the possible bottlenecks of the log whereas the activity/task "Intensive Course" was never started couple of the applicants/students.

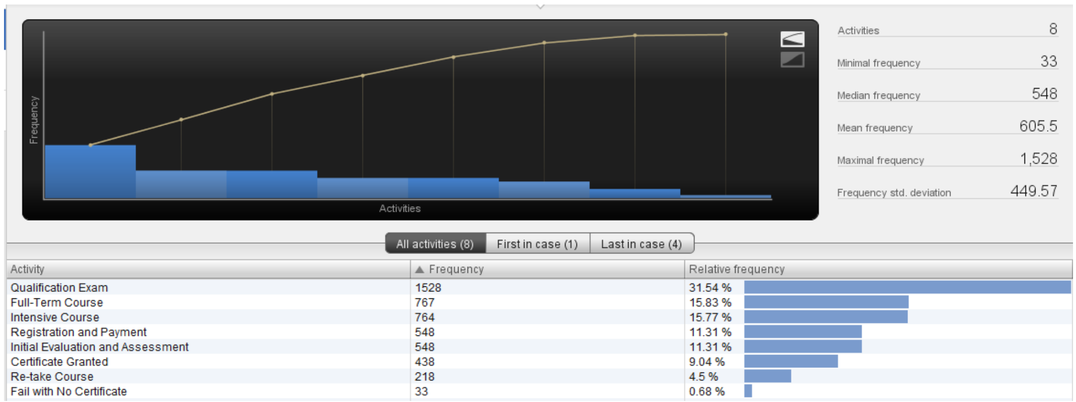

Fig. 7 A distribution of the log view information with respect to total and relative frequencies. The event log was originally collected from a project management training course at a private university in Bangkok, Thailand.

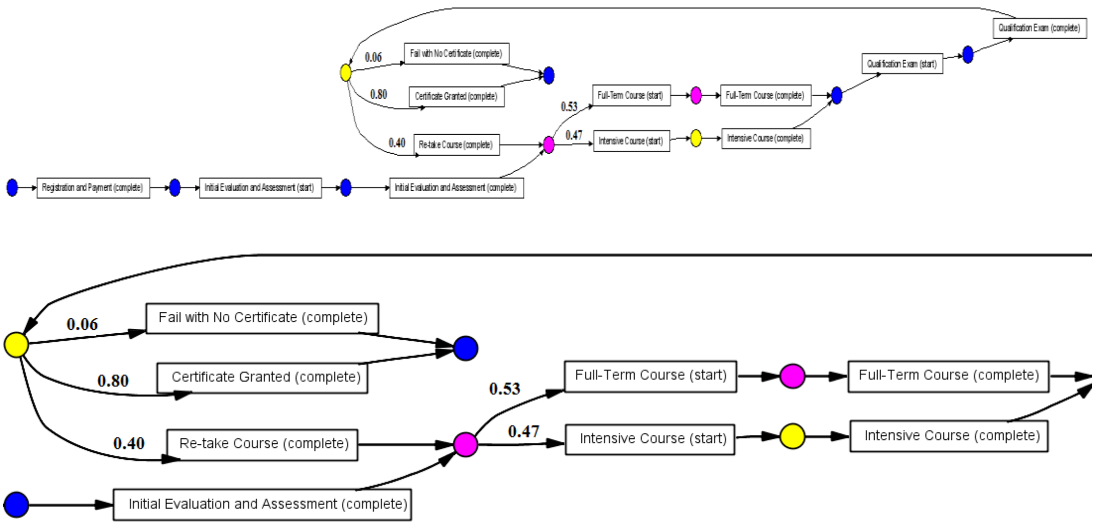

Fig. 8 80% of the applicants/students managed to achieve the project management certificate successfully, while 6% of the applicants/students fail to achieve any certificate (after maximum 3 attempts to re-take the course). 14% of the applicants/students (77 cases) neither achieved a certificate nor failed the course training. Further investigation revealed that a total of 77 of applicants dropped out the training course before any success or failure.

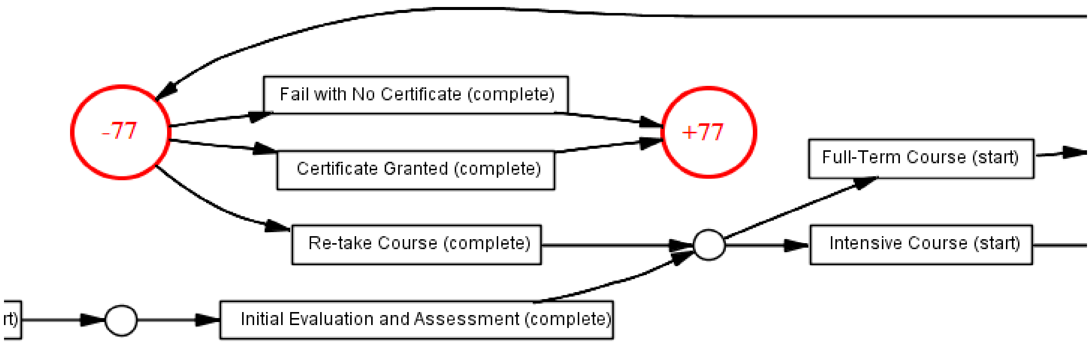

Fig. 9 The model perspective of the conformance checker technique could diagnose information about token counter (number of missing/left tokens), failed tasks (tasks that were not enabled), remaining tasks (tasks that remained enabled), path coverage (the tasks and arcs that were used during the log 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 -77 missed cases as well as the +77 remained cases in total. (adopted from [33],[36])

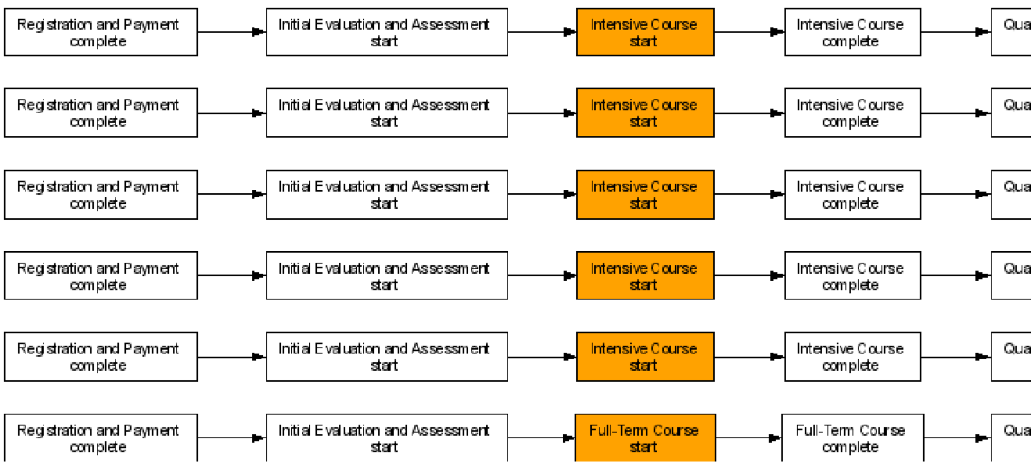

Fig. 10 A selected screenshot of the log perspective of the conformance checker technique could diagnose the existing bottlenecks whereas the activity/task of the "Intensive Course" was never started after the activities "Initial Evaluation" and "Assessment" during the project management training course. (adopted from [33],[36])

5. Conclusion

The main rationale for the study was to apply process mining as a process management technique based on an event log previously captured, stored and collected from a private university—during a project management course training— in Bangkok, Thailand. Using the Alpha and Heuristic algorithms (from discovery class), we could automatically construct process models compatible with the event log. The results showed that the tasks/activities "Qualification Exam", "Full-Term Course", and "Intensive Course" were the most significant tasks with absolute and relative frequencies of 1528 (31.54%), 767 (15.83%), and 764 (15.77%) cases, respectively. Moreover, 80% of the applicants/students managed to achieve the project management certificate successfully, while 6% of the applicants/students fail to achieve any certificate (after maximum number of total 3 attempts to re-take the course). In complete surprise, 14% of the applicants (i.e., 77 cases) neither achieved a certificate nor failed the course training. By further investigation, we found out that they had dropped out the training course before any success or failure in achieving the project management certificate. Accordingly, 77 processes were missed (i.e., -77), and 77 processes were remained (i.e., +77) just before and after the "Certificate Granted" and "Fail with No Certificate tasks, respectively. Interestingly, 40% of the applicants/students had re-taken (or repeat) the course more than one time. In total, 53% of the applicants took the full-term course and 47% of them followed the intensive project management training course.

In addition, Conformance Checker and Performance analysis techniques (from conformance/audit class) made us capable of discovering the current bottlenecks of the training course processes. The model perspective of the Conformance Checker approach enabled us to diagnose more 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 (i.e., how often every arc in the model was used during the log replay). Consequently, the log perspective of the Conformance Checker technique enabled us to indicate the points of non-compliant behavior for every case in the project management course training event log. By studying the results, we could specify the number of the traces that were not compliant with the log whereas some of the applicants never started and completed the "Intensive Course" activity by dropping out of the training course. Therefore, Process mining techniques enabled us to extract interesting information and knowledge from an educational event log which was initially consisted of 548 process instances and 5,390 events in total. By using the discovery and conformance checking techniques of the process mining tools we could discover models describing processes, organizations, and bottlenecks. Moreover, the proposed approach enabled us to monitor deviations by comparing the observed events with predefined models.

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

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

[4] 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].*volume7(3)*. pp. 215–249. Available: http://wwwis.win.tue.nl/~wvdaalst/publications/p128.pdf

[5] 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.

[6] 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

[7] 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

[8] M.K. Maxeiner, K. Kuuspert, and F. Leymann.(2001). Data Mining von Workflow- Protokollenzurteilautomatisierten Konstruktionvon Prozemodellen. *InformatikAktuell Springer*. Berlin, Germany, pp. 75–84.

[9] 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

[10] 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

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

[12] 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.

[13] 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.

[14] 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

[15] 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 (Benelearn2001).[Online]. Available: http://wwwis.win.tue.nl/~wvdaalst/ publications/p188.pdf

[16] 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

[17] 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

[18] R.S. Burt, and M Minor, *Applied Network Analysis: A Methodological Introduction*, coedited with Michael J. Minor. Beverly

Hills: Sage Publications, Newbury Park CA, 1983, 352 pages.

[19] J. Scott, and P. J. Carrington. (2011). Sage Handbook of Social Network Analysis.SAGE Publications Ltd. London. UK. [Online]. Available: http://arts.uwaterloo.ca/~pjc/pubs/Sage_Hbook/Sage%20 Chap17.pdf

[20] S. Wasserman, and K. Faust. Social Network Analysis: Methods and Applications. Cambridge University Press, Cambridge, 1994.

[21] J.L. Moreno, and H.H. Jennings, *Who Shall Survive?A new approach to the problem of human interrelations*.Nervous and Mental Disease Publishing Company, Washington, DC, 1934.

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

[23] W.M.P. van der Aalst, B.F. van Dongen, J. Herbst, L. Maruster, G. Schimm, and A.J.M.M. Weijters. (2003). Workflow Mining: A Survey of Issues and Approaches. *Data and Knowledge Engineering*.[Online].*vol* 47(2), pp.237–267. Available: http://140.115.80.66/data%20mining%20paper%20databases/Data%20and%20Knowledge%20Engineering/Workflow%20mining%20A%20survey.pdf

[24] 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]. Available: http://citeseerx.ist.psu.edu/messages/downloads exceeded.html

[25] P. Porouhan, W. Premchaiswadi, and W. Romsaiyud, "Process mining: Converting data from MS-Access Database to MXML Format", *Proceedings of the IEEE ICT & Knowledge Engineering*, Bangkok, Thailand, 2012, IEEE Xplore, pp. 205- 212.

[26] B.F. van Dongen, A.K.A. de Medeiros, H.M.W. Verbeek, A.J.M.M. Weijters and W.M.P. van der Aalst. (2009). User manual for converting data from a Microsoft Access Database to the ProM MXML format.TechnischeUniversiteit, Eindhoven, The Netherlands [Online]. Available: http://www.processmining.org/promimport/ tutorials

[27] 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.

[28] 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

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

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] Dumas, Marlon, W.M.P. van der Aalst and Hofstede, Arthur H. ter. Process-aware information systems: bridging people and software through process technology. New York, NY, USA : John Wiley & Sons, Inc. , 2005.

[35] Günther, C., and W.M.P. van der Aalst (2007). Fuzzy mining: adaptive process simplification based on multi-perspective metrics. InG. 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.

[36] 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.