

Adaptive Process Simplification and Conformance Checking in Event Logs

Parham Porouhan, Wichian Premchaiswadi
Graduate School of Information Technology
Siam University
Bangkok, Thailand

Abstract

In this paper, we used two classes of process mining techniques (i.e., Discovery and Conformance Analysis) in order to discover models and organizational structures related to the handling of proceedings' peer reviews in an international conference in Thailand. We monitored the deviations, comparing the observed events (real-life data) with the predefined models, as well. Knowing that ProM plugins receive the input logs only in MXML (Mining eXtensible Markup Language) and XES (eXtensible Event Stream) formats, we chose ProMimport as a framework to extract MXML log from an event log consisted of 87 cases (papers) and 3267 events. Alpha (a) Algorithm, Heuristic, Fuzzy and Social Network mining techniques (from Discovery class) were applied to automatically construct the proceedings' review models, primarily without any priori model. Though the Heuristic Miner closely followed the Alpha algorithm, the technique had the privilege to derive XOR and AND connectors from dependency relations of the event log. Next, the actual process behavior was projected onto fuzzy models. The result was an animation movie coming up with a better understanding of what has occurred in reality. Also, using Social Network Miner technique we could analyze the organizational perspective of the peer review process in terms of three metrics, namely as: (a) Handover of Work, (b) Working Together, and (c) Similar Tasks. Alternatively, having a priori model for review of the proceedings, we used LTL Checker and Performance Analysis techniques (from Conformance Checker class) to identify discrepancies between the log and the pre-defined model. After applying the LTL checking approach, the deviations were detected — leading to enrich the real model. In addition, Performance Analysis technique made us capable of projecting the bottlenecks all through the peer review system. In general, one of the main benefits of the techniques used in this paper is that information is objectively compiled. To say simple, we gathered valuable information about what actually was happening according to the review process of the papers and existing bottlenecks, and not what we just

thought or expected to see happening in the event log. Considering the results of the study, conference committee chairs can better evaluate the performance of the involving reviewers (as well as team members) within the assigned tasks. This will improve the performance, efficiency and effectiveness of the handling of reviews for prospective academic/educational conferences.

1. Introduction

Nowadays, it is editors' experience that the peer-review process is an essential part of the publication process. Not only does peer review provide an independent assessment of the importance and technical accuracy of the results described, but the feedback from referees conveyed to authors with the editors' advice, frequently results in manuscripts being refined so that their structure and logic is more readily apparent to readers.

Typically, the problem with manuscript selection is the inherent tension between referees and authors. Referees wish for only the most solid science to be published, yet when they 'switch hats' to that of author, they desire quick publication of their novel ideas and approaches. Authors of papers that blow against the prevailing winds bear a far greater burden of proof than normally expected in publishing their challenge to the current paradigm [1].

In 2011, Graduate School of Information Technology (and the Organizing Committee Members) at a private university in Thailand published a comprehensive web focus for International ICT & Knowledge Engineering Conference on the peer review system [2] [3]. In fact, the international conference has been held annually since 2003 with the cooperative effort for a number of organizations and universities in Thailand and other countries. The benefits of the developed reviewing platform for online submissions were as follows: (1) Online submissions were automatically acknowledged, (2) Authors could easily check the status of their submission during review, (3) Papers were easier to track, amend, and

update, (4) It was a simple method of sharing documents and sending communications throughout the review process, (5) Electronic submissions could also be edited more easily.

As a result, all of those contributions that were selected for peer-review were sent to at least three, but usually four or more, independent reviewers, selected by the editors. As a condition of agreeing to assess the manuscript, all reviewers undertook to keep submitted manuscripts and associated data confidential, and not to redistribute them without permission from the conference committee members.

2. Prom and ProMimport Framework

ProM is a generic framework for implementing process mining tools in a standard environment. The ProM framework receives the input logs in XES or MXML format. Currently, this framework has plug-ins for process mining, analysis, monitoring and conversion. The ProM framework has been developed as a completely plug-able environment. It can be extended by simply adding plug-ins and until now more than 90 plug-ins have been added [4].

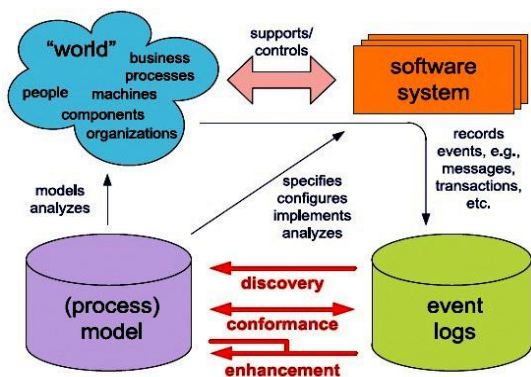


Figure 1. General Process Mining Model (Source: Process Mining: Discovery, Conformance and Enhancement of Business Processes) [5]

Figure 1 illustrates how such plug-ins can be categorized. The plug-ins based on data in the event log only are called discovery plug-ins because they do not use any existing information about deployed models. The plug-ins that check how much the data in the event log matches the prescribed behavior in the deployed models are called conformance plug-ins. Finally, the plug-ins need both a model and its logs to discover information that will enhance this model are called extension plug-ins.

More to the point, ProMimport can be used to import event logs from various systems (e.g. Staffware and FLOWer) such that they can be analyzed using ProM. Figure 2 illustrates the standard MXML format.

The ProcessInstance elements correspond to cases. One ProcessInstance element may hold multiple AuditTrailEntry elements. Each of these elements represents an event. Each AuditTrailEntry element may contain WorkflowModelElement, EventType, Timestamp, and Originator elements. The WorkflowModelElement and EventType are mandatory elements [6].

3. Literature Review

The idea of process mining is not new [7] [8] [9]. Cook and Wolf have investigated similar issues in the context of software engineering processes. In Cook and Wolf, they describe three methods for process discovery: one using neural networks, one using a purely algorithmic approach, and one Markovian approach [7]. The authors consider the latter two the most promising approaches. The purely algorithmic approach builds finite state machine where states are fused if their futures (in terms of possible behavior in the next k steps) are identical.

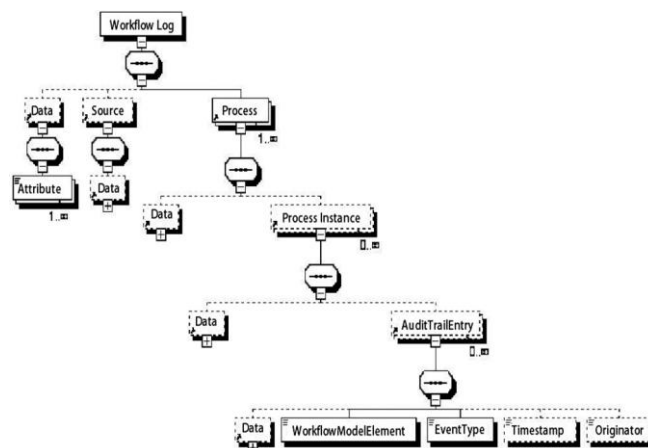


Figure 2. An Event Log (field WorkflowLog) has the Execution of one or more Processes (field Process), and Optional Information about the Source Program that Generated the Log (field Source) and Additional Data Elements (field Data). (Source: Process Mining: Discovery, Conformance and Enhancement of Business Processes) [5]

The Markovian approach uses a mixture of algorithmic and statistical methods and is able to deal with noise. Note that the results presented by Cook are limited to sequential behavior [7]. Cook and Wolf extend their work to concurrent processes [8]. They propose specific metrics (entropy, event type counts, periodicity, and causality) and use these metrics to discover models out of event streams. However, they

do not provide an approach to generate explicit process models.

Cook and Wolf also provide a measure to quantify discrepancies between a process model and the actual behavior as registered using event-based data [9]. The idea of applying process mining in the context of workflow management was first introduced [10]. This work is based on workflow graphs, which are inspired by work flow products such as IBM MQSeries workflow (formerly known as Flowmark) and InConcert.

In Maxeiner, a tool based on these algorithms is presented [11]. Schimm has developed a mining tool suitable for discovering hierarchically structured workflow processes. This requires all splits and joins to be balanced [12] [13]. Herbst and Karagiannis also address the issue of process mining in the context of workflow management using an inductive approach [14] [15]. The work presented is limited to sequential models [15] [16]. The approach described also allows for concurrency [15]. It uses stochastic task graphs as an intermediate representation and it generates a workflow model described in the ADONIS modeling language. In the induction step task nodes are merged and split in order to discover the underlying process. In other sources a heuristic approach using rather simple metrics is used to construct so-called “dependency/frequency tables” and “dependency/frequency graphs” [17] [18]. The preliminary results presented only provide heuristics and focus on issues such as noise [17] [18]. In Van Der Aalst the EMiT tool is presented which uses an extended version of α -algorithm to incorporate timing information [19]. For a detailed description of the α -algorithm and a proof of its correctness we refer to [20].

Sociometry (also referred to as sociography) refers to methods presenting data on interpersonal relationships in graph or matrix form [21] [22] [23]. The term sociometry was coined by Jacob Levy Moreno who conducted the first long-range sociometric study at the New York State Training School for Girls in Hudson, New York [24]. As part of this study, Moreno used sociometric techniques to assign residents to various residential cottages. He found that assignments on the basis of sociometry substantially reduced the number of runaways from the facility. Since the early work of Moreno, sociometry, and SNA in particular, have been active research domains [24]. Workflow management systems like Staffware register the start and completion of activities [25]. ERP systems like SAP log all transactions, e.g., users filling out forms, changing documents, etc. Business-to-business (B2B) systems log the exchange of messages with other parties. Call center packages but also general purpose CRM systems log interactions with customers. These examples show that many

systems have some kind of event log often referred to as “history”, “audit trail”, “transaction file”, etc. [26] [27].

4. Case Study

In this study, we used an event log describing the handling of reviews for the 9th International Conference of ICT & Knowledge Engineering at a private university in Thailand. The original event log was initially in Microsoft Access Data Base format (see Figure 3). The event log consisted of 87 papers (cases) and 3267 events. The conference committee used an international Editorial Board of over 30 academic experts as reviewers. Each paper was sent to three different reviewers. It was not always possible to make a decision after the first round of reviewing and therefore if there were not enough reports, additional reviewers were invited. This process was repeated until a final decision could be made by committee members. Finally, the Chair of the Conference Committee was responsible for making a decision to whether “accept”, “invite another reviewer”, or “reject” the articles (after consulting with other members). Because the event log contained sensitive information concerning the authors’ names, last names, place of the birth, e-mail address and so on, therefore we deliberately anonymized the event log in some necessary parts (such as originators names) before starting any Process Mining analysis. Table I shows some of the techniques used in the case study. These are all related to process mining plug-ins supported by ProM 5.2. However, it should be noted that ProM (both version 5.2 and 6) supports process analysis in the broadest sense.

5. Data Conversion

Before extracting information from the event log, the data should be converted into MXML format because ProM framework receives the input logs only in the XES or MXML formats. In order to convert the Microsoft Access DB to MXML format, four tables with certain structures were filled with data [28] [29]. The data dealt with information about the papers that were reviewed in an international conference in Thailand. Subsequently, the ProMimport tool was used to convert the data from these four tables (see Figure 4) to MXML format. In order to ease filling the four tables with relevant data, a Visual Basic script was written.

The first table “Process_Instances” was filled with the identifier of a certain process instance (field PI-ID) accompanying description (field description). The second table Data_Attributes_Process_Instances was filled with additional information about each process

instance (or data attributes). The third table Audit_Trail_Entries was filled with data about tasks that were performed during the execution of the process instance. The fourth table Data_Attributes_Audit_Trail_Entries was set up in a similar way as table Data_Attributes_Process_Instances.

In order to get the data into four tables, functions addPIandAttr and returnFieldNamesInArray in the Visual Basic script were used (see Figure 3) [28] [29] [40]. These functions were created according to the idea that we have a table (that may also be the result of a query) in which each row contains information about a unique process instance and some additional

information for each process instance. This means that we expected that each row contains information about the identifier of the process instance, the corresponding description, if available, and that the other fields may contain additional information about each process instance (the data attributes for a process instance). As illustrated in Figure 3, the Visual Basic script itself can be found in Modules button by double-clicking on the PMfunctions module. For more information about the functions refer to [28] [29].



Figure 3. Three Screenshots from the Initial Event Log in MS Database Format. Obviously, Tables I and II Contain Data that Should be Filled in the Process_Instances Table and the Data_Attributes_Process_Instances Table. In Order to get the Data into these Tables we used the Functions addPIandAttr and returnFieldNamesInArray in the Visual Basic script

Table 1. Some of the Process Mining Techniques used in the Case Study

Plug-in	Description
Alpha Miner	Discovers a Petri net using the α -algorithm
Heuristic Miner	Discovers a C-net using heuristic mining
Fuzzy Miner	Discovers a fuzzy model using fuzzy mining
Social Network Miner	Creates a social network based on a selected criterion
LTL Checker	Checks a property expressed in terms of LTL
Fitness	Computes fitness of Petri net based on event log
Reply log on flexible model	Conformance checker based on A* algorithm

Sub batch()

'Fill PI and DA_PI tables
`colNames = returnFieldNamesInArray("Table I : Author(s) Info", 1, -1)`
`addPIandAttr "Process_Instances",`
`"Data_Attributes_Process_Instances", "Table I : Author(s) Info", "Case ID", "", colNames, ""`

(1)

'Add Reviewing Process to the ATE table and data attributes for ATE table
`colNames = returnFieldNamesInArray("Table II : Review Process", 6, -1)`
`addATEandAttr "Audit_Trail_Entries",`
`"Data_Attributes_Audit_Trail_Entries", "Table II : Review Process", "", "Case ID", "Activity", "Event Type", "Date & Time", "Resource", colNames, ""`

For providing the parameters cols in function addATEandAttr a reference to an array, another function returnFieldNamesInArray was used, as you see in (1). Accordingly, after configuring an ODBC connection to PC, ProMimport was applied for the extraction of MXML-formatted log (see Figure 4 and Figure 5) from the initial log named Database11.mbd.

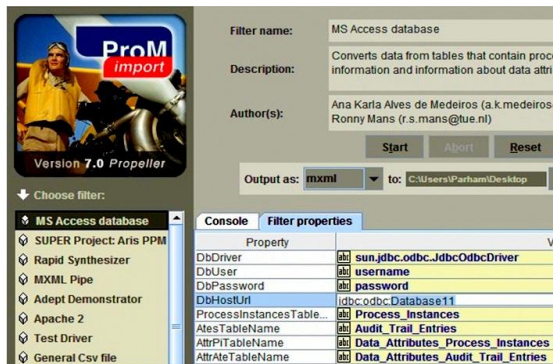


Figure 4. A Screenshot of the Modified Event Log (i.e., Four Tables through VB Script) in the ProM Import Framework

6. Findings and Results

As described earlier, there are many different algorithms for process mining, addressing the problem in different ways, and therefore applicable to different situations. However, process mining algorithms are divided into local and global approaches: (a) Local approaches that build up a process model from the causal relationships between individual tasks, for example Alpha algorithm and HeuristicsMiner. (b) Global approaches that attempt to start by producing the whole model and then refining it, for example the Genetics and Fuzzy algorithms [4].

```

process id="DEFAULT" description="Simulated Process IF for ICT&KE2011"
<ProcessInstance id="1" description="Simulated process instance">
  <AuditTrailEntry>
    <WorkflowModelElement>INVITE REVIEWERS</WorkflowModelElement>
    <EventType>start</EventType>
    <Timestamp>2011-05-31T18:30:00.000+01:00</Timestamp>
    <Originator>Mr. C</Originator>
  </AuditTrailEntry>
  <AuditTrailEntry>
    <WorkflowModelElement>INVITE REVIEWERS</WorkflowModelElement>
    <EventType>complete</EventType>
    <Timestamp>2011-06-06T23:12:00.000+01:00</Timestamp>
    <Originator>Mr. C</Originator>
  </AuditTrailEntry>
  <AuditTrailEntry>
    <Data><Attribute name = "Result by Reviewer B">REJECT PAPER </A
  </Data><WorkflowModelElement>RECEIVE THE SECOND REVIEW </Workf
    <EventType>complete</EventType>
    <Timestamp>2011-06-09T22:10:00.000+01:00</Timestamp>
    <Originator>Ms. B</Originator>
  </AuditTrailEntry>
  <AuditTrailEntry>
    <Data><Attribute name = "Result by Reviewer C">REJECT PAPER </A

```

Figure 5. A Screenshot of the Output (MXML file) Created by ProMimport

In this paper, since the event log contained information about originators, activities, timestamps and even more details, various process mining techniques including Discovery techniques (without a pre-defined model) and Conformance Analysis techniques (with a priori model) were applied.

6.1. Alpha (α) Algorithm

The α -algorithm is an algorithm used in process mining, aimed at reconstructing causality from a set of sequences of events. α -algorithm is defined 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 practical mining technique to identify the routing constructs within the proceedings review system collected from the International Conference of ICT&KE (hosted at a private university) in Thailand.

Since our focus was on the peer-review process as a whole, we based our analysis on the “completed” process instances only. Therefore, our log contained only two process types (i.e., Start and Complete). Filtering the event log (see Figure 6) allowed us to select the type of events (or tasks or audit trail entries) that we wanted to consider mining the log. In Figure 7, general Summary of proceedings review process in the event log is illustrated.

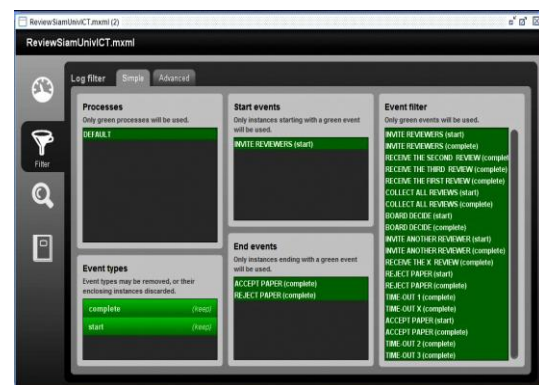


Figure 6. Filtering the Event Log so as to Select only Two Types of Events (Start and Complete)

As mentioned earlier, alpha-algorithm mines the control flow perspective of a process [5] [30] [31]. The control-flow perspective of a process establishes the dependencies among its tasks.

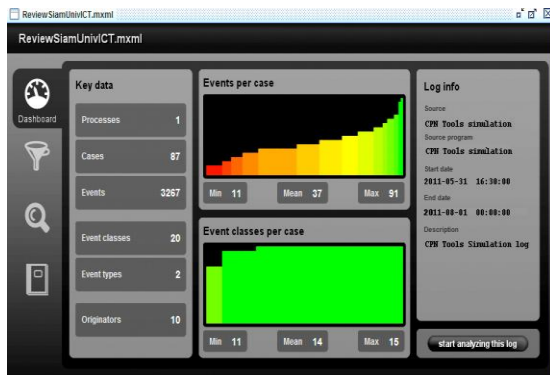


Figure 7. General Summary of Proceedings Review Event Log is Illustrated

As illustrated in Figure 8, Alpha algorithm could extract process knowledge (process models) from the assigned event log. Therefore, now we can interpret the Alpha algorithm results as : How are the papers actually being reviewed? Which tasks proceed with other ones? Are there concurrent tasks? Are there loops? Briefly, what is the process model that summarizes the peer-review flow followed by most/all cases in the log?

6.2. Heuristic Mining

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, α -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 [5]. Therefore, we used Heuristic Miner algorithm in order to utilize frequency being less sensitive to noise and the incompleteness of logs. However, the HM algorithm begins as for α . The log files are examined for causal dependencies (i.e. one task following another). In Figure 9, the square boxes are the tasks; the arcs indicate the dependency between tasks, and the number in the event box indicates the number of times (frequency) the tasks in that box are performed [5] [6] [32]. For instance, in this event log, there are 87 occurrences of the task “Invite Reviewers”. The number on the arcs indicates the number of times the connection is used. For example, numbers 19 and 14 on the “Invite Reviewers” to “Receive The First Review” and “Invite Reviewers” to “Receive The Third Review” arc indicates that the task “Receive The First Review” is 14 times followed by the task “Invite Reviewers”, and the task “Receive The Third Review” is 19 times followed by the task to “Invite Reviewers”. 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.

6.3. Fuzzy Mining

Various real-life applications of the different traditional process mining techniques have shown that the discovered models often look like “spaghetti”, showing all details without highlighting what is important. Indeed, logs may usually include less-relevant or infrequent behavior. Fuzzy mining is a process mining approach to avoid spaghetti-like models that are incomprehensible. In the same way as roadmaps provide suitable abstractions of reality, process models should provide meaningful abstractions of operational processes [4] [5] [6] [33]. Figure 10 (left screen) shows a fuzzy model corresponding to the event log that was used to construct the process model of the proceedings review in ICT&KE conference. This model shows all activities and all causal dependencies and it can be seen as the most detailed map of the process so far. The less frequent or less important activities can be removed. The simplest model would only show the most frequent activity and abstract from all other activities. Figure 10 (right screen) shows an animation based on historic information [39]. This animation shows the actual execution of cases (papers) on top of the discovered model. The animation movie can be played in order to come up with a better understanding of what has occurred in reality. In this way, while watching the movie, a process analyst becomes aware which parts of the model are important and where problems occur.

Replaying an event log can be done by simply passing control from one node to another in the fuzzy model through one of the available arcs. As the net is replayed based on information in the log, no executable semantics are needed. However, while replaying, the nature of splits and joins becomes clear. If control passes from a node A to another node B, then this is visualized through a token that moves along the arc from A to B. When passing along an arc, the token leaves a trail of glowing hot particles as it were a comet. The approach projects all log traces onto the model at once, resulting in multiple traces animated at the same time. The colors of the connections and the thickness of the arcs indicate recent activity. In this way, users can distinguish individual process instances and see the overall activity at a particular point in time. The Fuzzy miner of ProM has many parameters to seamlessly simplify models. For our case study there was no need to simplify the model.

6.4. Analysis of Social Network

In Figure 11, the semantic organizational miner technique discovered groups of reviewers and committee members who work together based on task

similarity. Tasks were considered to be similar whenever they were instances of same concepts.

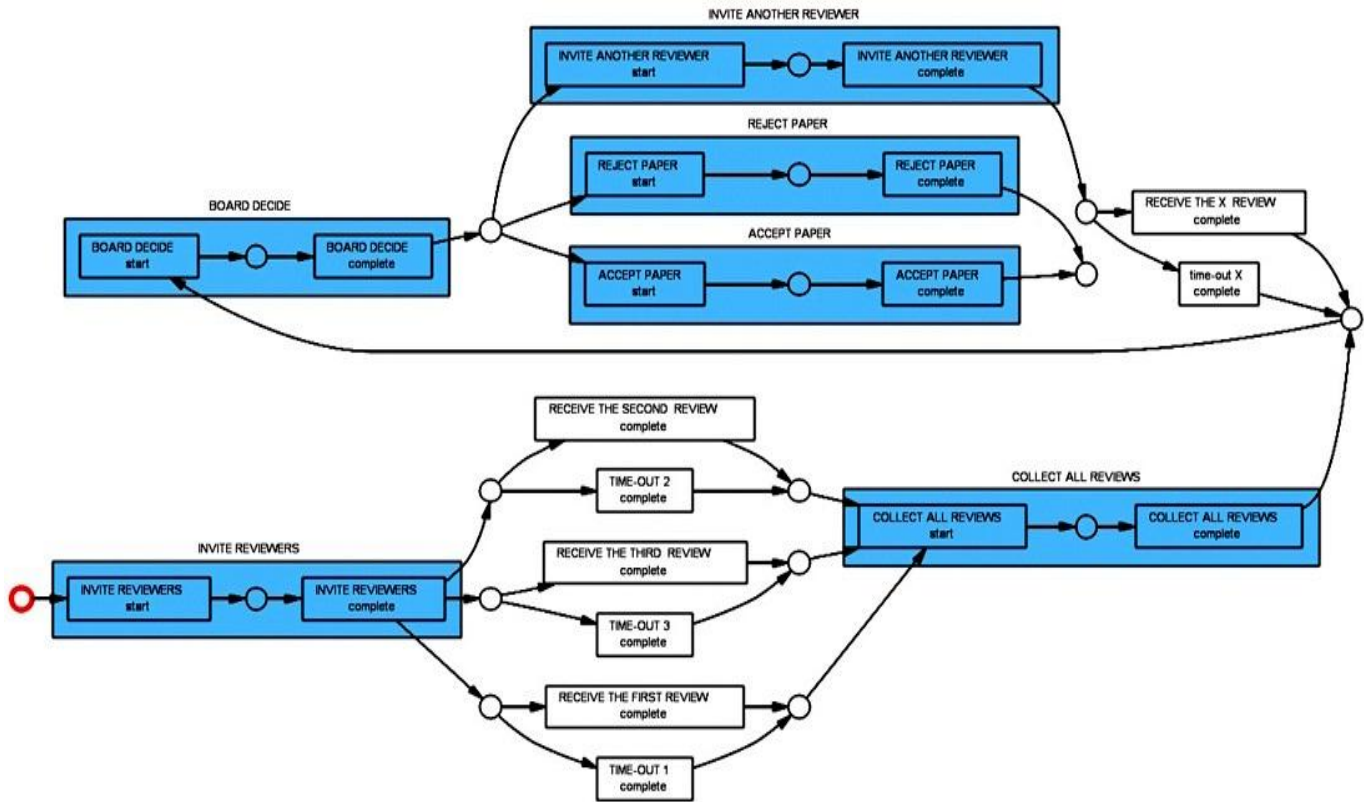


Figure 8. Screenshot of the Mined Event Log Through Alpha-Algorithm. Note that the Red Circle Represents the Beginning of the Peer-Review Process

Results - Heuristics miner on Filtered ReviewSiamUnivICT.mxml (Simple filter)

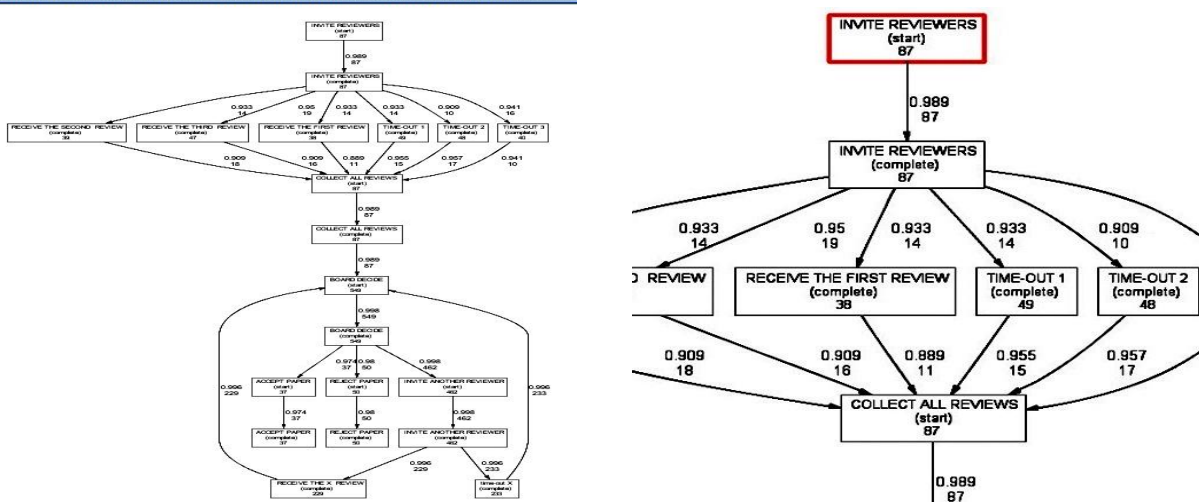


Figure 9. Two Screenshots of the Mined Event Log through Heuristic Mining Technique

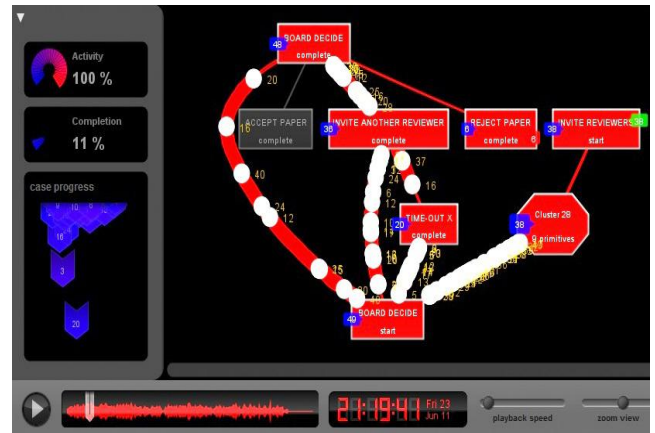
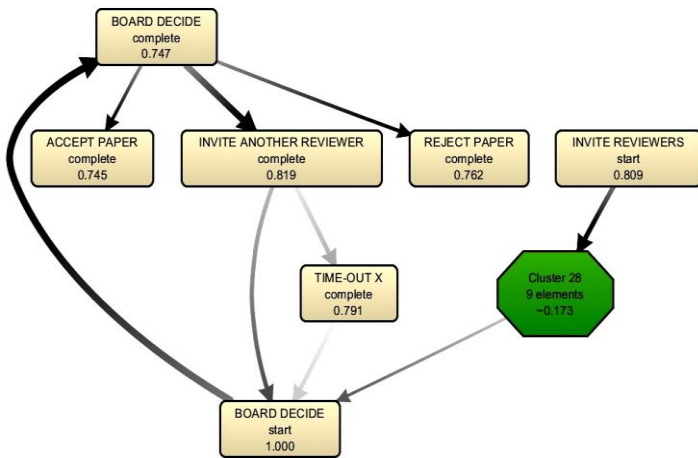


Figure 10. (a) The Left Screen shows a Discovered Fuzzy Model where all Activities are Included at the Highest Level of Abstraction. The Thickness of each Arc is Determined by the Number of Times this Path is Taken (i.e., frequency). (b) The Right Screen shows an Animation Based on the Historic Information (i.e., Unlike the Animations in Simulation Tools, the Animation is Based on the Factual Data). The Animation shows the Actual Execution of Cases (papers) on top of the Discovered Model. Therefore, it is Easy to see where in the Process Bottlenecks Appear and Disappear over time. It is Possible to Follow Concrete cases as well

The work presented in this paper applied the results from sociometry, and Social Network Analysis as well. The paper built a social network in terms of three types of metrics, namely as: (a) Handover of Work metric (this metric determines who passes work to whom.), (b) Working Together metric (this technique counts how frequently individuals work in the same case.), and (c) Similar Tasks metric (this technique determines who performs the same type of activities). The graph in Figure 12 shows which conference committee members hand-over work to other colleagues in the cases (process instances). The oval shape of the nodes in the graph visually expresses the relation between the in and out degree of the connections (arrows) between these nodes. A higher proportion of in-going arcs lead to more vertical oval shapes while higher proportions of outgoing arcs produce more horizontal oval shapes. As a result, Mr. A and Mr. B have been the most hardworking members of the proceedings review process because very few arrows go out of these two employees and only arrows coming to them. This means that they have gotten a big burden of work handed-over from other members to them [5] [6] [34].

Similarly, in Figure 13, within each case (i.e., process instance) there is a handover of work from individual i to individual j if there are two subsequent activities where the first is completed by i and the second by j . We used knowledge of the process structure to detect whether there is really a causal dependency between both activities or not. It was also possible to not only consider direct succession but also indirect succession using a “causality fall factor” β . The main idea was to count the number of times

individual j executed an activity in-between two activities executed by individual i . This may indicate that work was subcontracted from i to j . Our main goal from the beginning was to identify groups/communities of originators in the social network. We must be sure that people in the same community work together. To address this issue, we need to focus on the cases and not in the activities. Figure 14 shows a social network obtained by Working Together Algorithm from ProM. This algorithm identifies one distinct groups, but we only get one distinct groups because all the people in log file work in disjoint cases. The log file has not a single case where the same person belongs to two different teams. The algorithm Working Together is only helpful when there are disjoint teams from the beginning (in the log) [6] [34] [35].

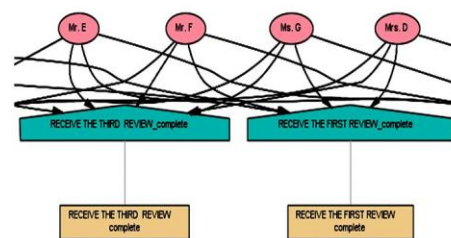


Figure 11. The Screenshot shows the Result of Applying the Semantic Organizational Mining Technique on the Event Log

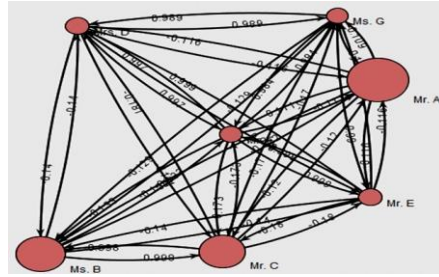


Figure 12. Screenshot of the Social Network Analyzer in the event log

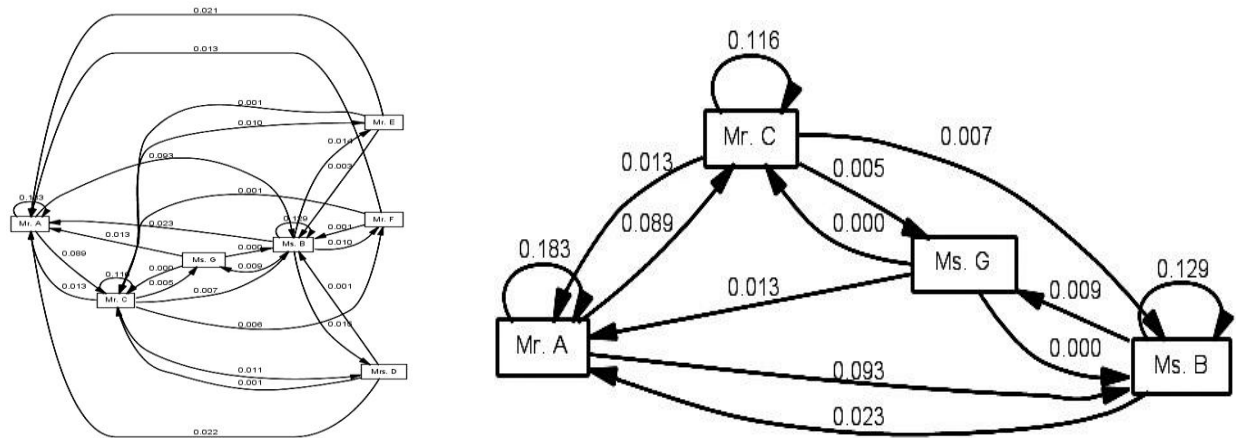


Figure 13. Two Screenshots of the “Handover of Work” Metric through Social Network Analyzer Technique

Finally, similar task metric (see Figure 15) does not consider how individuals work together on shared cases but focuses on the activities they do. The assumption here is that people doing similar things have stronger relations than people doing completely different things. Each individual has a “profile” based on how frequent they conduct specific activities. There are many ways to measure the “distance” between two profiles thus enabling many metrics. There are four kinds of distance metrics. Euclidean distance is the “ordinary” distance between two points that one would measure with a ruler. Pearson’s correlation coefficient is frequently used to find the relationship among cases. Similarity coefficient is a statistic used for comparing the similarity and diversity of sample sets. Hamming distance does not consider the absolute frequency but only whether it is 0 or not. It considers the type of event. Thus far we assumed that events correspond to the execution of activities.

On the contrary of the five previous techniques (Organizational Mining, Social Network Analyzer, Handover of Work, Working Together, and Similar Task), Role Hierarchy Miner technique derives a hierarchical model. This technique implements the Agglomerative Hierarchical Clustering technique based on joint activities. It means that the clusters are

determined according the activities that each originator performs. Figure 16 shows the dendrogram derived from this technique. Through the dendrogram, this technique allowed us to derive flat or disjoint organizational entities by cutting the dendrogram with a certain value. As you see in Figure 16, by cutting the dendrogram using a threshold of value 0, we obtained four clusters. From the generated tree, we can conclude that there are four different groups. The first group, group 1, includes Mr. A only. The second group, group 2, consists mainly of the Mr. E, Mr. F, Mrs. D, and Ms. G. The third group, group 3, was consisted of Ms. B. The fourth group, group 4, includes Mr. C. Therefore, organizational Miner technique was used at a higher level of abstraction than the previous techniques. While the Social Network Miner worked at the level of the individual, the Organizational Miner technique worked at the level of teams, groups or departments. Consequently, Role Hierarchy Miner technique could generate a role hierarchy based on the different activities performed by actors [4] [5] [6] [34] [35].

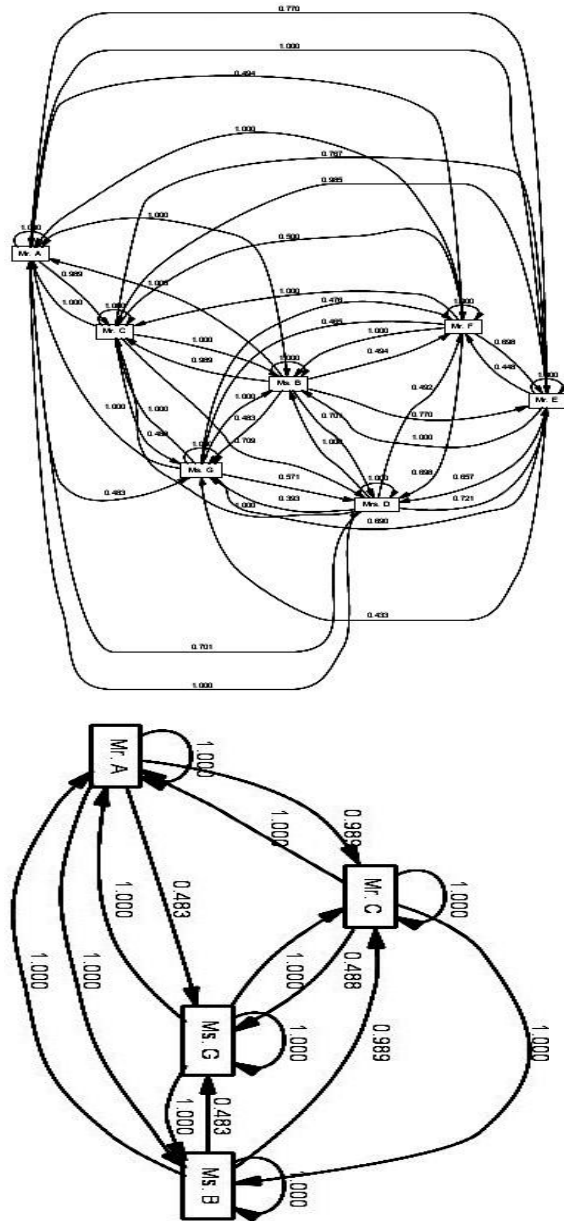


Figure 14. Two Screenshots of the “Working Together” Metric through Social Network Analyzer Technique

6.5. LTL Checker approach

It is often the case that processes in reviewing process should obey certain rules or principles. One common example is the “Four-Eyes Principle” which determines that a same committee member cannot execute certain tasks. One way to check if these rules are indeed being obeyed is to audit the log. In ProM, auditing of a log is provided by the analysis plug-in Semantic LTL Checker Plug-in. We used LTL Checker

Plug-in in order to verify the property: Does the task “Invite Another Reviewer” always happen after the tasks “Board Decide” and “Collect All Reviewers” (i.e., C after B after A) or not? (see Figure 17) The resulting screen showed the log split into two parts: one with the cases that satisfy the property (or formula) and another with the cases that do not satisfy the property. We applied the LTL plug-in over the correct process instances and we found out that in 80 of the process instances out of the total of 87, “Invite another Reviewer” has happened after “Board Decide” and after “Collect All Reviews”. Only in 7 process instances, C didn’t happen after B and A eventually [4] [5] [6].

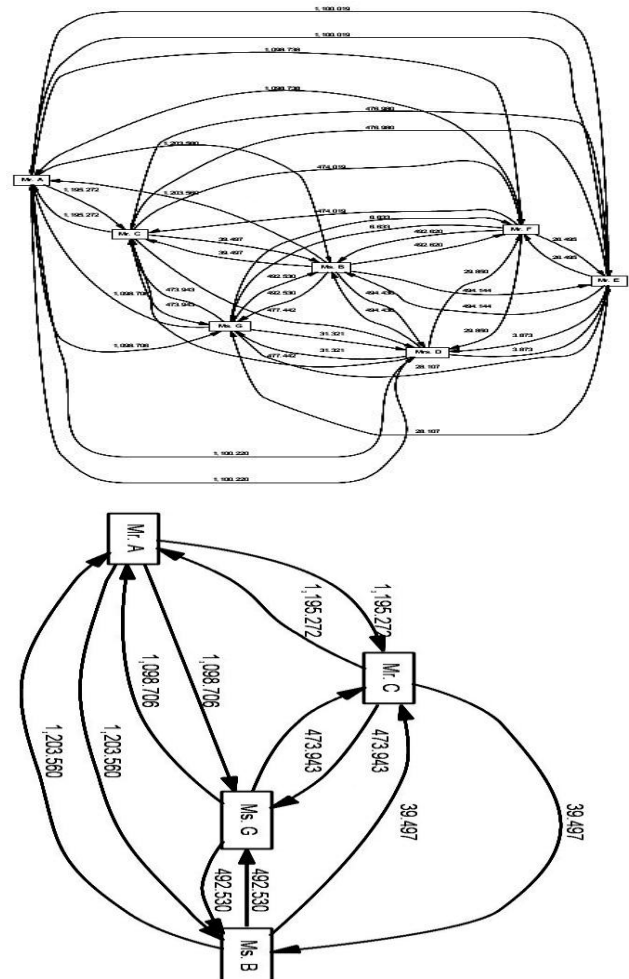


Figure 15. Two Screenshots of the “Similar Task” Metric through Social Network Analyzer Technique (Threshold zero)

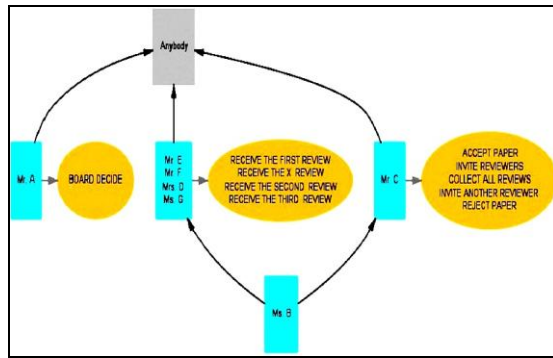


Figure 16. A Screenshot of the “Role Hierarchy Miner” in the Event Log

In Figure 16, a directed arrow between two actors/groups indicates that the actor/group at the base of the arrow can do at least the activities performed by the actor/group at the arrow head.

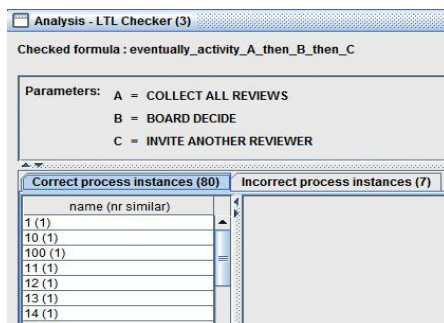


Figure 17. A screenshot of the Default LTL Checker technique is illustrated. The technique checks whether activity A has occurred (sequentially) before activity B—and before activity C—or not. Results show that only 80 cases (out of total 87) followed the formula “eventually_activity_A_then_B_then_C”. This means that reviewers often did not respond. As a result, it was not always possible to make a decision after the first round of reviewing (with three reviewers) and therefore additional reviewers were invited in majority of the cases. Understandably, the process has been repeated until a final decision could be made (accept or reject)

6.6. Conformance Checker & Performance Analysis

As mentioned earlier, when calculating the different performance metrics, all process instances in the input log are replayed in the input Petri net. When doing this, it can happen that a process instance does not fit in the Petri net.

The “model” perspective 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).

On the other side, it was so interesting to check how much process instances in a log match a model highlighting discrepancies through “log diagnostic” perspective. As an illustration, we checked the exported mined model for the log of the running case study against a new log provided by the Graduate School of Information Technology (Conference Organizer Committee) at a private university. Our aim was to check how compliant the real-life log is with the pre-defined model. Figure 18 and Figure 20, respectively, show different screenshots of the “model” and “log diagnostic” perspectives of the Conformance Checker plug-in. These two perspectives provide detailed information about the problems encountered during the log replay.

6.7. Model Perspective

The following metric is calculated from a model perspective in order to measure the degree of fitness. The token-based fitness measure f relates the amount of missing tokens with the amount of consumed ones and the amount of remaining tokens with the amount of produced ones. So, if the log can be replayed correctly, i.e., there were no tokens missing nor remaining, it evaluates to 1. In the worst case, every produced and consumed token is remaining or missing, the metric evaluates to 0.

There are a number of options, which can be used to enhance the visualization of the process model by the indication of [6]:

- 1) *Token Counter*. Visualizes the missing and remaining tokens during log replay for each place. This allows localizing those parts in the model where the mismatch took place, if any.
- 2) *Failed Tasks*. Visualizes the transitions that were not enabled (that is, not ready) during log replay and therefore could not be successfully executed.
- 3) *Remaining Tasks*. Visualizes the transitions that remained enabled after log replay, which indicates non-proper completion of the specified process, and hints that this task should have been executed.
- 4) *Path Coverage*. Visualizes the transitions that were executed during log replay (regardless of whether that happened successfully or had to be enforced creating the missing tokens). This enables to

follow the path a particular log trace, or a set of log traces, has been followed within the model.

5) *Passed Edges*. Indicates at each of the edges how often it was followed during the replay of the given process instances.

Here, the central panel in Figure 18 (the one with the Petri net model) shows (i) the bottlenecks (notice the different colors for the places) and (ii) the routing probabilities for each split/join tasks. Figure 19 shows information about the waiting times in reviewing process of proceedings of international conference in Thailand. Accordingly, waiting time in regard to receiving feedback from the reviewers is

classified as High, Medium and Low. This means that if a reviewer sends his/her feedback in less than 7 days, it would be considered as “Low” response time identified in blue color. However, if the reviewers send his/her feedback within 7 to 21 days, it would be considered as “Medium” response time identified in yellow color. Finally, if the reviewers send his/her feedback in more than 21 days, it would be considered as “High” response time identified in red/critical color.

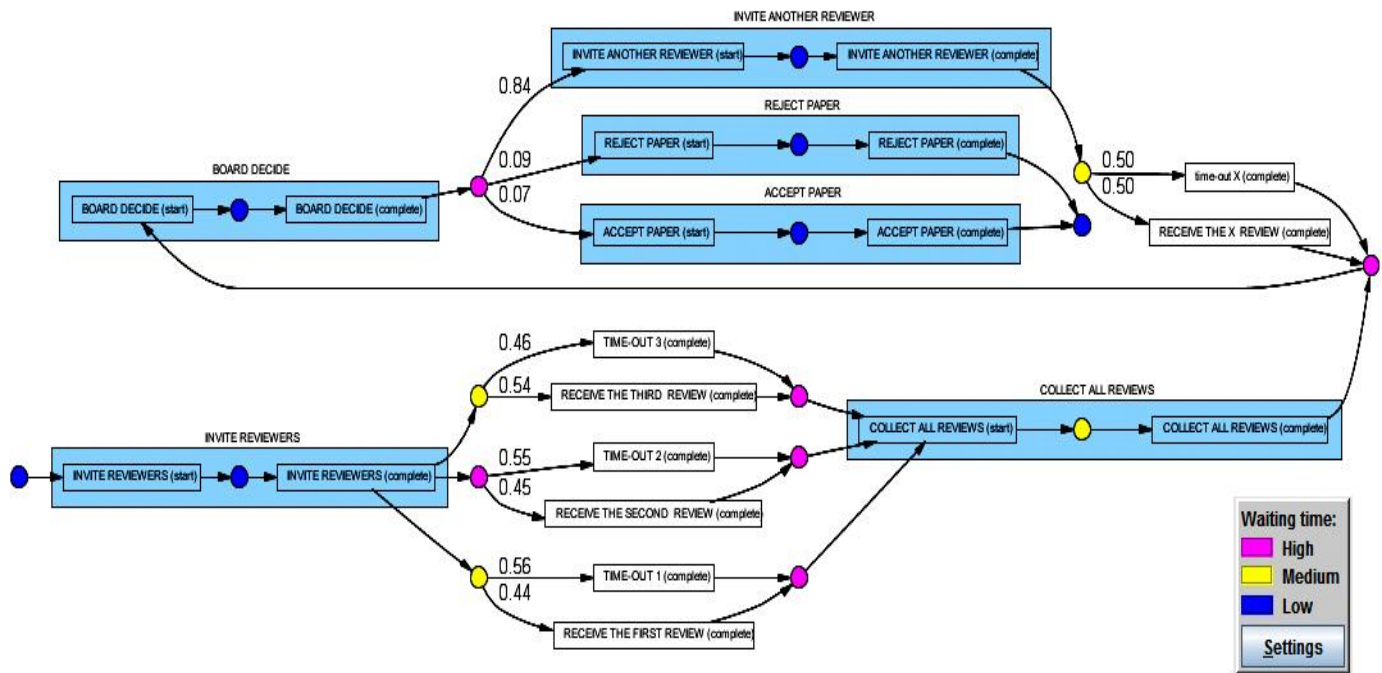


Figure 18. Screenshot of the analysis plug-in Conformance Checker (Model view). Using conformance checking techniques models can also be compared with reality and existing models can be enhanced with additional information, e.g., indicating bottlenecks in a process. Model view shows places in the model where problems occurred during the log replay

Referring to Figure 18, we consider that only in 55% of the cases the response (the review report) from the “Second Reviewer” has timeout collecting no feedback from the reviewer. This means that the reviewing path to collect feedback from the “Second Reviewer” has taken too much time on average (critical path) and need to be fixed for the prospective conferences in future.

6.8. Log Perspective

The diagnostic perspective can be changed to visualize the log file or a subset of log traces respectively. The following metrics are calculated from

a log perspective in order to measure the degree of fitness:

1) *Successful Execution*. The fraction of successfully executed process instances (taking the number of occurrences per trace into account).

2) *Proper Completion*. The fraction of properly completed process instances (taking the number of occurrences per trace into account).

There is one option, which can be used to enhance the visualization of the event log by the indication of:

3) *Failed Log Events*. Visualizes the log events that could not be replayed correctly, which corresponds to the failed tasks in the model view [4] [5] [6] [36] [37].

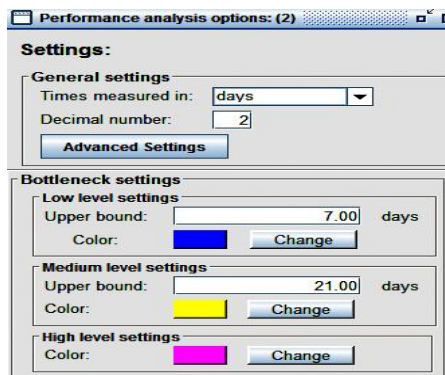


Figure 19. The Bottom Panel Shows Information about the Waiting Times in Reviewing Process of Proceedings

In Figure 20, the log perspective indicates the points of non-compliant behavior (highlighted in orange color) for every case between the real and pre-defined (priori) logs [38]. Additionally, analyzing the gap between the proceedings' review model and the real event log helped us to detect violations (i.e., rectangles in orange color) in the proceedings review process of the 9th International Conference of ICT&KE at a private university in Thailand.

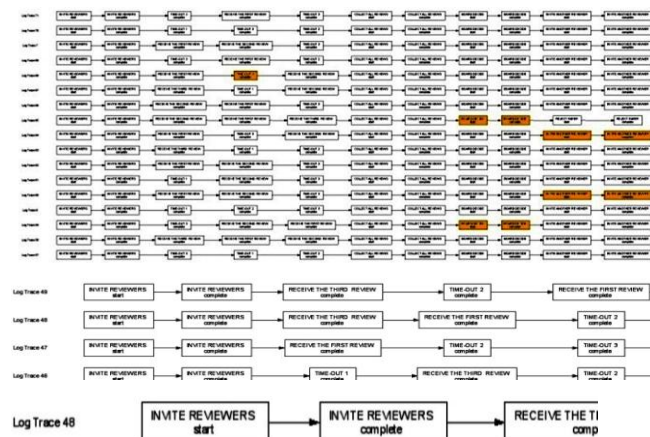


Figure 20. Three Screenshots of the Conformance Checker Technique (Log view). Log View Shows where Reply Problems Occurred in the Log

7. Conclusion

In this study, we used an event log describing the handling of reviews for proceedings of an International ICT & KE Conference in Thailand. The original event log was initially in Microsoft Access DB format. The event log consisted of 87 cases (papers) and 3267 events. We knew that ProM receives the input logs only in XES or MXML formats. Therefore, we used the functions `addPIandAttr` and `returnFieldNamesInArray` (in Visual Basic script) to create four tables with certain structures. These

functions were created according to the idea that we have a table in which each row contains information about a unique process instance and some additional information for each process instance. Later, ProMimport was chosen as a framework to extract MXML log from the data source. Since the event log contained private information concerning the authors, the event log was deliberately anonymized in some necessary parts. Next, an ODBC connection was configured between the data and PC. After successfully converting the data into MXML, we applied various process mining techniques in order to extract information from the event log. While Alpha-algorithm was the first process discovery technique used in the study, Heuristic Miner technique indicated the dependency between the proceedings review tasks. We applied ideas obtained from cartography to build process-model maps through Fuzzy Miner, as well. The result was an animation movie that came up with a better understanding of what has occurred in peer-review processes. On average, all accepted articles were reviewed by 3.9 experts.

Into the bargain, three metrics were exerted with the purpose of building graphical social network models in terms of: (a) Handover of Work, (b) Working Together, and (c) Similar Tasks. The main goal, here, was to identify different groups and communities of originators (clusters) through Social Network Miner techniques. Consequently, Conformance Checker and LTL approaches could assess the fitness of the model detecting discrepancies between the flows. Considering the results of the study, conference committee chairs can better evaluate the performance of the involving reviewers (as well as team members) within the assigned tasks. This will improve the performance, efficiency and effectiveness of the handling of reviews for prospective academic/educational conferences.

8. References

- [1] Guide to Review and Publication Policies. (2012). [Online]. Available: <http://www.nature.com/authors/gta.pdf>
- [2] "Proceedings 2011 Ninth International Conference on ICT and Knowledge Engineering". A private university, Bangkok, Thailand, ISSN: 2157-09812012. <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=06152422>
- [3] S. seema, "Ict & KE conference", Siam University, Bangkok, Thailand, 2012. [Online]. Available: http://www.siam.edu/siamedu_thai_mainpage/index.php?option=com_content&view=article&id=273:ict-ake-2012&catid=53:2010-08-27-04-59-29&Itemid=191

- [4] 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/prom/start>
- [5] W.M.P. Van Der Aalst. *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Springer-Verlag, Berlin, 2011.
- [6] 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=cache
- [7] 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>
- [8] 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.
- [9] 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>
- [10] 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>
- [11] M.K. Maxeiner, K. Kuuspert, and F. Leymann. (2001). Data Mining von Workflow-Protokollen zur teilautomatisierten Konstruktion von Prozessmodellen. *Informatik Aktuell Springer*. Berlin, Germany, pp. 75–84.
- [12] 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
- [13] 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>
- [14] J. Herbst, “A Machine Learning Approach to Workflow Management”, in *Proceedings 11th European Conference on Machine Learning*, Berlin, 2000, pp. 183–194.
- [15] 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.
- [16] 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.
- [17] 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>
- [18] 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>
- [19] 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
- [20] 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
- [21] 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.
- [22] J. Scott, and P. J. Carrington. (2011). Sage Handbook of Social Network Analysis. SAGE Publications Ltd. London. UK. [Online]. Available: http://arts.uwaterloo.ca/~pj/pubs/Sage_Hbook/Sage%20Chap17.pdf
- [23] S. Wasserman, and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, Cambridge, 1994.
- [24] 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.
- [25] W.M.P. van der Aalst, and K.M. van Hee. *Workflow Management: Models, Methods, and Systems*. MIT press, Cambridge, MA, 2002.

- [26] 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>
- [27] 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>
- [28] 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, pp. 205-212.
- [29] 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. Technische Universiteit, Eindhoven, The Netherlands [Online]. Available: <http://www.processmining.org/promimport/tutorials>
- [30] 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.
- [31] 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>
- [32] A.J.M.M. (Ton) Weijters and J.T.S. (Joel) Ribeiro. (2011). Heuristics Miner 6.0: Users Guide. Eindhoven University of Technology, Eindhoven, The Netherlands. [Online]. Available: <https://svn.win.tue.nl/trac/prom/UserGuideHeuristicsMining6.0.doc>
- [33] B.F. van Dongen, and A. Adriansyah, “Process mining: fuzzy clustering and performance visualization”. *Springer*, vol. 43, pp. 158-169, 2010.(Business Process Management Workshops , Lecture Notes in Business Information Processing)
- [34] P. Porouhan, W. Premchaiswadi, J. Reungrungsee, and S. Intarasema, “Prom: Analysis of Social Network in Students’ Registration System”. *Proceedings of the IEEE ICT & Knowledge Engineering*, Bangkok, Thailand, 2012, pp. 236- 243.
- [35] C. S. D. Costa Alves, “Social Network Analysis for Business Process Discovery”, M.S. thesis, Universidade Tecnica de Lisboa, Lisboa, Portugal, 2010. [Online]. Available: <https://dspace.ist.utl.pt/bitstream/2295/751982/1/Claudia-Alves-55815-DissertacaoFinal.pdf>
- [36] P. Hornix, “Performance Analysis of Business Processes Through Process Mining”, M.S. thesis, Technische Universiteit Eindhoven, Eindhoven, Netherlands, 2007. [Online]. Available: http://www.processmining.org/blogs/pub2007/performance_analysis_of_business_processes_through_process_mining
- [37] J. Muñoz-Gama, “Algorithms for Process Conformance and Process Refinement”, M.S. thesis, Universidade Politecnica de Catalunya, Barcelona, Spain, 2010. [Online]. Available: <http://www.lsi.upc.edu/~jmunoz/pub/MASTER10.pdf>
- [38] S. Driessen, “Finding Process Bottlenecks from Logged Process Data Using Data Mining”, M.S. thesis, Technische Universiteit Eindhoven, Eindhoven , The Netherlands, 2007. [Online]. Available: http://www.processmining.org/blogs/pub2006/process_bottleneck_mining_finding_process_bottlenecks_from_logged_process_data_using_data_mining
- [39] A. Rozinat. (2010). How to Get Started With Process Mining in ProM. [Online]. Available: <http://fluxicon.com/blog/2010/11/how-to-get-started-in-prom/>
- [40] 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. (2005). The ProM framework: A new era in process mining tool support. Presented at the 26th International Conference on Applications and Theory of Petri Nets (ICATPN 2005). [Online]. Available: <http://www.win.tue.nl/~hverbeek/downloads/preprints/Dongen05.pdf>