A Process Mining Approach in Software Development and Testing Process: A Case Study

Rabia Saylam, Ozgur Koray Sahingoz

***Abstract*——Process mining is a relatively new and emerging research area dealing with the process modeling and process analysis by studying on the event logs. By mining these logs, one can understand what is actually happening in the organization, which will bring quite different results than what people think. That is the reason why big organizations start using process mining to x-ray their organizations by various industrial and scientific process mining applications. This study aims to enlighten the researchers about the details of process mining and ProM framework. To accomplish this, firstly, the details and the conversion process of MXML format, which is the correct format applicable for ProM tool, are explained. After that, some main process mining algorithms are detailed by analyzing Alpha Algorithm, Heuristic Mining Algorithm and Social Network Algorithm. Finally, how to extract data through Software Development cycle process is presented, and the results, which are obtained by analyzing the related data, are depicted.**

***Index Terms*—Process Mining, ProM, Process Mining Algorithms, Alpha Algorithm, Heuristic Miner Algorithm, Data Collection.**

I. Introduction

Business process mining, or process mining in a short form, is an emerging research area, which brings a new way of analyzing and aims to improve the business processes by presenting the big picture to decision makers by using event logs. Due to this property, some researchers defined process mining as a kind of machine learning task [1]. The main idea is to learn (or deduct) some critical knowledge from these logs, which are recorded by the analyzed information system.

Mainly, event logs contain the necessary information about events in the analyzed system by referring to an activity or a case. *Enterprise Resource Planning* systems like *SAP* would be accepted as a good sample for process mining since it logs all transactions that cover the people and procedures. On the other hand, *Customer Relationship Management* systems log main interaction with their customers. If these types of processes cannot be mined well enough, it will be very difficult the catch where the bottlenecks or skipped sub-processes are. Therefore by using process mining techniques, the possibility of making a mistake is reduced and the chance of catching the opportunities is increased.

To enhance the efficiency of the mining system, mining algorithms are accepted as a key aspect in process mining, due to their direct impacts on mining results. There are many algorithms that can be plugged in as separate tools to the Process Mining Platforms. ProM is accepted as the largest platform, which covers all mining properties in one system.

Manuscript received March 14, 2014 revised April 10, 2014

R. Saylam, Department of Computer Engineering, Turkish Air Force Academy, Istanbul, Turkey, rsaylam@hvkk.tsk.tr

O.K. Sahingoz, Department of Computer Engineering, Turkish Air Force Academy, Istanbul, Turkey, sahingoz@hho.edu.tr

While looking from the software engineering view, software development process is very important in conducting software product, and it is a predefined ordering of some activities to develop these products. Both in the implementation and the maintenance phases, these products, are conducted by extracting software development and testing cycle, which processes event logs in the organization. Therefore, collecting these data is very important in process mining. After collecting them, they must be carefully analyzed. Therefore, firstly, the applicable tables, which include the correct data, are extracted from the big data garbage, then these log data are mined by using mining algorithms, and finally their results are evaluated by using four quality criteria as fitness, simplicity, precise and generalize [2].

In this paper, we describe a case study based on the log of software development process in a software company. The office is responsible for the construction and maintenance of the software, e.g., Financial Accounting (FI), Material Management (MM), Quality Management (QM), Project Systems (PS) and Human Resources (HR), of an organization over 30,000 personnel. We have used an event log containing 71 cases as a starting point for mining the process perspective. And we tried to compare the mining algorithms according our event data, and presented the result.

This is how the paper is structured: Section 2 introduces some background information about Process Mining, the ProM Framework and Process Mining Algorithms, Section 3 details a process mining case study about Software Development and Testing, and finally Section 4 concludes the study.

II. Background

A. Process Mining

Process mining is a relatively new and emerging research area dealing with the process modeling and process analysis, as well as business intelligence and data mining. The main purpose of process analyzing is to identify the processes by mining the event logs. There are two reasons that prove the usefulness of the mining process. First, it is used as a tool that provides information about how people and procedures really work. For example, SAP would be a good sample for this since it logs all transactions that cover the people and procedures. Second, process mining is a useful tool to compare predefined processes and the actual process.

An *Event* can be defined as an activity corresponding to the starting point of the process mining. Process mining techniques need a sequential relation between events. Each activity is a unique process instance, in other words it belongs to a specific event. Also, additional information such as the source initiating or carrying out an activity (a person or a device), the occurrence and ending time of events (may be

ISBN: 978-988-19252-7-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

activity-based), or data elements recorded with the incident (such as the size of an order), which is required in order to create a realistic model [3].

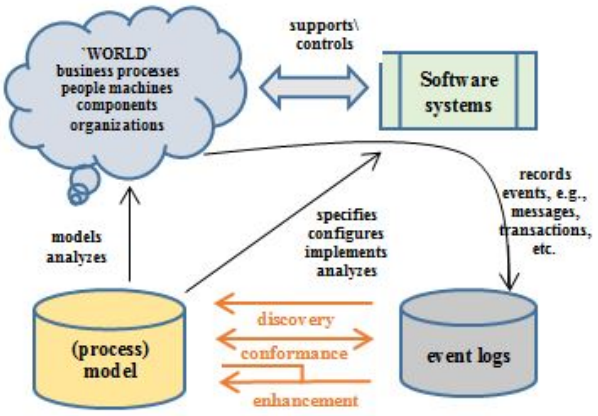

Fig. 1. Types of Process Mining [3]

According to the Figure 1, which is created by IEEE working group, there are three types of mining processes. These are respectively; **Discovery**, which deals with the generation of the model from event logs without using any meta-information; **Conformance Techniques**, which deal with comparison of a priori model with the model of event logs and aiming at detecting inconsistencies and/or deviations between the process models from the log files; and lastly **Enhancement**, which aims to extend or to improve the existing process model according to gather information from the event log [4].

B. The ProM Framework

ProM [2] is the most common and popular process mining tool. There are many algorithms that can be plugged in as separate tools, and ProM is accepted as the largest platform, which covers all properties in one system. These separate tools aim different goals such as exploring processes, analyzing social networks or validating the business rules [3]. This tool is open source and extensible. In other words, it can be improved by creating new *Plug-ins*. Up to the present day, over 280 Plug-ins are included in this tool. The most relevant and important *Plug-ins* are the ones which deal with mining processes. Figure 2 summarizes the general ProM architecture by demonstrating the relationship between the event logs and *Plug-ins*.

The event log, which is used as the input for the plug-ins is often in *Mining XML (MXML)* format. This format is based on XML standards and specially designed for ProM. Information Systems such as SAP have their own logging format. When an event log is required for mining, firstly this event log format is needed to be converted into a format supported by process mining. Therefore, the first step is cumbersome since the information about the format which is supported by the process mining tool has to be known besides the current information system format. To make this type of operations easier, ProM developers have created MXML. MXML follows a specific schema definition and indicates that the event log does not constitute irregular and random information, so there is known the location of items for the need of a plug-in. Figure 3 shows a snapshot of an MXML log.

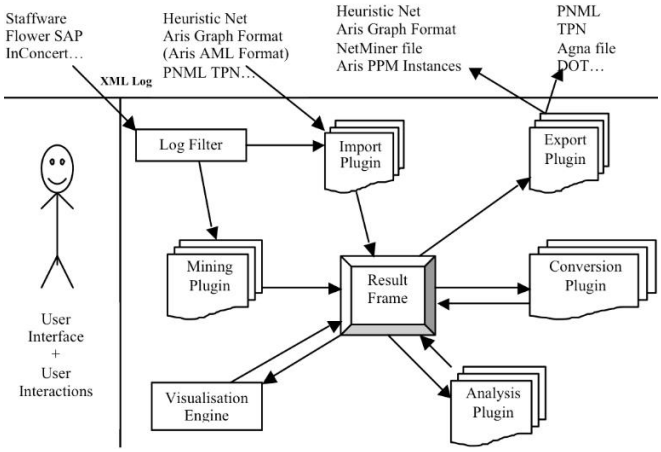

Fig. 2. ProM Architecture

Process log starts with WorkflowLog element which contains *Source* and *Process* elements. Source element means information about the event-logging system or software. Process element represents the process to which process log belongs. Process element consists of a variety of audit trail inputs. Here, audit trail represents an atomic event and recording information such as WorkflowModelElement, EventType, Timestamp, and Originator elements [5].

```
<?xml version="1.0" encoding="UTF-8"?>

<WorkflowLog xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="WorkflowLog.xsd"
description="Test log for decision miner">

        <Source program="name: , desc: , data: {program=none}">
                <Data>
                        <Attribute name="program">name: , desc: , data: {program=none}</Attribute>
                </Data>
        </Source>
        <Process id="0" description="">
                <ProcessInstance id="Case 4" description="">
                        <AuditTrailEntry>
                                <WorkflowModelElement>Register Claim</WorkflowModelElement>
                                <EventType>start</EventType>
                                <Timestamp>2002-04-08T09:52:00.000+01:00</Timestamp>
                                <Originator>Robert</Originator>
                        </AuditTrailEntry>
                        <AuditTrailEntry>
                                <Data>
                                        <Attribute name="Amount">500</Attribute>
                                        <Attribute name="CustomerID">C568120443</Attribute>
                                        <Attribute name="PolicyType">Normal</Attribute>
                                </Data>
                                <WorkflowModelElement>Register Claim</WorkflowModelElement>
                                <EventType>complete</EventType>
                                <Timestamp>2002-04-08T10:11:00.000+01:00</Timestamp>
                                <Originator>Robert</Originator>
                        </AuditTrailEntry>
                        <AuditTrailEntry>
                                <WorkflowModelElement>Check policy only</WorkflowModelElement>
                                <EventType>start</EventType>
                                <Timestamp>2002-04-08T10:32:00.000+01:00</Timestamp>
                                <Originator>Mona</Originator>
                        </AuditTrailEntry>
                        <AuditTrailEntry>
                                <WorkflowModelElement>Check policy only</WorkflowModelElement>
                                <EventType>complete</EventType>
                                <Timestamp>2002-04-08T10:59:00.000+01:00</Timestamp>
                                <Originator>Mona</Originator>
                        </AuditTrailEntry>
                        <AuditTrailEntry>
                                <WorkflowModelElement>Evaluate claim</WorkflowModelElement>
                                <EventType>start</EventType>
                                <Timestamp>2002-04-08T11:22:00.000+01:00</Timestamp>
                                <Originator>Linda</Originator>
                        </AuditTrailEntry>
                        <AuditTrailEntry>
                                <Data>
                                        <Attribute name="Status">approved</Attribute>
                                </Data>
                                <WorkflowModelElement>Evaluate claim</WorkflowModelElement>
                                <EventType>complete</EventType>
                                <Timestamp>2002-04-08T11:47:00.000+01:00</Timestamp>
                                <Originator>Linda</Originator>
                        </AuditTrailEntry>
                </ProssInstance>

                 ....

        </Process>
</WorkflowLog>
```

Fig. 3. Snapshot of an MXML log

C. Process Mining Algorithms

The core component of the process mining is its algorithm (process mining algorithm). It mainly determines how these

ISBN: 978-988-19252-7-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

models are produced. In the literature, there are many mining algorithms. In the following part, some of the important ones are detailed.

1) Alpha Algorithm: The Alpha algorithm is the milestone among process discovery algorithms that could deal with concurrency. It provides a good insight for the process mining world. However, it has some problems with noise and frequency, and its results are not very world-realistic [2].

2) Heuristic Miner Algorithm: Heuristic Miner (HM) algorithm focuses on control flow perspective and creates a process model in Heuristics Nets format for a given event log. Moreover, this algorithm uses frequency information, which solves the noise problem by expressing the number of connections between different tasks in the event log [6]. The basic feature of the HM is its robustness for incompleteness and noise. Because HM is based on the frequency patterns, it lets the user stay on the main behavior of event logs [7].

3) Social Network Analysis: Sociometry is a presentation method in graphical and matrix form referring to the data relating to the interpersonal relations. Sociometry term was revealed by Jacob Levy Moreno whose studies took place in 1932-1938. He used sociometric techniques for assigning neighborhood residents to various residential cottages as part of his studies. As a result of these sociometry-based assignments, it is proven that the number of fugitives has been reduced significantly. Since then, a great number of sociometry-based studies have been conducted. This study is based on evaluation surveys, in other words sociometric tests [8]. The data relating to the interpersonal relations is also hidden is the event logs. At this point, Social Network miner tool of process mining is used to reveal this relation. The focus point in the process of extracting a process model from an event log is the various process activities and their dependencies. Besides, the focus in the process of forming the roles and organizational entities will be the relation between the people and the processes. In other words, the main focus is on Sociometric Relations or Social Network [9].

Nodes correspond to organizational entities in a social network. It is also possible that the nodes may refer to roles, groups or sections. Arrows in a social network refer to the relationship between these organizational entities. Arrows or nodes may have their weights, which indicate the level of importance or frequency of them.

Sometimes, the term Distance is used as the opposite of weight. If the distance between two organizational entities is small, the weight of the arrow connecting these two entities is high. If the distance is large, then the arrow weight will be small. In certain cases, this arrow is not shown in social network. Numbers indicate the average number of jobs transferred from one source to another. These roles can be revealed by inspecting frequency pattern [2].

III. CASE STUDY: PROCESS MINING IN SOFTWARE DEVELOPMENT AND TESTING PROCESS

In this section, we explore the 71 cases of software development company, who is responsible for software of Financial Accounting (FI), Material Management (MM), Quality Management (QM), Project Systems (PS) and Human Resources (HR), for an organization which has more than 30,000 personnel. Application is conducted by extracting Software Development and Testing cycle process event logs in an organization using SAP (Systems, Applications and Products in Data Processing). Development and testing process is followed through SAP Solution Manager (SOLMAN) platform. After defining the requirements (SRS-Software Requirement Specifications) in case of a software update request, firstly SOLMAN message is created for the related transaction, which will be updated. The message representing the development is passed through the staff, respectively help desk/functional expert, module manager, functional consultant, developer, and test expert. The person who receives the message performs the transaction (design, implementation, test, etc.) assigned to him/her, adds the necessary documents into the message, standardizes it and passes the message to the next. The communication of the message between entities takes place by adding the ID of the target person into the Message Processor area. The status of the message is updated from the Status area. This message is unique, every message represents a case, and User, Status, and Time are the variables.

In order to analyze the current process in this organization with the help of process mining tools, the required logs and should be obtained from their appropriate tables. A data collection study is conducted to analyze how to extract such logs from SAP System, which logs all transactions.

A. Data Collection

Firstly, various tables are analyzed, and the relations between them are extracted by using Data Browser transaction, which provides the user to reach the contents of the tables and perform the necessary changes. These SAP tables are;

- *crmdorderadmh* table finds the unique field guid number from the message number. It is required since all the related information is connected to guid number, not to message number, which is available at the beginning.
- *cdpos and but000* table finds the assigned persons ID by using the guid number.
- *cdhdr* table provides sequential control.
- *crmjcds* table finds the changes in status and time by using the guid number.

Then, the program is designed to extract the necessary data from such tables. In order to obtain the suitable data format, Figure 4 shows the selection screen of the program.

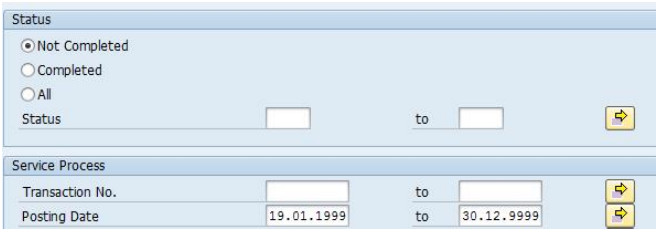

Fig. 4. Selection Screen of Designed Program

System uses event data which is formatted according to Figure 5, which contains mainly necessary information about message, user, status and time. In this event format, it is aimed to distinguish three different perspectives:

ISBN: 978-988-19252-7-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

- Process Perspective: This perspective mainly gives the answer of the question *How?*, and it focuses on the control flow of activities. By controlling this, it is aimed to reveal a good characterization of all possible paths.
- Organizational Perspective: This perspective mainly gives the answer of the question *Who?*, and it focuses on the user field and contains the involved users, programmers in the company.
- Case Perspective: This perspective mainly gives the answer of the question *What?*, and it focuses on properties of cases.

Message Number	User	Date and Time	Assigned User	Assigned Status
70000007597	139	2013/12/17 15:20:00	373	New
70000007597	373	2013/12/18 09:00:00	233	Develop-ment
70000007597	233	2013/12/18 17:12:00		Test OK

Fig. 5. Snapshot of the Data Format

After these processes, the data format is in XLS format, and it has to be converted to MXML format. In order to do this, firstly XLS format is converted to .csv (comma-separated values), since in conversion tools (such as Nitro, ProM Import) CSV extension is supported, and MS Excel can save an XLS format as a CSV format. To convert CSV to MXML format, firstly Nitro tool is used. As it can be seen on Figure 6; icons located on the upper part of the snapshot, can easily be matched with the relevant columns. After that, the most difficult part of the process mining Data Collection is completed.

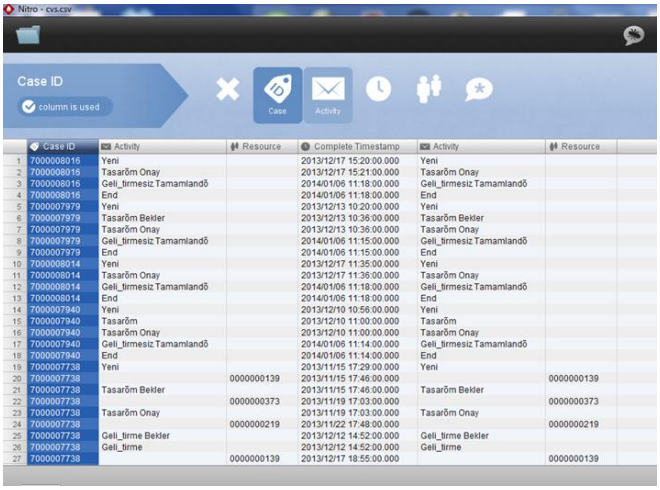

Fig. 6. Nitro

Unfortunately, Demo version of Nitro tool is inadequate, since the conversion of CSV format, which includes approximately 3,000 cases and 25,000 events is not free. Demo version of Nitro limits the event number with 500. Although it is very user friendly and very easy to deal with, another tool, ProM Import, as shown in Figure 7, is used for the conversion of a large number of events into MXML format.

In the future work of this study, it is aimed to test the proposed system with a large number of cases and events.

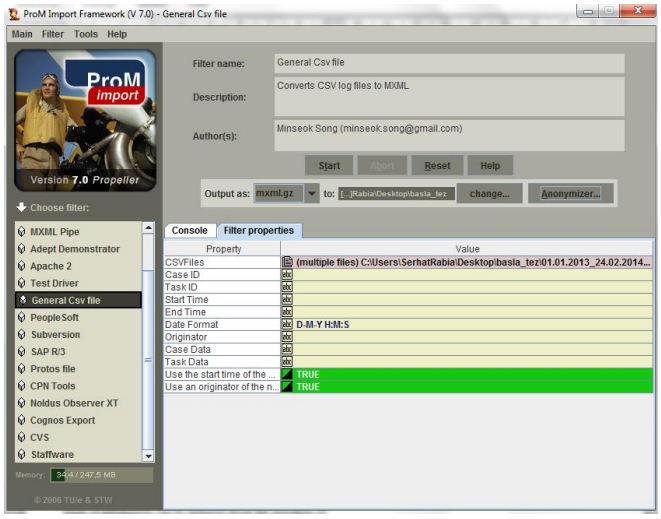

Fig. 7. ProM import screen to MXML

However, usage of this demo version also gives us some introductory knowledge and comparison about the mining algorithms and their results.

B. Processing Data

Data is firstly converted to MXML format, and then it is imported in ProM UITopia in order to discover the process of 71 cases. Figure 8 shows the interface of this tool.

Fig. 8. ProM UITopia

Then the results of using alpha miner and heuristic miner are analyzed according to the four quality criteria [2].

1) Four quality criteria: To evaluate the quality of the model, this technique is a very concrete tool.

- *Fitness*: If all traces in the log can be replayed by the model from beginning to end, it refers to a good fitness.
- *Simplicity*: If the model is as simple as possible and if it can explain the logs behavior well enough, then the model refers to a good simplicity.
- *Precise*: If the model does not have too much behavior then the model refers to good precision. A model that is not precise is underfitting, which over-generalizes the behavior.
- *Generalize*: if the model does not limit the behavior, the model refers to a good generalization. A model that does not generalize is overfitting, which allows for the exact behavior recorded in the log, So, process mining algorithms need a balance between overfitting and underfitting.

ISBN: 978-988-19252-7-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

When the alpha miner algorithm is applied to the event log, the model in Figure 9 is obtained. Figure shows that this model simply shows all different traces seen in the log. It can be easily seen that, although, it is precise and well-fitted, it is also very complex and over-fitting. As a result, it is clear that this simple algorithm is not enough to satisfy the trade-offs among the four quality criteria.

Fitness = +, Precision = +, Generalization = -, Simplicity = -

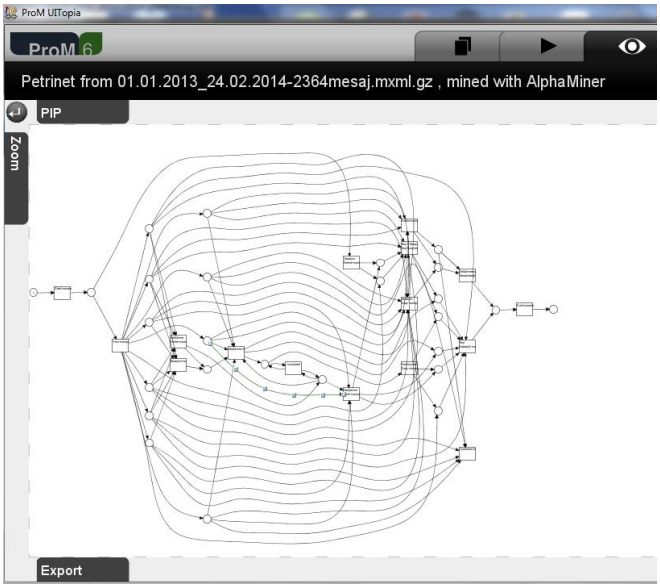

Fig. 9. Alpha Miner Algorithm

When Heuristic Miner algorithm is applied to the event log, the model in Figure 10 is obtained. This figure shows that the model is not only good enough but also simple and well-fitted. Besides, it balances between overfitting and underfitting.

Fitness = +, Precision = +, Generalization = +, Simplicity = +

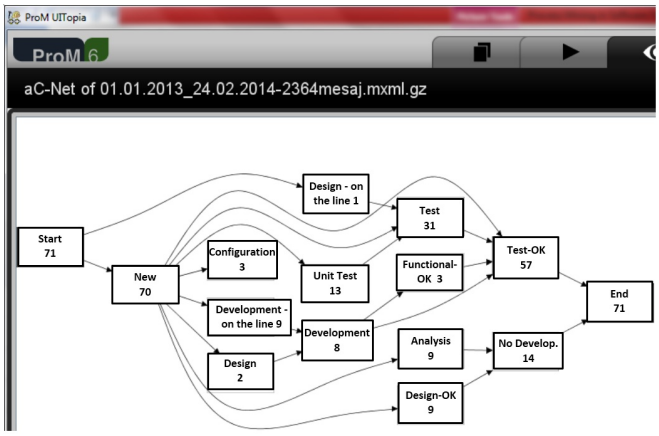

Fig. 10. Heuristic Miner Algorithm

IV. CONCLUSION

In this paper, Process Mining, the ProM Framework and Process Mining Algorithms are explained, and these algorithms are compared in a case study on software development and testing cycle process, which is carried out through SAP system, is analyzed. A case study is conducted on 71 cases in a software company who is responsible for Financial Accounting (FI), Material Management (MM), Quality Management (QM), Project Systems (PS) and Human Resources (HR) systems of a large organization, which contains more than 30,000 personnel in it.

To do that, data collection step is explained in detail. Then data is mined by using Alpha and Heuristic Mining Algorithms, results are evaluated according to their qualities. Results show that Alpha algorithm is inadequate to present the real picture of the related data. On the other hand, HM will be a better algorithm to analyze the event logs for organizations using an SAP-based information system.

The road ahead will be about building social network and Heuristic Miner algorithm model covering all event logs. Social Network will demonstrate all sociometric relations among users and Heuristic Miner model will reveal all other relations and quantify them. Alpha Miner algorithm is tested with a sample of event logs, and it is shown that this model is inadequate for this case. So, it is decided not to use this algorithm for the road ahead.

At the same time, as a future work, the proposed system can also be applied in distributed software development/execution environment such as [10]. By making this type extension it will be easy to increase the project team size and the software developers can be located in different cities/countries.

REFERENCES

[1] P. Weber, B. Bordbar, and P. Tino. A framework for the analysis of process mining algorithms. *Systems, Man, and Cybernetics: Systems, IEEE Transactions on*, 43(2):303–317, March 2013.

[2] Wil MP Van der Aalst. *Discovery, Conformance and Enhancement of Business Processes*. Springer, 2011.

[3] R. Saylam and O.K. Sahingoz. Process mining in business process management: Concepts and challenges. In *Electronics, Computer and Computation (ICECCO), 2013 International Conference on*, pages 131–134, Nov 2013.

[4] Wil van der Aalst, Arya Adriansyah, Ana Karla Alves de Medeiros, Franco Arcieri, Thomas Baier, Tobias Blickle, Jagadeesh Chandra Bose, Peter van den Brand, Ronald Brandtjen, Joos Buijs, et al. Process mining manifesto. In *Business process management workshops*, pages 169–194. Springer, 2012.

[5] C.S. Alves. *Social network analysis for business process discovery*. PhD thesis, Masters thesis, Instituto Superior Tecnico, Av. Rovisco Pais, 1, 2010.

[6] M.S. Saravanan and R.J. Rama Sree. Evaluation of process models using heuristic miner and disjunctive workflow schema algorithm for dyeing process. *International Journal of Information Technology Convergence and Services (IJITCS)*, 1(3), 2011.

[7] N.S.N. Ayutaya, P. Palungsuntikul, and W. Premchaiswadi. Heuristic mining: Adaptive process simplification in education. In *ICT and Knowledge Engineering (ICT Knowledge Engineering), 2012 10th International Conference on*, pages 221–227, Nov 2012.

[8] Wil MP Van der Aalst and Minseok Song. Mining social networks: Uncovering interaction patterns in business processes. In *Business Process Management*, pages 244–260. Springer, 2004.

[9] Wil MP Van Der Aalst, Hajo A Reijers, and Minseok Song. Discovering social networks from event logs. *Computer Supported Cooperative Work (CSCW)*, 14(6):549–593, 2005.

[10] N. Erdogan, Y.E. Selcuk, and O. Sahingoz. A distributed execution environment for shared java objects. *Information and Software Technology*, 46(7):445 – 455, 2004.

ISBN: 978-988-19252-7-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)